STILL LEARNING: INTRODUCING THE LEARNING TRANSFER MODEL, A FORMAL MODEL OF TRANSFER

By

Jeffrey David Olenick

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Psychology – Doctor of Philosophy

2020

ABSTRACT

STILL LEARNING: INTRODUCING THE LEARNING TRANSFER MODEL, A FORMAL MODEL OF TRANSFER

By

Jeffrey David Olenick

Although training has been a key topic of study in organizational psychology for over a century, a century which has seen great progress in our understanding of what a quality training program entails, a substantial gap persists between what is trained and what is transferred to the job. Reduction of the training-transfer gap has driven research on transfer-focused interventions which have proven effective. However, although we know a lot regarding *how* individuals learn new material, and correlates of *whether* they transfer that material back to their work environment, we know very little about *how* individuals go about choosing whether to apply their new knowledge to, typically, previously-encountered situations in their work environment and how those decisions unfold over time. Improving our knowledge regarding how individuals transfer learned material will lead to new insights on how to support the transfer of organizationally directed training, or any learning event, back to the work environment. Thus, the present paper introduces a formal model of the transfer process, the Learning Transfer Model (LTM), which proposes a process for how transfer unfolds over time and gives rise to many of the findings we have accumulated in the transfer literature. This is accomplished by reconceptualizing transfer as its own learning process which is affected by the dual nature of human cognitive systems, the learner’s social group, and their self-regulatory processes. The LTM was then instantiated in a series of computational models for virtual experimentation. Findings and implications for research and practice are discussed throughout.

This dissertation is dedicated to my loving family,
without whom I could not have made it this far

ACKNOWLEDGEMENTS

I would like to thank all those who have helped me over the years to get to this point.

To my professors: Dr. Steve W.J. Kozlowski for being my primary mentor through my doctoral career and always pushing me further in my thinking; Dr. J. Kevin Ford for being a good mentor and friend, and providing the developmental opportunities through which the ideas expressed within this dissertation were formed; Drs. Richard DeShon and Zachary Neal for being a part of my committee and introducing me to the kind of theory and models which provide the basis for my thinking in this paper; Dr. Ann Marie Ryan for guiding me when I wanted to switch careers and had no experience in this field; and Drs. Michael Stamm and Matthew Pauly for believing in me as a struggling undergraduate searching for my place in the world.

Thank you to my friends who have always been there to provide a distraction from the pressures of life.

A special thank you to my family, especially my mother and my grandparents for providing me with the foundation I needed to reach the success I have. And my father, although you left my life far too soon, you have provided a lifetime of inspiration.

Thank you to my son for providing a daily dose of motivation and levity, although it seems like a weird way to show it, this paper is very much a labor of love for you.

Finally, thank you to my wonderful wife, Catherine. You pulled me out of the darkest time of my life and helped get me back on track. All my accomplishments would be impossible without you.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

Introduction

Continuous learning is a mantra of organizations, often directed towards employees to emphasize the need for continually improving their knowledge and skills to maintain or increase their ability to perform their roles and advance their careers (e.g., London, 2012). To achieve continuous learning, organizations spend ever-increasing amounts of money on training programs, averaging more than $1,296 per employee in 2017 (American Society of Training and Development, 2018). Thankfully, organizations benefit from spending on training programs. For example, spending on training programs aids the development of knowledge, skills, attitudes, and other characteristics (KSAOs) which feed into the emergence of human capital resources for an organization (Ployhart & Moliterno, 2011), which in turn helps organizational profitability (Kim & Ployhart, 2014). Unfortunately, there remains a gap between what is taught in training programs and what gets transferred back to the work environment, sometimes referred to as the training-transfer gap (e.g., Vermeulen, 2002). Typical statements that only 10-percent of trained material is transferred to the job are not generally based in fact (Ford, Yelon, & Billington, 2011), but few would argue that no such gap exists. Transfer is "the extent to which the learning that results from a training experience transfers to the job and leads to meaningful changes in work performance" (Blume, Ford, Baldwin, & Huang, 2010, p. 1066). Thus, any rate of transfer less than 100-percent theoretically results in wasted money on the part of the organization as no meaningful changes in performance occur.

What, then, can we do to improve the rates of transfer and reduce the training-transfer gap? Over the last 100 years, research has taken a multi-pronged approach to this question, seeking to improve training programs at each of their three stages of pre-training, training, and post-training (e.g., Jaidev & Chirayath, 2012), which has led to a great deal of knowledge regarding the functioning of training programs (Bell, Tannenbaum, Noe, & Kraiger, 2017). That knowledge has improved programs largely by introducing principles to the learning event which can improve knowledge retention (e.g., Donovan & Radosevich, 1999; Dunlosky, Rawson, Marsh, Nathan, & Willingham, 2013). Though fewer principles exist within the post-training transfer stage of the training process, several consistent findings have emerged, including the use of implementation intentions (e.g., Gollwitzer, 1999), and the perceptions learners hold of the utility of their newly gained knowledge (e.g., Blume et al., 2010), among others.

Unfortunately, most studies on the transfer of learning in organizations are essentially correlational and/or cross-sectional in nature. Studies of training interventions typically measure learning and individual difference variables at the end of training, and then measure the transfer of learning at a single time point in the future. These tendencies can be seen in the types of studies available for meta-analyses of transfer (e.g., Blume et al., 2010; Blume, personal communication). With scientific emphasis on understanding causal mechanisms, it is tempting to interpret findings with time lags as causal in nature, but temporal precedence is only one precondition for establishing causality. Unfortunately, the effects of variables in transfer environments are often hard to isolate, especially in real organizations where random assignment is often difficult or impossible to achieve, though such isolation is possible (see Hanges & Wang, 2012, for a discussion of causal models). Thus, even though we are interested in the mechanisms which *explain* transfer, we are generally only studying *correlates* of transfer, and as we know, correlation does not equal causation, or alternatively prediction does not equal explanation (Muthukrishna & Henrich, 2019). To better understand the causal mechanisms that lead to transfer we must advance our understanding of transfer as it occurs over time and seek to discover the dynamic process which gives rise to what we currently observe as transfer. The study of such dynamic relationships is gaining increasing interest in our field (e.g., DeShon, 2012), and the study of person-level processes within the training context has been described as a frontier for research in this arena (Salas & Kozlowski, 2010), transfer-specific research must follow.

A dynamic process here refers to the interactions of lower levels of analysis that give rise to a higher-level observed variable in a process of emergence (e.g., Grand, Braun, Kuljanin, Kozlowski, & Chao, 2016; Kozlowski & Klein, 2000), those levels in this paper being the cognitive processes of an individual and their output behaviors. Other researchers are interested in studying dynamic processes of transfer and are attempting to unpack them. This can be seen in the increase of longitudinal designs studying transfer (Baldwin, Ford, & Blume, 2009), and, for example, the use of within-person analyses to understand the interplay of motivational changes over time with changes in transfer (Huang, Ford, & Ryan, 2017). However, most studies that claim to be interested in such dynamics do not really study dynamic relationships, and are largely restricted to cross-sectional designs or versions of growth modeling with few time points (e.g., Ford, Bhatia, & Yelon, in press; Gist, Stevens & Baveta, 1991; Cheng, 2016; Dierdorff & Surface, 2008; Zerres, Huffmeier, Freund, Backhaus, & Hertel, 2013). Even when longitudinal designs are utilized, the mere study of change over time does not constitute the study of explanatory dynamic processes because they rely on time as predictors and time is not explanatory (Dishop, Olenick, & DeShon, in press; Ployhart & Vandenberg, 2010). Training-transfer studies are motivated to show that change does occur, and thus that the program of interest is successfully affecting outcomes of interest. However, explaining change when it occurs is only half the battle and any process model should also be able to demonstrate when change will not occur, as a lack of change is still likely to be driven by a dynamic process and a lack of change does not imply the lack of dynamic processes (Dishop et al., in press).

Moves to better understand the dynamic process of transfer are being made, but only a few known models of transfer couch transfer in an iterative way such that it unfolds via repeated attempts that can dynamically affect future attempts. Existing models of the training process that do consider time generally treat transfer as an outcome that does not explicitly feed into future attempts (e.g., Baldwin, Magjuka, & Loher, 1991; Bell & Kozlowski, 2009; Cannon-Bowers, Salas, Tannenbaum, & Mathieu, 1995a; Cheng & Hampson, 2008; Colquitt, Lepine, & Noe, 2000; Thayer & Teachout, 1995), though some consider transfer as an input to future training cycles (e.g., Salas, Weaver, & Shuffler, 2012; Goldstein, 1986). A more dynamic view of transfer can be found in Chen, Thomas, and Wallace (2005) who proposed a multi-level model of training outcomes which discusses the episodic nature of the post-training environment. Additionally, Blume, Ford, Surface and Olenick (2019) introduced the Dynamic Transfer Model (DTM) which describes how trainees decide what to retain from their learning experience and apply their new KSAOs to their work environment in an iterative way. Their model is already impacting emerging research on how trainees transfer their news KSAOs over time (e.g., Vignoli & Depolo, 2019). However, the DTM has multiple weaknesses. One of these weaknesses is that the model is a verbal model instead of a mathematical formal model. Although such models are good for describing processes, they generally lack specificity and can struggle to make testable hypotheses (e.g., Vancouver, 2008). A second weakness of the DTM is that it relies heavily on self-regulation (e.g., Carver & Sheier, 1998), and person-situation interactionism (Hattrup & Jackson, 1996). Though these are good bases from which to begin building a dynamic theory of transfer, they leave out much of what we know about human cognition and social effects, at least, and therefore do not tell the whole story. Thus, further work is required to improve our theorizing regarding transfer as its own dynamic process.

To address existing gaps, this paper is guided by two research questions. First, what is the learning and decision-making process through which individuals go when attempting to transfer *new* knowledge to an *old* situation? Second, can a single, relatively simple, formal model of the dynamic transfer process account for our current findings in the transfer literature?

In addressing these two questions the present paper makes four key contributions. First, a process-oriented theory of learning transfer is introduced by building a formal mathematical model of that process, called the Learning Transfer Model (LTM). The LTM will begin to build a unifying theory of transfer in the workplace, partially answering calls for psychological science to move towards more unifying theories to improve the explanation of human behavior (Muthukrishna & Henrich, 2019). Second, the LTM further integrates several disparate but related theories, specifically: reinforcement learning (e.g., Sutton & Barto, 2018), Social Learning Theory (Bandura, 1977), and Control Theory (e.g., Carver & Sheier, 1998) within a dual process cognitive model framework (e.g., Kahneman, 2011). Third, computational approaches to reinforcement learning and dual process models will be more fully brought into the organizational literature. Finally, that integrative formal theory is instantiated in a computational model, allowing for virtual experimentation to explore the effects of the theory, the formation of testable predictions which may later be evaluated against real-world data, and providing potentially novel insights into the transfer process from which to build future practical interventions.

Review of Transfer Literature

Prior to building a process theory of transfer, we must take stock of the current transfer literature. This review will occur in two parts. The first is an overview meant to give a feel for where the field stands, particularly regarding our knowledge of transfer as a process and is not meant to be exhaustive. The primary points to be made are, first, that despite calls for viewing transfer as a process (e.g., Foxon, 1997), transfer has largely been treated as an outcome or a product. Second, because of the transfer-as-outcome view, transfer is typically measured at one or very few time points, which largely forgoes the ability to study transfer as a process, with few exceptions. Third, the nature of existing research is largely correlational and cross-sectional, resulting in a field of inquiry which can be characterized as a set of potentially useful but unrelated empirical findings. Fourth, the existing longitudinal transfer research does not generally examine dynamic processes, even if the authors state they are interested in them. Fifth, emerging theory on the training and transfer process is moving in the right direction to unpack within-person transfer processes but has far to go. The discussion will then move towards introducing a transfer process theory by more specifically describing some key concepts and findings which are critical to consider in the early stages of theory development.

Before diving into the review, we must define some terms. Broadly, this paper is interested in the learning process experienced by employees. Learning has been given many definitions (Salas, Weaver, & Shuffler, 2012). For example, learning can be viewed as a permanent change in the range of possible behaviors for an organism (Huber, 1991). More specifically it is “the process whereby knowledge is created through the transformation of experience” (Kolb, 1984, p.38). Training, then, is an organizationally directed learning experience aimed at introducing new knowledge, skills, attitudes, or other characteristics

(KSAOs) which expand the range of possible behaviors an employee may exhibit on the job. Training experiences are typically divided into three phases: pre-training, training, and post-training, or variations thereof (e.g., Beier & Kanfer, 2010). The present paper is focused specifically on the processes within the post-training phase. However, not all learning by employees is organizationally directed. Instead, employees learn much about how to accomplish their jobs and navigate their work environments when they are engaging with the relevant tasks and environment through informal learning processes (Tannenbaum, Beard, McNall, & Salas, 2010). This paper is concerned with all learning events, formal or informal, so the terms learners and learning will be used interchangeably with trainees and training in this paper and the model is suggested to apply to the transfer of learning from either formal or informal learning events.

The primary outcome of interest in the post-learning phase is the transfer of trained materials back to the work environment. According to Baldwin and Ford (1988), transfer consists of generalization and maintenance. Generalization is taking that which was gained through training and applying it to more or less similar situations as experienced in training once back on the job. Maintenance is the continued application of those new KSAOs to the job over time. The goal of this paper is to unpack *how*, not merely *whether*, learners transfer new KSAOs to their work environment. To begin studying the *how* of transfer, we will begin by assuming learners exit their learning experience with the ability to generalize that learning to their work, and now must find a way to actually commit to that transfer and maintain it over time. Therefore, the present paper is focused more directly on the maintenance portion of transfer than on the precondition of being able to generalize knowledge at all.

Historically, transfer has been treated as an outcome, or a product, instead of something that unfolds over time driven by a process. Foxon (1997) noted this tendency and its effects on our understanding of transfer, writing “the practice of measuring transfer as a one-dimensional ‘product,’ rather than assessing it in process terms, may have led practitioners to underestimate the amount of transfer. In other words, there may be more transfer occurring than is thought” (p. 43). Unfortunately, calls for more process-oriented transfer research have been largely unheeded until recently. Theoretical models of the training process still almost universally treat transfer as a single outcome. Part of this problem may be traced to the way researchers have traditionally treated the organizing model utilized by Baldwin and Ford in their classic (1988) review. In their model, training inputs lead simply to transfer, mediated by training outputs. However, instead of using the model to organize a disparate literature, researchers in part treated it as something to be tested. Although much useful knowledge arose from research inspired by Baldwin and Ford (1988), unfortunately the path that research took may have limited progress on the understanding of certain aspects of the training process by largely ignoring transfer over time.

The treatment of transfer as a product or outcome is the key limitation in understanding transfer specifically as a process. Interpreting transfer as an outcome or product led to the tendency to collect transfer measures at only one or a very limited number of time points. The typical study on transfer effects measures covariates of interest before, during, or at the end of a training event, or creates their experimental manipulations during training, and measure transfer at some later point in time. This tendency can be seen in the studies available for meta-analyses focused on transfer effects (e.g., Blume et al., 2010; Blume, personal communication), even though transfer includes both generalization and maintenance (Baldwin & Ford, 1988). Generalization could be considered as something which either happens or does not and therefore be measured at a single time, but maintenance implies the continuance of transfer over time and thus requires multiple measurements to study it. Baldwin, Ford, and Blume (2009) in their updated review found the number of time points examined in the transfer environment had improved, but that the number was still limited. The lack of stronger longitudinal designs, where all measures of interest are measured at multiple time points, limits the analyses and knowledge we may gain. For example, single measurements are inadequate for cross-lagged designs that can start to unpack dynamic process-like relationships underlying phenomena (e.g., Kenny, 2005). Thus, transfer research is not examining dynamic processes, even when researchers are interested in them. A recent example is a measurement piece by Ford, Bhatia and Yelon (2019) which reports a multidimensional measure of transfer as use. The authors state they are interested in the dynamics of transfer, but their data collection is a single time point for each participant and conduct no dynamic analyses.

Transfer research also lacks a general guiding theory. The lack of a guiding theory or framework has resulted in a large set of potentially useful but largely unrelated empirical findings. Existing models of the training process are not scientific theories in that they do not posit universal mechanisms underlying the process, especially in a way that can be applied directly to transfer. Instead, existing models are generally tools for organizing the vast set of empirical findings in a coherent way; they do not tie those findings together into a unified whole. This can be seen in any of a number of reviews which have expanded overtime to include more detail because the extent of empirical findings has also greatly expanded over the past three decades, but the essential structure remains highly similar (e.g., Baldwin & Ford, 1988; Salas et al., 2012). Within single studies, on the other hand, theory can be found to guide hypothesizing. For example, as reviewed by Beier and Kanfer (2010), common theories of motivation utilized to study transfer include goal choice and self-efficacy from self-regulation (Bandura, 1977), expectancy theory (a.k.a. Valence, Instrumentality, Expectancy (VIE) Theory; Vroom, 1964), individual differences such as the Big Five personality traits (McCrae & Costa, 1987), and transfer of training climates (Rouiller & Goldstein, 1993). The predictions made based on these theories independently show positive effects on transfer outcomes (e.g., Blume et al., 2010), but are not integrated into any comprehensive whole, leaving the empirical findings scattered and in need of some underlying scientific framework to unify them. Such work also answers calls for scientific frameworks to enhance the rigor of psychological science (Muthukrishna & Henrich, 2019).

As mentioned by Baldwin et al. (2009), the number of studies examining multiple time points has improved over the years. In many ways, these studies are applications of more typical learning or performance studies which examine learning curves on a task of interest. A few examples will suffice. Gist, Stevens, and Baveta (1991) tested post-training interventions to improve maintenance and transfer, finding that pre-training self-efficacy relates to both initial and delayed performance on a target test. They also found that the effect of efficacy on maintenance was moderated by the type of training the learner received. Vancouver and Kendall (2008) made the important point that relationships may differ when examined at the within- instead of between-person level, when they showed efficacy can be negatively related to performance and motivation in some learning contexts at the within-person level. Their finding opposes the common view that efficacy and performance are positively related, but which is typically studied at the between-person level. Dierdorff and Surface (2008) showed that skill-based pay was related to skill maintenance over a seven-year period with multiple measurement points. Finally, Scholz, Nagy, Schuz, and Ziegelmann (2008) studied 30 initially untrained runners for a year, taking 11 measurements of their running tendencies as they prepared for a marathon. At the between-person level they found trend in efficacy predicted trend in the amount of running, and that fluctuations in efficacy predicted fluctuations in running. At the within-person level, controlling for between-person trends, amount of running was predicted by efficacy and intentions, among other variables.

Recent studies aim to unpack within-person effects specifically on training transfer. The best examples may be those of Huang and colleagues. Huang, Blume, Ford, and Baldwin (2015) showed in a meta-analysis that maximal and typical transfer are weakly related, and that predictors of the two forms of transfer differ. Specifically, maximal transfer was better predicted by abilities, while motivational measures were better predictors of typical transfer. Their findings suggest more research is necessary to unpack why those factors differentially predict aspects of transfer. Huang, Ford, and Ryan (2017) then studied within-person variability in transfer in a multi-wave design. They showed that initial attempts to transfer were best predicted by post-training self-efficacy and that motivation to transfer better predicted rates of change in transfer. Unfortunately, the basis for this study relies on growth modeling so is not truly dynamics (DiShop et al., in press), but it represents a significant step forward conceptually in understanding the within-person nature of transfer.

However, all is not lost, and researchers are making theoretical advances regarding the process underlying the transfer of learning. Repeated calls are being made to study the training process, including transfer, from a multi-level perspective. Such arguments center on not only the need to better understand higher level organizational effects on training and transfer, but also to consider the within-person nature of the training and transfer processes (e.g., Mathieu & Tesluk, 2010; Sitzman & Weinhardt, 2019). Such calls have in part manifested in micro-level research, such as on preventing knowledge and skill decay (Cascio, 2019). These advances in part emphasize that transfer is an episodic process and theory aimed at unpacking that process is emerging. Blume, Ford, Surface and Olenick (2019) described transfer as a self-regulatory process where learners proceed through episodes of deciding to retain or discard new KSAOs in favor of their existing repertoire, attempt to apply those KSAOs, receive feedback on their attempts, and reiterate the decision process. That process interacts with organizational factors to determine how it unfolds over time. Surface and Olenick (forthcoming), are developing a mechanistic model of transfer seeking to unpack the cognitive processes underlying the general process described by Blume et al. (2019). This new model (Surface & Olenick, forthcoming) describes how the transfer process relies on cognitive processes and the overriding of automatic responses to transfer new, non-automatic KSAOs, and how the individual develops in this process over time. Both theories make substantial strides in describing transfer as a process. However, they remain limited by their informal linguistic nature. Further work is required to build on these models to enhance formalization, increase prediction precision, and falsifiability.

All these advances are important and have provided a wealth of useful information. However, much work remains to provide a process-oriented explanation for when and why learners transfer to their work environments. This paper argues that advancement may be made by reconceptualizing transfer as another learning process, rather than something theoretically removed from the processes which drive a learning event. By reframing transfer as learning, we can draw on existing process-oriented theories of learning to provide a strong foundation from which to begin, including both informal natural language theories, and more formal mathematical and computational approaches. For example, Tannenbaum et al. (2010) described a dynamic model of informal learning on the job where employees learn over time through an iterative process of intent, experience, feedback and reflection, which is affected by organizational and individual factors. More formal conceptualizations of learning through experience can be found, such as reinforcement learning (e.g., Sutton & Barto, 2018). Some of the basic mechanisms, such as experience, of these theories and others will be evident in the model explicated below. However, the primary point is *that transfer theory may be advanced by approaching transfer as the process by which individuals learn if a new KSAO is a good fit for their job*. The model presented here will be called the Learning Transfer Model (LTM) for the double meaning of transferring learning to a target job environment, and individuals going through what amounts to a process of learning to transfer their new KSAO to the target environment, or not. This conceptualization emphasizes the individualized nature of the transfer process where learners eventual transfer outcomes are largely a function of the ability of their training to fit the needs of their job, and them learning through experience the fit between their training and their needs.

Computational Modeling and the Modeling Cycle

Before beginning, it is important to set expectations regarding the approach to theory building undertaken in this paper, and discuss the implications that approach has for the theory outlined below. The present paper takes a computational approach to theory building. Computational modeling is a useful tool for building new process-oriented theory for multiple reasons. First, it forces the formalization of one's theory, improving the theory's internal logic, especially as it evolves over time (e.g., Vancouver, 2008; Vancouver & Weinhardt, 2015). Second, computational modeling allows for the exploration of the theory's implications in a low-risk environment. Third, those virtual experiments allow for better understanding of phenomena of interest, and can, but do not have to, lead to novel insights which may not be apparent in one's theorizing or unguided data collections (e.g., Miller & Page, 2012), though such insights can then be tested using targeted data collections on real subjects (e.g., Vancouver, Weinhardt, &

Vigo, 2012). Importantly, a formal theory can also provide specific point estimates for effect sizes one would expect to observe in the real world. Although making such specific predictions is not the historical norm in psychology, doing so is a stronger form of science where we can support or refute an underlying theory by assessing the fit of observed effects to predicted ones using Bayesian inference (Dienes, 2019). More generally, when one makes the mechanisms of a theory explicit as is required by computational modeling you can be absolutely sure of what has led to the outcomes of the model in a way not typically achieved in traditional theory building. That is, when we collect empirical data in our field, we often propose hypotheses regarding the direction of relationships between constructs of interest which we believe follow from the logic of some underlying theory we are drawing upon. For example, we might predict that self-efficacy and performance are positively related while drawing on Social Cognitive Theory (Bandura, 1977) to discuss why we should expect such a relationship. However, when we only measure self-efficacy and performance and find the predicted relationship we have not actually tested the underlying mechanisms driving that relationship, such as effort (e.g., Vancouver & Kendall, 2008) and therefore cannot be certain our underlying theory is the actual explanation for that relationship, we can only be sure that the relationship is consistent with our expectations. However, when you use a computational model of the type used in the present research you can be sure that the mechanisms you specify led to the relationships between any higher-level emergent properties you may be interested in because they are the only mechanisms in play. Finally, a computational approach to theory building allows for an iterative process whereby a relatively simple form of a theory can be built, explored, then expanded over time as necessary to account for phenomena of interest. Researchers have argued that this approach is the direction in which our field should be evolving (e.g., Kozlowski & Chao, 2012), and may be of particular use for studying the training process (Salas & Kozlowski, 2010).

The iterative approach to theory building and modeling was described by Railsback and Grimm (2012) as the Modeling Cycle. The Modeling Cycle is composed of six total steps: 1) formulate the question, 2) assemble hypotheses, 3) choose model structure, 4) implement the model, 5) analyze the model, and 6) communicate the model. The process is iterative in that step five feeds back to step one, except when the author decides the time has come for communication. Over time, the theory and associated model is developed and explored, becoming increasingly sophisticated or more representative of the phenomenon of interest. By starting "simple", this paper acknowledges that the resulting theory will not be a perfect picture of the transfer process, but that is not the intent. Rather, this model provides a starting point for future development while hopefully providing useful insights into the transfer process. This approach stays true to the principles of theoretical parsimony as outlined by Box (1976, p. 792) that "since all models are wrong the scientist cannot obtain a 'correct' one by elaboration. On the contrary following William of Occam, he should seek an economical description of natural phenomena", or the corollary of Occam's Razor, popularly attributed to Einstein, that a theory "should be made as simple as possible, but not simpler".

Transfer Findings for Which to Account

Along with his admonition for parsimony, Box (1976) describes "worrying selectively" as an aspect of the scientific method. That is, "since all models are wrong the scientist must be alert to what is importantly wrong" (p. 792). In this section, potentially important concepts and findings will be discussed with reasoning for why or why not they need to be included in initial steps in building a transfer process theory. The discussions here are not meant to be in-depth reviews of each topic. The goal is to define the concept and general findings, based in meta-analytic evidence where possible. This approach is deliberate in considering the initial stages of development for the present theory. An overemphasis on examining nuance can inhibit the development of sound theories of human behavior because it stands in the way of the abstraction on which good theory depends (e.g., Healy, 2017). Within psychology, researchers are incentivized to focus on theoretical contributions in their work (e.g., Olenick, Walker, Bradburn, & DeShon, 2017) which for most studies means extending an existing theory by examining a new application or moderation of that theory. However, with no incentive to replicate findings the supposed nuance gained by such studies can long go unchallenged and cloud the development of a core theory to unify those findings. Further, relying on single studies to build informal theory is treacherous at best because interpretations and conclusions from single studies can differ greatly depending on who does the analysis and interpretation (Starns et al., 2019). Thus, it is imperative that a potentially unifying theory account for general findings before exploration of more nuanced findings which may be misleading. To this end, the meta-analytic effects discussed here are not to be treated as precise targets for replication in the models explored in this paper. Instead, the meta-analytic effects are general guides for the patterns of relationships expected from the LTM as there are limitations to the use of meta-analyses as exact targets such as variability in the contexts in which their underlying studies were conducted, measures used, and theoretical underpinnings among other between-study differences that get aggregated across when estimated meta-analytic effects. The model presented in this paper is meant to be a general theory of training transfer and should therefore represent the general findings of applicable meta-analyses, but the exact point estimates from those meta-analyses may be overly restrictive targets for a theory in the initial stages of development as is the LTM, and future work should look to refine the LTM to better target precise effects in their applicable research contexts.

Practice and Overlearning

The effects of practice on important training outcomes is well established. Practice on tasks is related to important performance outcomes as individuals tend to improve over time with exposure to said task. For example, Hausknecht, Di Paolo, and Moriarti Gerrard (2007) found that test scores show an increase upon retesting with a meta-analytic effect of .26. Such practice effects are critical when considering personal outcomes, such as employment decisions (e.g., Olenick, Bhatia, and Ryan, 2016). Similarly, practice is critical within learning contexts for improving important outcomes and is considered one of the best strategies for improving learning and retention (e.g., Dunloski, Rawson, Marsh, Nathan, & Willingham, 2013). Within the transfer environment, practice of skills is also essential for maintenance, where Arthur, Bennett, Stanush, and McNelly (1998) found that skills deteriorated significantly over time without use in a meta-analysis.

Relatedly, researchers have explored the use of overlearning as a design feature of training. Overlearning is essentially the use of extreme levels of practice to develop automaticity before the learner leaves the learning event. The development of automaticity is a key outcome in training and the development of expertise (e.g., Ericsson, 2006; Goldstein & Ford, 2002). Meta-analytic investigation of the effects of overlearning on retention show an uncorrected relationship between overlearning and retention of .298 (Driskell, Willis, & Copper, 1992). Thus, it is important for the transfer process theory to account for improvement in transfer when practice and overlearning are part of the training design before the learner even enters the transfer environment.

Utility Reactions

Utility reactions are a learner's perception of the usefulness of their learning experience (e.g., Ruona, Leimbach, Holton, & Bates, 2002), typically collected via an affective reaction measure at the end of a training session. Researchers predict that when adult learners see new information as useful to them, they are more likely to utilize that information in the future. This prediction fits with training principles regarding the need to improve trainee motivation to learn or transfer by connecting the material to personal outcomes (e.g., Bauer, Orvis, Ely, & Surface, 2016). Interestingly, relatively few studies actually examine utility reactions despite their demonstrated strength in predicting transfer outcomes. In Blume et al.'s (2010) meta-analysis, only nine studies were found that met their inclusion parameters, but those studies demonstrated a corrected relationship with transfer of .46, making utility reactions one of the strongest overall predictors of transfer and important to account for in the present model.

Work Environment

Work environmental factors have long been considered an important driver of transfer, often referred to as transfer climate. Transfer climate includes aspects of supervisor and peer support, opportunity to use, supervisor sanctions, positive and negative personal outcomes, and resistance to change (Nijman et al., 2006: Rouiller & Goldstein, 1993; Holton et al., 1997; Holton et al., 2000). This paper focuses on supervisor and peer support and opportunity to use. Supervisor and peer support are important antecedents of training success (e.g., Baldwin & Ford, 1988). These two factors are part of social support, which is an ability to draw on emotional and task resources of others (Steele-Johnson, Narayan, Delgado, & Cole, 2010). Social support has important effects on stress and well-being of individuals, with perceptions of support being potentially more important than actual support (e.g., Kessler, 1992). The importance of support for the transfer of training has been confirmed via meta-analysis with Blume et al. (2010) finding a corrected relationship of .21 between support and transfer. Several studies on the effects of supervisor support, specifically, are interested in exploring the mechanisms through which support operates to affect training outcomes. For example, Nijman et al. (2006) found that support affects transfer through perceptions of transfer climate and motivation to transfer. However, most studies in this area are cross-sectional in nature. Even Foxon (1997), who argued for examining transfer as a process, examines supervisor support but collected measures at a single time point in the transfer environment. Similarly, Nijman et al. (2006) develop a process model of transfer but are limited to a small sample and a cross-sectional design. Thus, it is important to consider the effects of support for transfer of a new KSAO, but the development of support effects over time need further examination.

The situations in which learners find themselves attempting to apply their new KSAOs also impact transfer. One important way situations differ is the degree to which they are weak or strong. Situations are strong to the extent they provide clear context clues on the appropriate courses of action to take (Meyer, Dalal, & Hermida, 2010). Strong situations dictate the actions that must be taken while weak ones allow more room for individual differences to influence how to proceed, and thus affect related outcomes. For example, Judge and Zapata (2015) showed that the effects of personality traits on performance were higher in weak contexts than in strong contexts. In transfer environments situation strength manifests in various ways, such as if the received training is the organizationally required way to carry out a task transfer would be more likely. Or, in the relationship between supervisor and trainee, closer and less-autonomous supervision should create a stronger situation and lead the trainee to transfer their new KSAOs in a way more consistent with the desires of their supervisor (e.g., Yelon & Ford, 1999).

All such higher-level factors fit with calls for multi-level investigations of training and transfer effects (e.g., Mathieu & Tesluk, 2010; Sitzman & Weinhardt, 2019). Multi-level theory (Kozlowski & Klein, 2000) emphasizes the nested nature of phenomena in organizational psychology. Namely, measurements across time are nested within individuals, individuals within teams, teams in organizations, and so on. Nesting has implications for both how we study phenomena, and how phenomena are likely to manifest. It has been argued that research should examine target phenomena from a bracketed perspective, including effects of both one level above and one level below the target phenomenon (Hackman, 2003). In the present study this includes explication of an individual process which occurs over time, and higher-level effects on that process imposed by such concepts as situation, opportunity, and climate. Overall, environmental effects including transfer climate, support, as well as constraints or opportunities for use have a meta-analytic relationship of .22 with transfer (Blume et al., 2010).

Implementation intentions

Psychologists in several areas of inquiry have studied the potential for implementation intentions to reduce the intention-behavior gap (e.g., Schniehotta, Sholz & Schwarzer, 2005). Implementation intentions link situational cues and action responses in an “If-Then” format, such that when situation X arises the person will respond by doing Y (Gollwitzer, 1999), and have been shown to have a substantial meta-analytic effect on goal attainment (Gollwitzer & Sheeran, 2006). They also account for attainment beyond the strength of one’s goal alone (Sheeran, Webb, & Gollwitzer, 2005). Weiber, Thurmer and Gollwitzer recently (2015) described the mechanisms underlying the functioning of implementation intentions. Implementation intentions in the “If-Then” format form a strong relationship between mental representations of the theoretical goal-relevant situation and the goal-directed action, delegating action control to a lower order cognitive process, changing the normal top-down processing approach of goal attainment into a more automatic and efficient bottom-up process.

Health psychologists have utilized implementation intentions to improve the effects of patient education programs. For example, Harris and colleagues showed that both implementation intentions and self-affirmation increased fruit and vegetable consumption at seven-day and four-month follow-ups (Harris et al., 2014). Kendzierski, Ritter, Stump, and Anglin (2015) showed the moderating effect of self-schemas on implementation intentions. In two studies they showed that implementation intentions increased healthy eating habits among individuals who already held a self-schema of being healthy eaters, meaning implementation intentions work better for individuals who already see themselves as approximating the end goal. A systematic review recently suggested the effect of implementation intentions has a small but reliable effect on healthy eating behaviors (Turton, Bruidegom, Cardi, Hirsch, & Treasure, 2016). Thus, although not ubiquitous in organizational training studies, implementation intentions show important effects and should be accounted for in a model of transfer.

Maintenance Curves

Maintenance is one of the two primary aspects of transfer outlined by Baldwin and Ford (1988) and Baldwin, Ford, and Blume (2009). Baldwin and Ford (1988) describe possible trajectories a learner may take in displaying transfer which are labeled maintenance curves. These potential trajectories range from initial lack of transfer with later increases in transfer rates, to initially high levels of transfer that decrease over time. Such trajectories can be studied using growth modeling techniques, as accomplished in the study by Dierdorff and Surface (2008) on the effects of skill-based pay on maintenance. Unfortunately, because the study of maintenance curves requires several waves of data collection, they are rarely studied in primary research. A transfer process model should be able to explain why an individual may take any one of the potential general transfer trajectories. One advantage of using computational modeling to explore the present theory lies in the ability to explore such curves in an environment that does not necessitate large scale data collections.

Self-efficacy

Self-efficacy is the belief of an individual in their ability to execute desired behaviors in the pursuit of some outcome (Bandura, 1977). Efficacy is a central variable in self-regulation theory which will be more thoroughly introduced below. Importantly, according to Bandura, efficacy is the primary way in which individuals show agency in affecting their personal environments. Within the learning context, efficacy drives outcomes through the amount of effort the individual is willing to place into the task in question (e.g., Vancouver & Kendall, 2006). In examining the effect of efficacy on transfer, it is common to collect feelings of efficacy at the end of a training event to predict future use. Across studies efficacy has been shown to be a moderate predictor of transfer (Blume et al., 2010). Given the centrality of efficacy to the key theory of self-regulation and the demonstrated effect of efficacy on transfer, efficacy is another variable which holds importance for the LTM.

Skill type

A potentially critical aspect to consider is the nature of the skill targeted for transfer. A typical delineation between skill types is open versus closed. Closed skills have a relatively strictly defined way in which they may be applied, for example there may be only one way to successfully operate a machine. Open skills are those over which the trainee has more discretion regarding how they are applied to their job, for example how to handle an interpersonal interaction (e.g., Yelon & Ford, 1999). Similarly, Laker (2011) introduced soft and hard skills.

Hard skills are technical skills or those that define how to do a given task. Soft skills are those that have a more inter or intrapersonal focus. These categories are like open and closed skills but are argued to go further in differentiating the skill types in question. The type of skill studied may have important implications for transfer on its own and affect parts of the LTM. For example, Laker (2011) argues that soft skills are less likely to transfer because the trainee is more likely to have prior experience that needs to be overcome, and that feedback is more difficult to receive accurately. In addition, the level of support for transfer may matter more for open and soft skills than closed and hard. For example, Salas, Milham, and Bowers (2003) argued that as the military moves towards more open-skills training programs a more supportive environment would be required to enhance transfer as trainees would have greater discretion over the implementation of their new skills. Yelon and Ford (1999) further discuss the interplay between closed versus open skills and the level of autonomy a trainee has from their supervisor in determining transfer outcomes.

However, it is important to begin building explanatory theories as simple as possible and later iterations may build in complexity. For that reason, the initial LTM will be more directly applicable to hard or closed skills because they are more straightforward. This does not mean the proposed theory is inapplicable to more open-type skills as the underlying process driving transfer is likely the same and future investigations will be required to unpack any nuance required to account for differences in transfer outcomes between the various skill types.

Near versus Far Transfer, Adaptive Transfer and Adaptive Performance

Near and far represent a key distinction in describing the nature of the transfer task. Near transfer is when tasks in the transfer environment closely resemble those on which the learner received instruction, allowing more direct application of what was learned to the transfer environment. Far transfer is when the task in the transfer environment is different in some larger degree from the task on which the learner received instruction, requiring greater adaptation on their part (Beier & Kanfer, 2010). The type of transfer has potentially differential effects on other important variables. For example, it was originally demonstrated that self-efficacy was only related to transfer when near transfer was required (e.g., Mathieu, Tannenbaum, & Salas, 1992; Martocchio, 1992). However, it was later shown that self-efficacy is important in determining far transfer as well, though potentially to a different degree (e.g., Kozlowski et al., 2001). Related to far transfer is adaptive transfer. Adaptive transfer occurs when knowledge from training is applied to a task which is not identical to that which was trained but is instead an adaptation of that task. Adaptive transfer can also involve the generation of novel approaches to problem solving (e.g., Beier & Kanfer, 2010; Smith, Ford, & Kozlowski, 1997).

More broadly, Baard, Rench, and Kozlowski (2014) reviewed research on adaptation and adaptive performance, which are related to generalization. Based on their review, the field of adaptive performance is largely unorganized, characterized by multiple approaches which are not in agreement with one another. To provide some structure, the authors introduce a taxonomy of performance adaptation. The most relevant category they define for the present purposes is that of domain-specificity, which is based in training and skill development. They write that

> "a key *assumption* of this approach is that specific capabilities underlying performance adaptation can be learned and that their application is specific to a knowledge and skill domain rather than general across a range of work situations. The primary *target* for this work is to develop knowledge, skills, and capabilities via training or other developmental experiences that can increase performance in

a task context that shifts in novelty, difficulty, and/or complexity" (p. 51; emphasis in original).

Further, within adaptation research, decision-making and learning are important topics of study, which are primary foci of the LTM. Examples of research in this domain include decision-making tasks (e.g., TANDEM), and how individuals adapt their decision making in changing situations which drives adaptive performance.

However, adaptation is more concerned with applying existing knowledge to new and changing situations, not with applying new knowledge to old situations, which is more the domain of transfer. This is a close but important distinction. The adaptation of existing knowledge is important and interesting, but a large portion of actions undertaken by typical employees are relatively routine, even in complex jobs (Susskind & Susskind, 2017). Further, estimates based on experience samples are that 45-percent of behaviors are repeated in the same location every day (Neal, Wood, & Quinn, 2006; Wood, Quinn, & Kashy, 2002).

This paper most directly concerns situations where the encountered situation is stable *enough* that the same *general* approach to the task may be applied, thus avoiding the complications of adaptation, skill type, near or far transfer, etc., for the time being. This is directly applicable to types of jobs that are very consistent in their nature but is also in line with the idea that teaching principles they can apply to a broad range of situations is beneficial. The argument is made that the same basic process of learning about the potential uses of a newly trained KSAO will be applicable to both situations. However, it is agreed that this process is complicated by attempts to apply training to more adaptive tasks. Thus, the initial LTM should be interpreted as directly applicable to transfer tasks which are broadly definable as near transfer, but with potential insights for the processes underlying far transfer as well.

Study 1: Base Learning Transfer Model

To investigate the noted gaps in the transfer literature, the remainder of this paper will be dedicated to introducing and exploring a formal model of the transfer process called the Learning Transfer Model (LTM). The complete model will be described and tested in multiple iterations drawing on existing work in fields other than organizational psychology to form the basis of the proposed transfer process. The first model is primarily based on theories of Dual Process Cognition(e.g., Kahneman, 2011) and reinforcement learning (e.g., Sutton & Barto, 2018), and informed by work on habit formation and change (e.g., Neal, Wood, & Quinn, 2006).

Dual Process Models and Habits

I argue that a primary shortcoming in the existing training and transfer literature for the study of transfer as a process is a lack of basis in established cognitive theory. One particularly underutilized framework, not just in the training literature but across Organizational Psychology more broadly, is that of Dual Process Cognition. By drawing on existing Dual Process Theories we can provide an overarching framework from which to explain how learners may process their transfer situations and make decisions regarding how to respond. Once established, we can discuss how other important theories may further explicate key mechanisms withing the dual processing framework.

One thorough and accessible explanation of dual process theory comes from Nobel Laureate Daniel Kahneman (2011), though other versions exist (e.g., Pennycook, Fugelsang, & Koehler, 2015; Bago & De Neys, 2017). Kahneman (2011) explains that humans have two separate information processing and decision-making systems. The first system, conveniently labeled System 1, is characterized by fast, automatic information processing which requires little effort and makes decisions based on heuristics learned over time which tend to result in an acceptable level of success, whatever success may be. Automatic decisions allow humans to carry out most of their daily information processing and decision making without becoming cognitively overloaded, but these decisions also tend to be biased and suboptimal. On the other hand, System 2 is an effortful processing system which moves slower and requires conscious cognitive effort. System 2 tends to make more nuanced decisions but may lead to the same conclusion which would be made by System 1. Kahneman (2011) also argues that humans are lazy cognitive processors and will default to the use of their System 1 processing whenever possible. Kahneman's approach to cognition and decision making has the added benefit of arising from behavioral economics, which tends to be more formal in its theorizing and replicates more frequently than traditional psychological research. It has been suggested that behavioral economics and dual processing theories show promise for the building of unifying, but falsifiable, psychological theory (Muthukrishna & Henrich, 2019; Popper, 1959).

Criticisms of dual processing theories have been levied by many researchers. Evans and Stanovich (2013) outlined and responded to the five most common criticisms. Those criticisms include 1) dual process theorists have offered multiple and vague definitions of those processes, 2) proposed attribute clusters are not reliably aligned, 3) the existence of a continuum of processing styles and not discrete types, 4) single-process accounts may be offered for dual-process phenomena, and 5) evidence for dual processing is ambiguous or unconvincing. Evans and Stanovich (2013) respond to each of these in turn, but generally such criticisms are levied against dual process theories *en masse* instead of against single theories, ignoring specific developments within dual process theories. Their points include that characterizing cognitive processing as strictly dichotomous is oversimplified and processing should be viewed as more varied, with some processes being more automatic and others less so. Such a view overcomes the continuing charge of unreliable alignment of attribute clusters (Melnikoff & Bargh, 2018a, Melnikoff & Bargh, 2018b; Pennycook, De Neys, Evans, Stanovich, & Thompson, 2018). Evans and Stanovich (2013) further outlined that a dual process conceptual approach better fits the data patterns of cognition than any other explanation, such as a single process model, and that it is largely nuances within the field of dual processing itself that remain to be fleshed out rather than disregarding the framework as a whole. The view of dual processing as the essential framework for cognition becomes stronger when organizing it from a default-interventionist perspective. The default-interventionist perspective views processing as being essentially automatic in nature for most instances, where we generate automatic responses and it is then up to the more deliberate processes to intervene or not. Finally, clarity may be brought by referring to these two processes as type 1 and type 2, which is meant to overcome the shortcomings of using the system terminology that gives the false impression that there are two clearly identifiable processing systems.

The current paper cannot clarify the nature of dual processes. Instead, this paper argues that the dual process framework, though imperfect, is a useful dichotomization for forming parsimonious explanations for meso-level processes which are driven by underlying cognitive systems. The dichotomy used here will refer to type 1 and type 2, with type 1 processes being generally more automatic and unconscious and type 2 being generally more deliberate and conscious, though it is understood this is not necessarily a perfect characterization. In addition, this paper adopts the view of Evans and Stanovich (2013) that the two processing types occur in a default-interventionist, sequential, fashion. Approaching dual processing from this general perspective will provide a framework from which to approach the transfer process, representing an imperfect but significant step forward in understanding that process.

As previously stated, this paper is fundamentally about learning, and researchers have previously described dual process models of knowledge and learning. For example, Dienes and Perner (1999) distinguished between implicit and explicit knowledge. Implicit knowledge largely, but not exclusively, being that which is automatic, unconscious, nonverbalized, and declarative. Implicit knowledge underlies explicit knowledge as knowing something explicitly implies you know the information underlying it but knowing something implicitly does not necessitate being able to make it explicit. Sun, Slusarz, and Terry (2005) built on the distinction between implicit and explicit knowledge by explicating the CLARION model of learning which includes both implicit, bottom-up, and explicit, top-down, forms of learning in skill acquisition. In implicit learning individuals gain knowledge through direct experience, which a more unconscious form of learning and may not lead to knowledge which the individual can directly articulate. Such knowledge occurs in the development of learning patterns in complex recognition tasks, or in the learning of grammatical rules in real or made-up languages. Explicit knowledge acquisition can be delivered directly from the outside environment, such as being told the decision rules required for a given task. Over time, implicit knowledge can work its way up to become explicit where the learner can refine rules in a more conscious way. This exemplifies the split between more unconscious type 1 and more conscious type 2 processing in a learning environment. However, their model is directly concerned with skill acquisition, so is more directly applicable to the training event itself in an organizational training process, and not to the process of transferring that skill.

The automatic nature of type 1 processing is of major importance for the present paper. Successful training interventions have long attempted to develop a degree of automaticity in skills that are being targeted. Interventions have been able to develop automaticity particularly through overlearning approaches (e.g., Arthur, Bennett, Stanush, & McNelly, 1998), which effectively have the learner repeat a process until they are engrained to the point of an automated response. However, what we do not appreciate enough is that when we introduce a new KSAO to an employee there is likely some existing KSAO that the new one must override which, at the very least, has a head start on the development of automaticity. We do study expertise development where the essential process is the breaking of old automatic processes and replacing them with better processes (e.g., Ericsson, 2006). However, this process is covered at a very high level and does not reach the granularity of deciding to apply some new given approach over the old. In addition, the expertise literature is tangential to the training and transfer literature. Within the more traditional training literature we appreciate that adult learners come to their learning with a personal history (Knowles, 1984), and that this affects their outcomes from the learning event. This essential process of overcoming an existing automatic behavior to implement a new one is a central focus of the LTM.

Focusing on overcoming existing automatic behaviors fits with a broader trend in psychology: the re-emergence of interest in habits and habit change. Habits are conceptualized in many ways in the literature, but can be categorized as tics, neural networks, conditioned responses, everyday activities, routines, customs or rituals, character, or habitus (Clark, Sanders, Carlson, Blanche, & Jackson, 2007). Of these, the most important forms of habits for the present discussion are 1) conditioned responses – actions learned through reinforcement and conditioning, 2) everyday activities – things we do every day with little or no conscious thought, and 3) routines – more complex than single activities, involving sequences and combinations to create order. A second typology of habits by Southerton (2013) defines habits as either 1) dispositions, 2) procedures, or 3) sequences. Dispositions are the most important here, which are propensities to act in a particular manner when suitable circumstances arise.

Whether one takes the more macro, dispositional, or more micro response approach to habits, they fit well with the dual process model. If one takes the broader, dispositional, "habits as behavioral tendency" approach, habits are the output tendency of the dual process system, and any instantiation of an action could be driven by either type 1 or type 2 processes. This broader conceptualization of habits works because even an effortful process may arrive at the same conclusion as an automatic process. Thus, a habitual reaction in any given situation could be due to an automatic reaction, or due to a more thoughtful consideration of one's behavioral options. On the other hand, viewing habits specifically as behaviors which are in some way automatic firmly places habits as the outputs of type 1 processes. From this view, any habitual behavior in an organization is that which an employee may default to through automatic processes. It may seem that few work behaviors would fall under such habitual responses with the increasing complexity of the work world, but Susskind and Susskind (2017) argue most work acts, even by individuals in relatively complex jobs, are fairly repetitive and mundane, making them ripe for habituation. Further, the development of automaticity is essentially the development of habitual responses. It is that developed habitual response I argue we must overcome which we do not always account for explicitly in training research, and especially in considering if newly trained KSAOs will transfer back to the work environment.

Difficulties in overcoming existing automatic responses of adult learners is evident in some topics of study within the training literature. A primary example can be seen in attempts to train for implicit (automatic) racial attitudes to reduce racially biased attitudes and connected behaviors, and this point is worth some exploration as it has implications for the LTM.

Greenwald and Banaji (1995) explained that much of social behavior is driven by implicit or unconscious processes which allow individuals to take the correct actions in social situations without effortful processing. However, it makes changing social behavior difficult because many decisions in those situations occur outside of direct cognitive control. Relatedly, Wilson, Lindsey and Schooler (2000) proposed a dual model of attitudes specifying the relationship between implicit and explicit attitudes held towards an object or group. Specifically, individuals hold both implicit and explicit attitudes that do not necessarily agree with each other. Wilson and colleagues argue implicit attitudes are the product of long-term learning processes and are usually rooted in childhood experiences. Explicit attitudes may agree but are more susceptible to learning in adulthood. Which attitude determines behavioral outcomes is driven by the dual processes of cognition such that the implicitly learned attitude will drive behavior unless the individual is *given the opportunity* and resources to call on their explicit attitudes. This model also explains why the average correlation between implicit and explicit measures of attitudes tends to be low (e.g., Brauer, Wasel, & Niedenthal, 2000).

The dual nature of attitudes and cognitive processes pose problems when we attempt to change them. Wilson et al.'s (2000) argument that implicit attitudes are automatic judgements learned over long periods of time makes them habitual, suggesting there are deeply ingrained cognitive processes and structures which must be altered or overcome to cause lasting change. Changing existing habits is possible but difficult, and the longer one uses a given KSAO successfully the harder it will be to change it. In the case of implicit attitudes, such as racial attitudes, an employee is likely coming to the learning event with decades of experience using that attitude. Organizations then attempt to affect such attitudes through diversity training, which often lasts four hours or less (Kalinoski, Steele-Johnson, Peyton, Leas, Steinke, & Bowling, 2013). Thus, it should be no surprise that training initiatives to change racial attitudes generally fail to cause lasting change in explicit and implicit attitudes, as well as the outcome behaviors to which those attitudes lead (Lai, Hoffman, & Nosek, 2013; Lai et al., 2016). In most cases, at least pertaining to racial attitudes, individuals are not exposed to a strong enough shock to fundamentally alter their beginning set point and fail to maintain the hoped-for change over time and merely return to baseline tendency, in this case habit, after some period (Olenick et al., in press; Baldwin & Ford, 1988). A similar, though potentially less extreme, effect likely occurs for many KSAOs, and the LTM can account for such an effect.

Reinforcement Learning

As mentioned above, one way to view habits is as the product of the reinforcement of actions through their past successful application. This framing makes Reinforcement Learning a natural place to look for an existing learning theory to explain learning mechanisms within the LTM. Reinforcement Learning has been thoroughly researched by both psychologists and computer scientists and is both informative regarding how individuals learn and has the benefit of the level of formality and thoroughness required to form overarching theoretical frameworks (Muthukrishna & Henrich, 2019).

Psychological study of reinforcement learning dates to at least the studies of Ivan Pavlov (1927), and what is now known as classical conditioning. Pavlov's studies examined how the pairing of a stimulus and a reward could result in the later excitation of a response which had previously not been associated with the stimulus. For example, the initial presentation of a bell does not cause a dog to salivate. However, if over time food is presented in tandem with the bell, the dog will begin to salivate with the ringing of the bell alone. More formally, an initial unconditioned response (salivation) is normally paired with a natural trigger (unconditioned stimulus), but later can become a predictable response (conditioned response) to an unnatural trigger (conditioned stimulus). Over time, the dog comes to expect food at the ringing of a bell because of previous experience. Formalized versions of classical conditioning exist, such as the Rescorla-Wagner model (Wagner, 2008) which proposes, in part, that weighting of stimuli and response connections are updated when animals are surprised by outcomes (e.g., Kamin, 1969).

Another model of reinforcement learning can be found in the operant conditioning approach of Skinner (1938, 1963), or instrumental conditioning in the language of Thorndike (1898). Both study behavior-contingent reinforcement, and the subsequent effects of that reinforcement on future behaviors. Classic experiments by Thorndike include the use of puzzle boxes in which cats were placed and required to escape. The cats could escape such as by pushing a lever or pulling a string. Initially the cats would struggle and often only escaped by a chance solving of the puzzle, but their ability to escape increased as they gained more practice at performing the required action and were reinforced by being able to escape their confinement.

Computer science drew inspiration from the original research on animal learning completed by psychologists to development reinforcement learning algorithms (Sutton & Barto, 2018). The essential function of a learning agent in a reinforcement problem is to identify the best behavioral strategy, labeled a policy, to apply in a given situation to maximize the reward it receives from its environment. As an agent encounters its environment, it applies some policy available to it, and receives rewards based on the success of that policy. Over time, the agent estimates the expected value of that policy and can compare the expected values of multiple policies. Over time the agent applies more and more valuable policies to its task and improves its performance. Through this iterative action, feedback, and learning process agents can develop novel and powerful solutions to complex problems which are often more efficient and complex than those which humans develop on their own. Examples include robots navigating an environment (Sutton & Barto, 2018), and games as varied as checkers (e.g., Samuel, 1967), *Jeopardy!* (Tesauro, Lechner, Fan, & Prager, 2013), and backgammon (e.g., Tesauro, 2002).

Algorithms of varying complexity for reinforcement learning exist depending on the type of learning problem (Sutton & Barto, 2018). Regardless of the complexity of the chosen algorithm some of their essential features can be directly tied to the types of psychological conditioning described previously. For example, one of the laws of learning discovered by Thorndike (1898) was the *Law of Effect*, which states that behaviors which produce satisfying outcomes are more likely to occur again when presented with the same situation, and those which produce unsatisfying outcomes are less likely to occur again in that situation. Sutton and Barto (2018) connect reinforcement algorithms to the *Law of Effect*, writing:

> "First, reinforcement learning algorithms are *selectional*, meaning that they try alternatives and select among them by comparing their consequences. Second, reinforcement learning algorithms are *associative*, meaning that the alternatives found by selection are associated with particular situations, or states, to form the agent's policy. Like learning described by the Law of Effect, reinforcement learning Is not just the process of *finding* actions that produce a lot of reward, but also of *connecting* these actions to situations or states" (p. 358-359, emphasis in original).

Although computer science application of reinforcement learning is designed for agents in idealized environments, their algorithms are useful for understanding and modeling animal learning in psychology (Sutton & Barto, 2018), and may hold the key for understanding transfer as a learning process.

The Learning Transfer Model

Using the background of dual process theory and reinforcement learning, I propose the Learning Transfer Model (LTM) as a process theory which may account for common effects observed in transfer research. The LTM proposes that learners exit training with a new KSAO for which they must learn if it is a better fit for their work tasks than their previously used KSAOs. Once in the transfer environment, learners encounter relevant tasks and must choose which of their available KSAOs to apply. Based on dual processing, the learner will have an initial automatic response based on how habitual that KSAO is at that time. Once this initial automatic response occurs, it may be intervened upon by more deliberate decision processes if the learner can engage in such processes. However, even in cases where more deliberate processing is possible, the learner may still apply their old KSAO instead of their new one. Over time, the learner gains experience which will inform their future transfer decisions, and, with many applications, develop their new KSAO into a new automatic response. The basic outline of the LTM can be found in Figure 1.

This description represents the general form of the LTM, but to build strong theory for testing and future development, a key point of this paper is to develop a formal model. The rest of this section will be dedicated to explicating that formal model.

The backbone of the formal LTM is based on the algorithms of k-armed bandit problems, and unless otherwise noted all information presented in the following discussion is based on Sutton and Barto’s (2018) introductory text to reinforcement learning. In k-armed bandit problems, a learning agent, synonymous with an individual transferring knowledge in the current theory, attempts to choose the optimal solution from a number (k) of pre-defined behavioral options. That choice is made through estimating the long run value of each available policy through an iterated sampling and feedback process. K-armed bandits have four important components. First, each behavioral option available to the agent is called a policy. In the LTM, each agent has access to two policies representing their pre-training KSAO (*Policy A*) relevant to the theoretical work situation targeted by the training intervention, and the organizationally-introduced KSAO relevant to that situation (*Policy B*). The assumption that only two policies are of interest for transfer questions makes the approach used here a 2-armed bandit problem. Second, each policy has a reward function, or true value, which dictates the distribution of rewards the agent receives when the agent chooses to apply that policy. Third, estimates of the value of the policy which represents the predicted reward of the Policy According to the agent's experiences applying that policy. Thus, the agent is estimating the reward of each policy and attempting to discover the best policy to apply at each time step. Fourth, the agent does not always exploit the policy which it currently deems the most valuable, and sometimes explores other potential policies instead. The inclusion of a minor amount of exploration terms such methods as *E-greedy*, where the agent greedily exploits the current most valued policy but explores with some rate of error.

Several important aspects to this approach to reinforcement learning are worth mentioning. First, agents learn based on the evaluation of actual actions they take, not from instruction by outside entities. This is one point which separates the current model from the CLARION (Sun et al., 2005) model previously discussed. Second, learning in such agents is limited to a single, unchanging situation. That is, the value of each policy is fixed because the environment to which they are applicable is unchanging. Sutton and Barto (2018) describe such approaches as non-associative, where the agent does not need to choose which policy to use in different situations. There are more sophisticated reinforcement learning approaches that can be applied to changing situations, but these are more complex than necessary at this stage of developing the LTM but could be utilized in the future to study adaptive transfer. For now, we will assume the transfer situation is stable enough to apply their newly learned policy. Third, the k-armed bandit approach assumes that the goal of the agent is to maximize the long-term value of their actions. Fourth, events in bandit problems are episodic as opposed to continuous. Finally, the reward received by the agent at each episode is randomly chosen from a stationary distribution of the rewards associated with that policy.

The application of k-armed bandits to humans in transfer environments requires at least three other assumptions to be made. First, individuals/agents exit their learning experience with the ability to apply the targeted KSAO represented in their new policy. This assumption suggests that this model is currently more applicable to maintenance than generalization within the transfer space. Second, the learner will not alter the given policy to fit their own needs once that policy is created. Third, the agent must possess perfect recall of their experiences when attempting to apply the available policies in order to accurately calculate the expected value of that policy.

Given this background, we can fully describe the formal LTM and outline how a computational instantiation of that model would operate. Agents, synonymous with learners from here forward, are presented with an abstract task at each time point. For our purposes, the task does not actually matter and will remain undefined, other than that the task is such that the agent can be successful or unsuccessful only. The probability of success on any given attempt is defined by the policy which the agent chooses in that attempt and is equal to the true value of that policy. For example, if a given policy has a true value of .80, the agent will have an 80-percent chance of succeeding on the task when applying that policy. In the computational version of the model, success on an attempt will be determined by a random draw from a uniform distribution from 0 to 1, with any number below the true value of the policy being considered successful. If successful, the agent is rewarded with 1 point, otherwise it receives 0. In this way, the mean of a large enough sample of rewards received by the agent will approximate the true value of the policy. The true value of Policies A and B will be represented by the variables R_a and R_b respectively. The random component here adds a crucial stochastic element to the model (e.g., Railsback & Grimm, 2012) making the model non-deterministic, and Monte Carlo simulation important for exploration. This stochastic component represents the idea that any given task attempt in one's work environment is essentially a random draw from all possible attempts of that task.

As an agent attempts its task it must estimate the value of its policies. Estimated values for a policy are a dynamic process by which the estimation of the value at any time t + 1 is a function of the value estimate at time t, the difference between the expected value and reward on a given application of the policy, and a step-size parameter that defines the rate of learning for the agent. The essential framework for reinforcement algorithms follows this framework of

$$NewEstimate \leftarrow OldEstimate + StepSize[Target - OldEstimate]$$

where Target is the reward at a given time step (Sutton & Barto, 2018). In k-armed bandits, those value estimates can be obtained through action-value methods, which use the experience of the agent to drive the estimation. A simple calculation of the value estimate is to average the received rewards up to that point in time, thus:

$$Q_t(a) = (\text{sum of rewards when } a \text{ taken prior to } t)/(\text{number of times } a \text{ taken prior to } t)$$

where $Q_t(a)$ is the value function for Policy A. A more sophisticated way to track the value estimate is as a function of the *n*th reward:

$$Q_{t+1}(a) = Q_t(a) + \frac{1}{t_a}[R_{t|a} - Q_t(a)]$$

Algorithm 1. Value Estimate Calculation

where the expected value of Policy A at step t + 1 is a function of the estimate at step n plus a weighted function of that prior estimate and the received reward R_t at that time. Estimating values this way defines that value as a dynamic process underlying the primary transfer decision process in this model (Dishop et al, in press). In addition, this equation defines the learning rate as the inverse of the number of steps taken, meaning learning will decrease over time, fitting with the power law of learning (Newell & Rosenbloom, 1981). The above equation provides a learning agent's estimate of the policy's value as it develops over time, but the agent also can be given an initial estimate of that policy. The initial values given to an agent can affect the behavioral decisions of that agent over time and can improve long term results under certain conditions (Sutton & Barto, 2018). In the LTM, that initial estimate can be defined by $Q_1(a)$ and $Q_1(b)$ for Policies A and B respectively.

Tracking the expected value of each policy is only part of the agent's learning process. Whenever the agent encounters its defined problem the agent must choose which policy it will apply. Typically, this occurs through action-value methods of selection, where the chosen policy is the one with the highest estimated value $Q_t(a)$ or $Q_t(b)$. Let P_t represent the policy the agent chooses at a given time point. By choosing the highest value policy the agent is choosing the policy which it believes will offer the greatest reward at that time point. However, always choosing the highest value policy does not allow the agent to effectively test other potential solutions. Instead, the agent can be allowed to explore other policies not currently seen as the most valuable to find other, potentially better, policies. This is the classic exploration versus exploitation choice seen in studies within the organizational literature (e.g., March, 1991). The rate of exploration can be defined by a variable E; this approach is referred to in reinforcement learning as an *E-greedy* method (Sutton & Barto, 2018). In the transfer case of choosing between two possible policies, the agent's choice at any time t is defined as the greater of the two value functions $Q_t(a)$ and $Q_t(b)$ with some probability $1 - E$.

The process described so far is very rational on the part of the agent. However, not all choices by individuals are so clearly logical. The form of choice outlined thus far more closely aligns with type 2 processing systems from dual processing, however type 2 systems are not always engaged and are theorized to intervene, or not, in decisions already made by type 1 processes (e.g., Evans & Stanovich, 2013). Thus, we must expand the LTM to include an initial automatic decision and learning process to represent type 1 processes, and a mechanism to determine if the type 2 processes will intervene in that decision.

The type 1 process hypothesized here is based on the number of times a policy has been applied. This is the idea that repetition leads to automaticity and that the more times a stimulus and response are paired, the more likely they are to be activated in the future. Let $Z_t(a)$ be the probability of choosing Policy A over B. $Z_t(a)$ is a function of the number of times that policy has been chosen out of potential times it could have been chosen from A and B. In addition, the agent in a learning transfer context will likely have experience with their Policy A prior to entering the learning event where Policy B is introduced. Thus, the agent should already have some value estimate of that policy based on their experiences and an associated number of times they have applied it. However, it is also possible that the agent receives some actual experience with their new Policy B prior to entering the transfer environment, such as in the learning event itself. To account for those applications let L represent the number of practice attempts the agent has had with the new Policy B. The calculation of $Z_t(a)$ is then:

$$Z_t(a)|T_1 = \frac{\sum P_t(a)}{\sum P_t(a) + \sum P_t(b) + L}$$

Algorithm 2. Type 1 Process Equation

The default choice of the agent at time t is Policy A at the rate $Z_t(a)$, and b at the rate $1 – Z_t(a)$.

Once the type 1 process has chosen a policy, it is then up to a type 2 process to intervene. However, they do not always do so because they are not always able. For example, the agent may not have the necessary resources, whether those resources be cognitive, or exterior to the agent such as time. It would be possible to theorize about the specific effects of various factors that may affect the likelihood of employing type 2 processes. However, for simplicity the present model will cover all effects in a percentage chance that type 2 processes are implemented. The chance of engaging in type 2 processing at any time point will be defined as S_2 and ranges from 0 to 1. If the agent engages in type 2 processes, then the decision process outlined previously is utilized which may or may not result in the same decision arrived at by type 1 processes, which refines the policy choice of type 2 processes to be:

$$P_t|T_2 = \max[Q_t(a), Q_t(b)]$$
$$with\ probability\ 1 - E$$

Algorithm 3. Probability of Choosing Type 2 Processes

Which represents the policy P chosen at time t given type 2 processing is the maximum value of policies a and b with a likelihood dependent on the amount of exploration desired, E. If the agent does not apply type 2 processes, the type 1 decision is utilized. In either case the agent updates relevant equations based on the outcome of their action and moves on to the next attempt.

All parameters and equations for the model can be found in Table 1 and Table 2 respectively. It is important to note that almost all aspects of this process could draw on more complicated conceptualizations of that individual theory, however that is not the point of starting a modeling cycle in an area that has never been covered before. The present model should be viewed as a building block for future theoretical development.

Study 1: Method

The described model above was instantiated into an agent-based model using the simulation program NetLogo (Wilensky, 1999). Although NetLogo does not offer as much flexibility as other programs such as R, NetLogo is a specially designed platform for implementing agent-based simulations. Although only a single agent is being studied in the present model, utilizing this platform allows for easy expansion into later iterations to examine multiple agents in networks, teams, organizations, etc. The equations outlined above were used to determine the learning and behavior of the agent modeled over time. A snapshot of the modeling environment and the code for use in NetLogo are available in Appendix A, and a copy of the program itself are available from the author upon request.

Model outcome metrics

To analyze the potential of the model to account for the important training effects described above, two primary outcomes were chosen to track within the modeling environment. Much has been written about what aspects of training outcomes are important to measure to describe training success. Kirkpatrick's (1994) classic typology describes important outcomes at four levels: reactions, learning, behavior, and results. Much research in organizations is limited to reactions to training, despite reactions being probably the least informative. Other emphasis has been placed on cognitive outcomes of training, such as learning, which has driven much research over the last couple of decades (Kraiger, Ford, & Salas, 1993; Ford, Kraiger, & Merritt, 2010). These two levels of outcomes have implications for the present models. Utility perceptions are a type of reaction to training, but learning outcomes take a background role in the LTM as the agent having successfully learned the new policy is an assumption made for simplicity.

To measure important outcomes in the modeling for this paper, a reemphasis must be placed on measuring behavior and outcomes. The shift in emphasis to focus on cognitive outcomes of training moved the field from a focus on behavioral change (Kraiger & Ford, 2007), but these outcomes are focused on effects emerging from the training event itself. The study of transfer of those learning outcomes to on-the-job behavior is an area of needed research (Ford et al., 2010) and the present model is intended to help describe the process of that transference. Behavior and performance outcomes of the agents in the models then become the key variables of interest. A behavioral measure was created as the percentage of time the target policy, Policy B, is implemented by the agent. Measuring behavioral choice outcomes in this way also aligns with definitions of learning which focus directly on behavioral change (e.g., Myers, 2004). In addition, performance of the agent was tracked over time and was defined as the percentage of times the agent successfully completes its abstract task. Additionally, each agent stored their performance after an initial burn-in period which represents the pre-training phase and is a time when the agent can only apply its first policy. The agent then stored their performance at the end of the defined transfer period both for their overall performance and their performance just within the transfer period. These variables were defined as types of "success" as performance in this model is equal to the percentage of time the agent successfully completes its task. Further, saving performance both pre- and post-training allowed the model to be analyzed as a pre-post intervention design, providing for greater insight into the causal effects of adding a defined second policy to the agent's decision tree. Further, doing such a pre-post performance comparison aligns with our adopted definition of transfer as "the extent to which the learning that results from a training experience transfers to the job and leads to meaningful *changes* in work performance" (Blume et al., 2010, p. 1066; emphasis added). This will be accomplished via calculation of Cohen's *d* for conditions comparing pre-training and post-training performance, allowing for both easier comparison to existing effect sizes in the research literature, and placing results into a standardized metric to help correct for any idiosyncrasies that may make the interpretation of raw effects misleading.

Analysis

Analysis of computational models does not follow the typical procedure of empirical research. Instead of testing traditional statistical models, testing of the LTM followed common cycles of computational model exploration (e.g., Railsback & Grimm, 2012). Important steps include verification, showing generative sufficiency, and exploring sensitivity and robustness. Verification includes confirmation that the implemented model is consistent with the proposed theory (Banks, Carson, Nelson, & Nicol, 2010). This was accomplished via logical consistency checks by the author, and testing of the mechanisms of the model to ensure basic relationships expected occur when the model is executed. Generative sufficiency entails confirming that the model can recreate general effects known from real data. Achieving generative sufficiency does not confirm that the proposed model is *the* explanation of the process being studied, but it does confirm the model is a *possible* explanation of that process (Epstein, 1999). Finally, sensitivity and robustness entail an exploration of the model parameters to determine how sensitive the model is to changes in initial conditions and violations of assumptions (Railsback & Grimm, 2012), which achieves three goals. First, it is not clear at which levels of various parameters common effects seen in the literature may manifest, exploring the model allows the tuning of parameters to more accurately reflect reality. Second, model exploration allows the discovery of potential discontinuous effects of parameters where the results of the model change rapidly as initial conditions for that parameter change. Third, it may reveal unexpected or interesting findings, which is not the goal of the model but can be useful for providing insight to real world phenomena or guide future research. All four of these steps were executed and will be outlined below.

Since computational models create simulated data, output statistics need to be interpretable without traditional significance tests because they lose meaning when you can simulate as much data as you desire and you program in most primary effects (e.g., Railsback & Grimm, 2012). Instead, we must use summary statistics and correlations to describe effects of interest. We can use these to calculate effect sizes of parameter changes on model outcomes and compare these to meta-analytic effects. Another key tool in the computational modeler’s kit is heat maps, which can provide visualizations parameter effects which are easily interpretable and can show transition points in model parameters that drastically impact model outcomes. The same basic approach was utilized for analyzing all models in this paper.

Study 1: Simulation and Results

In this section I will outline simulations directed at the four main steps in exploration: verification, generative sufficiency, sensitivity, and robustness checks.

Model verification

Prior to beginning any simulation, the model was subjected to a series of verification checks (Banks, Carson, Nelson, & Nicol, 2010) outlined here.

Logical Consistency

The model was executed in NetLogo as outlined in the theoretical development and research methods. The only alteration of the theory made for computational efficiency was to code the relationship between type 1 and 2 processing slightly differently than the default-interventionist approach outlined in the theory. Instead of making a default decision then choosing if the agent will use their type 2 processes to intervene, the agent decides if they will use their type 2 processes or not first. If not, then they make a habitual choice and implement it, if they do use their type 2 processes, they make their more rational decision as if they were intervening in an existing but now inconsequential habitual reaction. Although the code is not strictly default-interventionist, its outcomes should be identical while avoiding the computational inefficiency of performing a default judgement when it would be overridden anyways.

Parameter Effects Check

Once implemented in NetLogo, a series of tests were run to ensure that when parameters were adjusted, corresponding and expected changes occurred within the model. The following outlines a series of tests to show the adjustment of each parameter corresponds with the desired effects.

Simulation Length

The first test confirmed the desired lengths of the pretraining and post-training simulations. Parameters besides the pretraining and transfer for these simulations are of no consequence and were held at constant levels. To test the length of the pretraining periods, one simulation each was run with a length of 250- and 500-time steps with no transfer time allowed. These returned the 250- and 500-time step lengths expected. A similar test was then completed with transfer lengths of 250- and 500-time step lengths but no pretraining period. These again returned the expected lengths to 250 and 500.

Policy Value

To test the effect of the true value of the policies, the success rates of the policies were checked across a series of simulations. To check the value of Policy A, the simulations focused on the pretraining period because only Policy A is available to the agent. Simulations were run for 500-time steps, with true policy values of .50 and .75. Success rates for these simulations were .50, and .76. Given this was a single simulation, this confirms the expected effect of the success rate of Policy A.

A test of the value of Policy B is more complicated because it was only available in the transfer environment. To isolate the effect of Policy B, the value of Policy A was set to 0, and no pretraining time was allowed. In addition, no exploration was allowed and Type 2 thinking always employed. This should force the agent to apply Policy B alone. Values of Policy B were tested at .50 and .75, with 1000 transfer attempts. True success rates in these conditions for a single run were .51 and .75, in line with expectations.

Policy Value Estimates

Two tests were completed to check the veracity of policy value estimates, corresponding to initial and final estimates. Initial estimates should correspond to the set initial value estimate for the defined policy, such as .50 or .75. To check this, models were executed and the value of the policy estimate at the first time point was verified to be equal to the value set for the simulation.

Additionally, at the end of the simulation, we should expect value estimates to approximate the true value of the underlying policy representing an accurate judgement on the part of the agent under ideal conditions. To assess this, models were run to isolate the effects of both Policy A and Policy B at levels of .50 and .75. For Policy A, only pretraining time was allowed, run for 500 steps (the maximum allowed in the simulation). The model was run 10 times at each level. For these 10 runs, results ranged from .472 to .546 with a mean of .51, and from .742 to .802 with a mean of .769 for the .50 and .75 levels respectively. For Policy B, the transfer environment was isolated and run for 1000 steps, the maximum allowed in this simulation. For these runs, the range for the .50 policy was from .487 to .518, with a mean of .50, and for the .75 policy the range was .734 to .761 with a mean of .75.

Exploration Rate

To check that type 2 thinking processes are willing to explore at a defined rate, the simulation was set up with a value for Policy B at 0, and Policy A at 1, and a 100% chance of Type 2 processes engaging. This should result in Policy A being chosen nearly every task attempt, except for a rate approximating the defined exploration rate. This simulation was run 10 times with an exploration rate of 10%. These simulations ranges from .088 to .106 in regard to rates of choosing Policy B, with a mean of .096. This is in line with the expected value of .10.

Given the results observed in these checks, it appears the simulation is operating as expected.

Generative Sufficiency, Sensitivity and Robustness

Following model verification, a series of experiments were conducted to assess the model for generative sufficiency. This section outlines the attempts to determine if the model could generally account for existing findings in the training and transfer literature. Due to the nature of the experimentation, the model was essentially simultaneously checked for sensitivity and robustness as parameters were tuned to better represent naturally observed phenomena. To accomplish this, parameters were manipulated initially via coarse sweeps of the available space for the parameter of interest, holding all other parameters constant, to determine the effects of the parameter and to ensure that the model code reliably changes the levels of parameters (which is in some ways a continuation of the verification process). As modeling proceeded, experimentation became iteratively more complex and focused on potentially interesting facets of the model in a way guided by the emerging findings of the modeling process. In addition, though generally desired end results were known from meta-analyses, little if any guidance exists on how strong a manipulation is from a mathematical standpoint to determine the size of manipulation to make in the experimental code *a priori.* Therefore, initial exploration aimed to tune the model parameters to create reasonable transfer outcomes, for example obtaining Cohen's *ds* on pre-post measures of performance on .3-.5, and not exceedingly large effects such as 2 or more.

True policy values

The first set of models aimed to tune the model into a reasonable parameter space regarding the values of both Policy A and Policy B. Defining policy values that are both representative of the type of tasks in which we may be interested in the real world, and the separation in policy values which will reproduce reasonable transfer effects are important considerations. For example, if we were interested in improving baseball batting skills, the success rate of each policy should be very low, such as .25 to approximate the batting average of Major League Baseball players. On the other hand, success rates for performing well defined tasks on an assembly line are likely .95 or higher. Most closed skills in regular organizations probably exist at this high end of the value continuum, but open skills may be much lower.

Two slightly different ways of parameterizing the policy values were explored here. In the first version, the true policy values of A and B were independently set. Based on the above discussion of the possible range of relevant values, the true values of both Policy A and B were swept from 0 to 1 in .05 increments, fully crossed with 500 replications each. Runs were a 250 burn-in and transfer period, exploration rate set to 10%, system 2 activation 50%, and initial policy value estimates set to .5. To analyze the results heat maps were generated of the effects on behavioral transfer rates, and pretraining-post-training changes in performance as measured in Cohen's *d*. Results for behavioral transfer and performance change can be found in Figures 2 and 3 respectfully.

In examining these results, we can see that behavioral transfer rates range from about 5% to about 55%. Low numbers make sense given effect of habitual response and time allowed for Policy B to override that previously habitual response. This low rate of transfer also aligns well with expectations given the low amount of transfer commonly cited in the research literature (Ford et al., 2011). Performance change also shows the generally expected pattern. We see a diagonal where when policy values are equal lead to no performance change, as expected. There are negative performance changes below that diagonal representing training a policy that is less valuable than the existing policy, and positive values above the diagonal representing improvements of the new policy over the old. In addition, improvements appear to be stronger than corresponding decrements, which makes sense because agents should abandon the new policy if they do not see it as an improvement. The sudden change in magnitude of effects across this diagonal suggest a sensitive area of the model where values change suddenly and dramatically. Changes in terms of Cohen's *d* range from -3.28 to 11.69. Obviously, the upper and lower portions of this range are well outside of what we might expect in the training literature, indicating that some areas of policy values are essentially *out of bounds* regarding their ability to replicate reality. However, along the diagonal where values of Policy B are just barely higher than the values of Policy A, we see performance effects of about $d = .30$, indicating that when the value of Policy B is slightly greater than Policy A the model is able to reproduce the essential effect of training we expect from research experience.

Although this initial result is promising, the way these parameters are defined limits the ability to vary policy values along with other variables in the future while maintaining low enough dimensionality that the results may be interpreted. Thus, the decision was made to redefine the true value of Policy B in direct relation to the true value of Policy A. This was accomplished by inclusion of a parameter indicating the *change* in the true value of the policies moving from Policy A to Policy B. So, for example, if Policy A was given a true value of .50, and the change in policy value defined as .10, Policy B would have a true value of .60. In the first set of models we saw that what appears to matter in making the model a plausible representation of the real world is Policy B having a slightly greater value than Policy A. By reconfiguring the model to define Policy B in direct relation to Policy A, we can better home in on a difference between the two policies which best represents reality.

In the updated model, simulations were run sweeping Policy A's value from 0 to 1 in .05 increments, with policy value change swept from -1 to 1 in .05 increments. Runs were a 250 burn in, exploration rate set to 10%, system 2 activation 50%, and initial policy value estimates set to .50. 500 replications of each were run. Behavioral transfer and performance change results can be seen in Figures 4 and 5 respectively. Results show little behavioral transfer when the policy change is negative. This is expected and indicates agents are discarding the new policy except for a small amount due to the exploration factor when the new policy is worse than their old one. This is a pattern that we would hope to observe in the real world as we would not want to have employees using a worse behavior if they do not have to. Overall, transfer rates range from about 6% to 55%. Interestingly, some nonlinearity appears to be occurring where the highest transfer happens when policy values start initially low and change a lot (as would be expected), but when policies start low and only improve a little transfer actually does not occur as much as when policies are already valuable and change upwards a little bit. Transfer rates of about 30% run in a line along a change of .40 when Policy A starts at 0 to .10 when Policy A starts at .70.

We see similar patterns in performance change. There are slight performance decrements when the new policy is worse than the old as we would expect because a worse policy does get applied sometimes. We also see expected incredibly high performance improvements when the existing policy is low and the new one is high. However, higher *d*s are seen with higher starting policies in many instances as with the above behavioral change. The kind of effect sizes we tend to see for performance improvement, or at least expect, occur in a similar diagonal to behavioral transfer where it requires less improvement when prior policy values are higher. A great example is at a Policy A being .70 and an improvement of only .05 is a *d* of .34. Given the convergence of behavioral transfer rates and performance improvement to reasonable ranges when Policy A is .70 and policy change is .05 these values were selected for use in further modeling efforts.

Timing of interventions

With a defensible level of policy values to use to define the model, it was also important to explore the effects of pre-training and transfer times on the model to ensure proper time was allotted for each. History with Policy A represents the length of time the agent applied that policy before the introduction of Policy B. That history was modeled by a burn-in period where the only policy available to the agent is Policy A. Exploring timing of an intervention importantly accounts for the history adult learners bring to their learning events (Knowles, 1984), and begins to account for overcoming established automatic responses once that learner returns to their work environment. It was expected that longer periods of time where the agent could only access their Policy A would result in reduced transfer of Policy B as A will be more likely to be activated by type 1 processes, and this effect will hold longer in the face of repeated application of Policy B.

To explore the effects of training and transfer time simulations were run sweeping burn-in and transfer time each from 25 to 500 time points in 25 step increments. As discovered in the above simulations as an interesting and applicable level of policies Policy A was set to a reward of .70 and the policy change to B at .05. Exploration rate set to 10%, system 2 activation 50%, and initial policy value estimates set to .50. 500 replications each. At the condition level, pretraining (burn-in) time was correlated with behavioral transfer at $r(200000)$[1] = -.48 ($p < .001$),

[1] Sample sizes, degrees of freedom, and significance have been reported for statistical analyses, but it should be reiterated that traditional interpretation of significance holds no meaning in the context of computational models. Sample sizes and associated degrees of freedom for statistical tests are arbitrary when one has control over the modeling environment as more data can always be simulated. Therefore, readers should focus on reported effect sizes and interpret any associated significance conclusions with extreme caution. See Cumming (2014) for further discussion of the limits of null-hypothesis significance testing and move towards the use of effect sizes to improve research in general.

and performance change at $r(200000) = -.31$ ($p < .001$), indicating less transfer when the agent had used its old policy for longer prior to training, as expected. On the other hand, transfer time was related $r(200000) = .80$ ($p < .001$) to behavioral transfer, and $r(200000) = .58$ ($p < .001$) to performance change, indicating that the longer the agent had to attempt transfer the more likely they were to do so.

To further examine these effects, behavioral transfer and performance change (in Cohen's *d*) were plotted in heat maps, these can be found in Figures 6 and 7. In these depictions, we see that earlier training improves transfer rates. There also appears to be a possible augmentation effect where the combination of early training and a long time to adopt the new behavior leads to much greater transfer rates. Interestingly, it is also apparent that pre-training time quickly overwhelms the effect of longer transfer time. From these results, a burn-in of 100 with a transfer of 500 might be reasonable to use for future exploration of other parameters to produce transfer and performance improvement levels commensurate with real-world levels. These results also seem to be suggesting that although the present process may be a good approximation for a possible transfer process, the effect of habits might be too strong. But this could be caused by either the habit process itself, or the low level of agent ability to engage in type 2 thinking. As such, that was explored next.

Type 2 Processing

Next, the effect of being able to engage in type 2 processing was examined independently of other variables. Here, a greater ability to engage in type 2 processes equates to a greater opportunity to use their new skills, or a situation strength where they are free to make that transfer choice more independently. For these, Policy A value was held at .70, Policy B as .05

better, 100 burn in time, 500 transfer time, .10 exploration rate, .50 initial value estimates. Likelihood was swept from 0 to 1 at .01 intervals. 500 replications were chosen for resolution.

Because only one variable was being examined here, a slightly different approach was utilized to examine the results. First, a correlation between type 2 likelihood and behavioral transfer at the replication level revealed a relationship of $r(50500) = .51$ ($p < .001$). As the likelihood of engaging in type 2 behaviors has been argued here to be akin to the opportunity to use one's training and other environmental factors, the comparable effect size was around .30-.40 (Blume et al., 2011). This suggests the model was able to essentially replicate the expected pattern of results. For further analysis, instead of heat maps, a linear regression was utilized to examine both linear and curvilinear relationships between the likelihood of engaging in type 2 processing and behavioral transfer and performance change. Through this analysis it was found that behavioral transfer was predicted by type 2 likelihood, at the condition level, at a linear rate of .756 ($\beta = 1.43$, $t = 58.253$, $p < .001$), and curvilinear rate of -.233 ($\beta = -.46$, $t = -18.520$, $p < .001$)intercept was -.016 ($t = -5.848$, $p < .001$; $F(2, 98) = 12949.35$, $p < .001$, $R^2 = .998$). Additionally, Cohen's *d* of performance change was predicted from type 2 likelihood at a linear rate of .973 ($\beta = 1.28$, $t = 10.888$, $p < .001$), and curvilinear rate of -.249 ($\beta = -.34$, $t = -2.882$, $p = .005$; intercept of -.292 $t = -15.117$, $p < .001$; $F(2, 98) = 519.44$, $p < .001$, $R^2 = .96$). Graphs of predicted and observed behavioral transfer and performance change can be found in Figures 8 and 9. From these you can see that as type 2 likelihood improves, so does behavioral transfer and performance change. Interestingly, performance change displays a negative effect at low levels of type 2 likelihood, but turns positive once likelihood is above about 33%. In addition, a likelihood of .80 seems important as there is a spike in performance improvement and transfer rate improves to near .40. Due to this effect, further models used likelihoods of .80 unless stated otherwise.

Practice and Overlearning

We know that practice with a new skill can improve transfer outcomes, especially when that skill is practiced to the point of overlearning. of the comparable meta-analytic effect of overlearning is .298 (Driskell et al., 1992). Although the Driskell et al. meta-analysis uses retention as an outcome, retention may be approximated by whether the agent is still applying the new policy at the end of the simulation run. The effect of overlearning in the present model was tested by the manipulation of the level of *L* from 0 to 200 in steps of 25. The high end of 200 was chosen because it represents up to twice number of pretraining task attempts. For these simulations Policy A was set to .70, change in value to .05, system 2 activation .80, 100 pretraining time steps, 500 post training time steps, and exploration to .10, with 500 replications of each condition.

When examined at the replication level, that is individual agents, we see a correlation between practice attempts and behavioral transfer is $r(5500) = .118$ ($p < .001$), and the correlation with post training performance (so only performance after the training event) is $r(5500) = .074$ ($p < .001$). These relationships are in the expected direction, but substantially lower than the comparable meta-analytic effect. For further analysis, condition-level results were calculated for behavioral transfer and performance-change. These results can be found in Table 3. There, you can see that there is a clear benefit to practice as we would expect, though it is difficult to tell the strength of the effect. To remedy this, the data was reanalyzed as a series of experiments comparing a control condition with no practice attempts to ever increasing amounts, with differences expressed to the control condition in Cohen's *d* for both behavioral transfer and performance change in Table 4. Through these analyses we see that performance improvement remains relatively low, even in stronger conditions, while behavioral transfer improves quite substantially as practice increases. Despite this, it is evident that although the general positive effect of practice is obtained, it may not be easily tuned to better approximate typical research findings.

Utility reactions

Utility reactions are a learner's perception regarding the usefulness of their learning experience (e.g., Ruona et al., 2002) and are strong predictors of transfer (Blume et al., 2010). Utility reactions can be equated to the initial value estimates of a learning agent in reinforcement models. For example, Sutton and Barto (2018) use "optimistic initial values" to encourage exploration by the agent. In the same way, a transfer agent in the LTM that has a higher initial expectation of the value of their new policy should be more likely to transfer that policy because they are willing to explore its potential. To test this effect, the model was explored by varying initial policy value estimation for Policy B from 0 to 1 in .05 steps. Policy A was set to .70, change in value to .05, type 2 activation .80, 100 pretraining points, 500 post training, exploration .10, and 500 replications per condition.

At the replication level, results reveal almost no relationship between initial value estimate and our outcomes of interest. Specifically, the relationship between initial value estimate and behavioral transfer was on $r(10500) = .02$ ($p = .025$), and only $r(10500) = .01$ ($p = .254$) with post training performance. These results are not in line with what was hoped for regarding existing effects of utility reactions. One possibility is that there was too much noise at the individual level regarding outcomes, so results were also examined at the condition level. There, the relationship between initial value estimates and behavioral transfer was $r(21) = .476$

(p = .029) and was $r(21)$ = .570 (p = .007) for performance change. The condition level results for this exploration can also be found in Table 5.

Transfer trajectories

A subset of models was run to examine the development of transfer over time to explain the development of the various trajectories described by Baldwin and Ford (1988). To examine these trajectories, models were run using the baseline parameters of Policy A value of .70, change in policy value of .05, type 2 activation of .80, 100 pre-training, and 500 post training time points. The goal here is only to show the transfer trajectories described by Baldwin and Ford (1988) are possible within this model. Thus, the model was run several times, examining the shape of the behavioral transfer rates within the modeling environment. Some examples of transfer trajectories from the model can be seen in Figure 10A-D. These examples show a variety of transfer trajectories, such as (A) initially high levels of transfer and later tapering off; (B) initial failure to transfer with later increased transfer; (C) immediate and consistent transfer; and (D) a general failure to transfer.

Implementation Intentions

As discussed, implementation intentions are used to establish an automatic link between situation and response to improve the automaticity of that response (e.g., Gollwitzer, 1999). Although also impacting the automaticity of applying Policy B, implementation intentions are not the same as practice or overlearning which is already included in the model. To account for the improved automaticity brought by implementation intentions let us instead define a variable I as the percentage increase in chances of applying Policy B when type 1 processes are enacted. This changes our calculation of $Z_t(a)$ to be:

$$Z_t(a) = \frac{\sum P_t(a)}{\sum P_t(a) + \sum P_t(b) + L} - I$$

Algorithm 4. Type 1 Process with Implementation Intentions

We can then manipulate the level of *I* to explore the effects of implementation intentions. This tweak was coded into the model for exploration. Due to the underlying math behind the simulation, when the agent has no history of engaging in Policy A, the likelihood of doing so when only Type 1 processes are available should be equal to 0 minus the defined level of implementation intentions. This was verified with a level of implantation intentions at .10, which returned a simulated critical value for automatically applying Policy A of -.10, as expected.

Implementation intentions were explored from 0 to .50 in .05 increments, 500 replications each. Likelihood of type 2 processing was set to .80, Policy A value was held at .70, Policy B as .05 better, 100 burn in, 500 transfer, .10 exploration rate, and .50 initial value estimates. From these, the replication-level correlation between implementation intentions and post training performance was $r(5500) = .111$ ($p < .001$) and was $r(5500) = .193$ ($p < .001$) with behavioral transfer. Condition-level results for this experiment can be found in Table 6, which shows a steady improvement in behavioral transfer as implementation intentions increase. However, the effect on performance improvement is much less consistent.

Exploration rates

It is typical in both reinforcement learning problems (Sutton & Barto, 2018) and in organizational research (e.g., March, 1991) to have an exploration parameter in the model. Within this model, exploration represents the degree to which an agent is willing to explore behavioral policies that they do not currently see as their most valuable. In the real world such exploration would be akin to an employee searching for a better way to do their job than their current dominant approach. Typically, there is a trade-off between exploration and exploitation for overall performance where some degree of exploration is beneficial but too much can hinder performance (e.g., March, 1991). One possible implication of this model is identifying a degree to which trainees should be willing to explore new task approaches in order to maximize their performance. To explore this possibility, while holding all other parameters constant, the exploration parameter was swept in .01 increments from 0 to 1.0.

In examining this simulation, we find a negative overall relationship between exploration and behavioral transfer ($r(50500) = -.368$, $p < .001$) and post training performance ($r(50500) = -.168$, $p < .001$). Similar results were seen at the condition level with behavioral transfer ($r(101) = -.600$, $p < .001$) and performance change ($r(101) = -.514$, $p < .001$). Such relationships are initially surprising as it was expected that a willingness to explore would allow the agent to find more optimal solutions. To understand this relationship better a regression was run examining both the linear and curvilinear effects of implementation intentions on behavioral transfer and performance change. In doing so, we find exploration to have a linear relationship with behavioral transfer of .882 ($\beta = 2.08$, $t = 13.333$, $p < .001$), and a curvilinear relationship of -1.136 ($\beta = -2.77$, $t = -17.754$, $p < .001$; intercept of .322, $t = 22.492$, $p < .001$; $F(2,98) = 273.85$, $p < .001$, $R^2 = .92$). With performance change we see a linear relationship of 1.335 ($\beta = 2.16$, $t = 10.757$, $p < .001$), and curvilinear relationship of -1.653 ($\beta = -2.76$, $t = -13.767$, $p < .001$; intercept of .145). In addition, you may find predicted and observed values of behavioral transfer and performance change in Figures 11 and 12. These results show the effects of exploration peak at some moderate level and further exploration proves detrimental for transfer outcomes.

Exploratory experimentation

Finally, one strength of building a computational model of a proposed theory lies in the ability to execute virtual experiments which can guide future real-world data collections. This allows us to test novel moderations or interventions which would be difficult to justify spending the resources on to test in real data collections without some prior empirical guidance. In addition, it could lead to the discovery of novel interactions which can lead to targeted data collections to help further support or refute the veracity of the proposed theory. Given the positive relationships found in the present model between implementation intentions, type 2 likelihood, and our studied outcomes, a virtual experiment was designed to test the mutual effects of implementation intentions and type 2 likelihood on those outcomes. Given their independent positive effects on behavioral transfer and performance change, it was expected the two would have an augmenting effect where high levels of both would result in the highest outcome levels.

To explore this possibility, a virtual experiment was designed where the parameters for both implementation intentions and type 2 likelihood were swept from 0 to 1.0 in .05 increments, fully crossed with each other. The other parameters were held constant at the levels settled on above: 100 pre-training burn-in time periods, 500 post training time points, Policy A value of .70, change in value of .05, exploration rate of .10, and initial policy value estimates of .50. To explore these effects, both a more traditional multiple regression approach to analyzing interactions and heat maps were employed. Predictors were mean centered prior to estimating the regression and an interaction term created from the products of our two predictors. In predicting behavioral transfer, it was found that type 2 likelihood ($F(3, 2020496) = 57700.917$, $p < .001$, $R^2 = .663$, $b_0 = .646$ ($t = 1457.75$, $p < .001$), $b_1 = -.217$, $\beta_1 = -.236$ ($t = -148.16$, $p < .001$)) actually had a negative main effect, but intentions had the expected positive ($b_2 = .496$, $\beta_2 = .539$ ($t = 338.45$, $p < .001$)) main effect, along with a negative interaction effect ($b_3 = -.925$, $\beta_3 = -.305$ ($t = -191.32$, $p < .001$)). A similar pattern of results was found in predicting post training performance (b_0 = .732 (t = 14774.36, p < .001), b_1 = -.011, β_1 = -.133 (t = -67.09, p < .001), b_2 = .025, β_2 = .302 (t = 152.50, p < .001), b_3 = -.046, β_3 = -.169 (t = -85.29, p < .001)). These interactions have been graphed in Figures 13 and 14 respectively. From these visualizations, we can see that the expected augmentation effect does not emerge. Instead, we see the best transfer and performance occurs when intentions are high, but type 2 processing is low. To better understand this effect, heat maps were created examining the condition-level results on behavioral transfer, post training performance, and performance change which can be seen in Figures 15-17. These heat maps confirm the best outcomes occur with high intentions but low type 2 likelihood. They also show the interaction effect is more nuanced than suggested by traditional analyses in that the worst outcomes only occur when both implementation intentions and type 2 likelihood are low, as we originally expected. However, the fact that both the worst and best outcomes occur when type 2 likelihood is low, combined with some non-linearity in the change in effects across implementation intentions as type 2 likelihood increases obscures the benefits of type 2 likelihood in this experiment.

Study 1: Discussion

The primary goals of this paper are to: 1) build a process-oriented theory of training transfer, 2) further integrate disparate related theories, 3) incorporate dual process cognition and reinforcement learning more fully into the organizational sciences, and 4) provide a computational model for virtual experimentation which may provide novel insights for both theory and practice. Over the course of several rounds of virtual experimentation, progress has been made towards all these goals. Let us discuss a few of the more important theoretical and practical implications uncovered thus far.

Theoretical Implications

The primary goal of exploring a computational version of the LTM was to show the theory can reproduce common findings in the broader research literature. This is the process of showing generative sufficiency (Epstein, 1999). In the explorations discussed here, it appears that the model can reproduce general patterns of findings in the literature for several important effects, especially regarding the direction of those effects, if not the precise magnitude. However, not all expected effects were observed, indicating the model, although promising, is not yet complete. Here, I will review the standing on some of those effects.

First, the explorations here show that the proposed theory can reproduce the generally expected effects of training on behavioral transfer and performance outcomes we see in the literature. For example, behavioral transfer rates fall in the 10-50% range, which covers typical estimates of transfer in organizations (e.g., Ford et al., 2011). However, plausible effects for observed training outcomes only occur in the model within relatively narrow ranges, especially regarding the parameters governing the true value of the behavioral policies. This limitation could be for at least two reasons. First, it is possible the model breaks down outside of this narrow band of policy values and does not necessarily operate in a way clearly mappable onto real-world phenomena outside of this band. Such a limitation would not itself invalidate the theory, merely place limitations on its generalizability as occurs with any model. Second, it could be an indication of the narrow range of situations we tend to study in the research literature, which is likely to at least be somewhat part of the explanation. In studying training interventions in organizations, we typically enter an organization to deliver a training program to employees who have some degree of experience on the job where they are already successful to a greater or lesser extent. The intervention delivered is likely to be a slight improvement on however they were trained, or however they discovered, to do the job prior to our arrival, despite any organizational claims of the great improvement individuals are likely to see. This naturally creates only slight differences in policy values in the terms of the presented model, and therefore it makes sense that such small differences are where the model best matches existing data. Rarely, if ever, in research would we encounter a situation where we are training individuals, and collecting the necessary data, who are completely incompetent at a task and providing them with the skills to be almost perfectly successful on that task. This situation obviously occurs to some extent when new employees are trained from scratch, but this is not the focus of the kinds of individual studies which are generally conducted. If we were to compare completely novice performance to their later post-training performance, it is more likely that we would see the kind of extreme effects demonstrated at the edges of the present model. As such, the presented model in some regions may be more broadly applicable to studies of the development of expertise than just training transfer (e.g., Benner, 1982)

In addition, implementation intentions (Gollwitzer, 1999) appear to work well in comparison to the limited body of research on their use in organizational training interventions.

The observed effects in the present model were in the expected direction, and plausibly scaled effects were found for both behavioral transfer and performance. Unfortunately, there is not yet a meta-analytic estimate of this effect known to the author of this paper, but typical training results for implementation intentions appear to fall in the medium to large effect range (e.g., Friedman & Ronen, 2015), much as observed here. Thus, the present model appears to account for the general effect of intentions.

Unfortunately, the effect of practice in the present model creates effects in the desired direction but does not really work as would be expected in real training situations. Namely, although we know practice and overlearning opportunities are a key driver of training success, the present model only creates substantial effects when the level of practice approaches and subsequently exceeds the level of experience the agent had previously with the task and not close to the recorded meta-analytic effect (Driskell et al., 1992). Such experience in a real training environment is obviously impractical as I have argued that the degree to which an individual has prior experience with the task is a major driver of training outcomes and most individuals will enter training with large amounts of experience. In such situations, small amounts of practice should have large effects to better match research findings. Future iterations of the model should examine how to better account for the practice effect. One idea would be to count practice attempts as essentially more impactful than regular attempts, if we assume training-based practice attempts count for more than regular attempts this could fit with the idea of deliberate and focused practice being a key to skill development (e.g., Ericsson, Krampe, & Tesch-Romer, 1992).

Utility reactions in this model are an interesting case. The comparable meta-analytic effect targeted was the .46 corrected relationship between utility reactions and transfer described by Blume et al (2010). When analyzed at the replication level, which in this case represents individual agents, there was essentially no relationship between the initial value estimate of Policy B, the stand in here for utility reactions, and our outcomes. This initially seemed to indicate that the model did not work regarding utility reactions. However, when the data was analyzed at the condition level, a correlation between the initial value estimate of Policy B and transfer was .476, almost perfectly matching the meta-analytic estimate. In no other experiment run here was there such a great disparity in observed relationships at the individual versus the conditional level. It might be the case that in this model the effect of utility reactions gets drowned out by random noise when examining individuals. This explanation makes sense when investigating a series of individual value estimate trajectories, where it becomes obvious that the initial estimate for B quickly becomes overwhelmed by the weight of experience and ceases to have drastic effects. However, once we study hundreds of individuals, that noise averages out and the effect of the initial estimate becomes more obvious. Thus, the effect of initial estimates for Policy B's value appear to recreate the observed effect from the literature better than any other parameter examined in this study, but they do not do so in the way initially expected and may need further.

Along similar lines, a set of models were run looking only at the behavioral transfer trajectories of single agents in individual runs of the model. Through these, as displayed in Figures 10A-D, even within the same base parameters of the model agents can follow several types of trajectories in their transfer over time. These trajectories display many of the types outlined for maintenance by Baldwin and Ford (1988). Thus, even the simplest version of the LTM appears capable of generating a classic effect from the transfer literature even without substantial empirical guidance given that such trajectories are rarely studied in practice.

As a demonstration of the potential for the present model to guide future research, the interaction between implementation intentions and type 2 processing on our chosen outcomes was explored. In this experiment, it was expected that the two would have an augmenting effect where high levels of each would result in the best outcomes. However, this was not the case. Instead, we saw the best outcomes when implementation intentions were high, but type 2 likelihood was low. One reason for this might be that the effect of automaticity in the model is driving the interaction here since both variables affect the automatic process either by forcing the agent to engage in that process, or directly altering it. Thus, when type 2 likelihood is low implementation intentions can have a more direct effect on outcomes because they have a chance to work, whereas their impact becomes diluted when agents can more often engage in type 2 processes. This is the kind of initially counter-intuitive finding which can be brought to light by computational models. Future research should now test this effect in either laboratory or real-world situations. If the same interaction effect is found which is predicted by the model then more support will be lent to the theory proposed in this paper, if the opposite is found then the current theory would be falsified.

Practical Implications

A primary goal for the LTM and such associated computational models as has been explored here is the ability to provide useful insight for real world application. The first practical takeaway here is that for jobs at any level of current performance, even small improvements, as long as the trainees are able to discern that the new training is an improvement on whatever they currently do, *can* lead to fairly substantial gains in performance. In addition, it does not necessarily take incredibly large amounts of behavioral transfer to result in substantial performance gains. There are many conditions in the simulations presented here where behavioral transfer is 50% or less, but performance improvements display simultaneous effect sizes we would consider to be large in the traditional research literature. Thus, while it is true that a substantial training-transfer gap exists, we should take heart in the ability of even moderate transfer rates to have substantial effect on important performance outcomes.

One interesting finding in this model is the apparent strong effect of pre-training time on the inability of agents to successfully change their behaviors and performance, as we see in Figures 6 and 7. This finding aligns with viewing training from a nonlinear dynamics perspective (Olenick, Blume, & Ford, in press) which in part suggests that training effects are governed by attractors which develop with experience over time, and stronger interventions would be required to affect permanent change on the job for employees who had been doing the job a certain way for longer prior to the intervention. In the present model, that pre-training time allows for development of such an attractor which the relatively mild intervention studied in this model (seen in the policy change being set at .05) is unable to overcome. Such a finding reemphasizes the need for considering the timing of our organizational training interventions as delay in such training is likely to lead to sub-optimal outcomes.

The modeling of type 2 likelihood for effects on training outcomes may be of particular importance. As discussed in the introduction to this paper, the general failure of training interventions to result in expected outcomes is of great concern to organizations. The LTM suggests that one reason for this failure may be not only a lack of opportunities to use the training, but an opportunity for the trainee to make the kind of effortful decisions that are more likely to lead to them applying their training instead of reverting to their old practices. In fact, there appears to be a critical level below which positive effects are essentially impossible and that this critical threshold (at 33% likelihood of type 2 thinking in this model) must be passed in the transfer environment for positive transfer to occur and result in performance improvements. This could be especially important for environments where such time for thinking is not necessarily always available, such as fast-moving assembly lines. One implication of this model, then, is that organizations should not only make sure trainees can use their training, but also have the opportunity to think about the tasks upon which they have been trained.

Another interesting implication is the degree to which it is useful to encourage exploration in the transfer environment. The results that observed rates of behavioral transfer peak when the exploration rate is about 25%, and performance change follows a similar, but noisier, pattern. Such a curvilinear relationship in general is not surprising as we would expect exploration rates above 50% to be detrimental because agents are then purposely not exploiting their better-perceived policy. However, it is mildly surprising that the optimal exploration rate is so much lower than 50%, suggesting that it is better for an agent to err on the side of exploiting their currently perceived better policy than to explore to some degree. This finding potentially informs the implementation of existing tools, such as the popular Error Management Training, where learners are encouraged to make errors as they explore a new KSAO (e.g., Keith & Frese, 2008). Such an approach to training helps improve outcomes as trainees learn from their mistakes and push through initial struggles with a new skill. According to the present model, this error-based approach extended to the entire post-training period would likely be beneficial for outcomes as well, but only to a point. Therefore, we should encourage trainees to continue trying a newly trained task approach but only to a moderate degree because we do want them to settle on a behavioral approach for the long term, and we want them to discard approaches which do not actually improve performance outcomes.

Finally, the virtual experiment here exploring the mutual effects of implementation intentions and type 2 likelihood can provide us some guidance on how to best include implementation intentions in our training designs. The results do suggest that implementation intentions are generally beneficial regardless of the environment (assuming here that the likelihood of engaging in type 2 processes is largely a function of one's work environment), but that the required strength to have a substantial effect and the ability of implementation intentions to improve our outcomes changes based on that environment. When jobs are such that the use of type 2 processes is highly likely, the inclusion of implementation intentions is unlikely to have a substantial impact on our desired outcomes, though they would still be useful. However, if we are conducting training in an environment where type 2 processing is especially *unlikely*, such as a fast-paced assembly line or similar environment, implementation intentions are likely to be highly beneficial to include in our training programs. However, we must work to ensure these intentions are as strong as possible, as weak intentions are also unlikely to have a great effect on outcomes.

Conclusion

The above represents the first iteration of modeling in the building of the LTM, which has the goal of becoming a unifying theory to explicate the moment-to-moment process underlying training transfer. Although not perfect, the general patterns of results appear to align well with existing findings. We know the model is wrong, but the degree to which it is meaningfully wrong (Box, 1976) could be contended to be rather small for the time being as exactly replicating existing meta-analytic effects is not completely necessary. Future iterations of this portion of the model should attempt to refine the operation of parameters and the math governing their effects to better match their real-world counterparts, such as the effects of practice attempts, but that is a task for another time. For now, I argue this is a reasonable first iteration of the LTM with apparent implications for both theory and practice. With that, we know that there are substantial ways in which the model as it stands *is* meaningfully incorrect, such as not accounting for social learning mechanisms. To rectify this shortcoming, another iteration of theorizing and modeling was endeavored upon.

Study 2A: Adding Social Learning to the LTM

The first iteration of the LTM appears to have done well in describing the transfer decision and learning process of a single agent/learner, but people in the real world do not learn in isolation. Instead, they also learn from the models around them. Thus, the model was iterated to include a social learning (Bandura, 1977) process allowing agents to learn from other agents in their environment.

Social Learning Theory

Bandura (1977) introduced Social Learning Theory (SLT) partly as a reaction to the then-dominant behavioral approaches to learning exemplified by early reinforcement learning. Bandura (1977) posits that individuals not only learn from their own experiences, but that they also learn from others. In fact, Bandura argued that most learning occurs through observing others in action, a process called modeling. Once the learner observes a model complete an action, they can form an idea of how the new behavior is to be performed, and later use that as a guide to their own actions. Learning through observing others is more efficient than only learning through individual experience as less trial and error is required to learn a given behavior. Bandura's (1977) approach also emphasizes reciprocal determinism between cognitive, behavioral, and environmental influences. At the risk of oversimplifying these influences, the cognitive processes of the individual affect their behaviors, which affect their environment. The individual receives feedback from the environment based on the effects of their behavior, which lead to changes in cognition and behavior in the future.

The essential proposed change in the LTM once we consider social learning is that there is an effect of other individuals on the learning process of our target learner. The LTM accounts for this effect by considering multiple learners engaging in the transfer process simultaneously.

Obviously, this does not represent all potential social learning influences on our target learner. Instead, the current approach best represents an idealized version of a work team or community of practice which are all exposed to the same learning intervention and then must attempt to transfer back to their work environment. Though this conceptualization is admittedly simple, it aligns with arguments that involvement in communities of practice can enhance training transfer through the sharing of information across the community network (e.g., Tentin, 2001).

The new conceptual model, seen in Figure 18, incorporates any number of learners in addition to a target learner. Every learner in the model is assumed to have access to the same two policies, and to proceed individually through the basic decision and learning process described above. However, the learning step in the model no longer only relies on the learner's own experience but includes feedback from the experiences of all other agents. That is, it is a mechanism whereby other agents in the environment have modeled for the target learner the behaviors represented by the two policies, and the agent is then informed of their effectiveness through that observation. In this way, the perceived value of each policy becomes a type of pooled estimate from all learners. This pooling procedure is not assumed to take all experiences of learners equally. Instead, the pooling for any individual is such that they weight their own experiences differently than any of their co-learner's, the extent to which we can control via a parameter in the model. Once policy values are updated for each learner, the decision and learning process iterates. It is expected that additional learners in the model will improve transfer and performance of target learners because it reduces poor estimation of the new policy's value through the more rapid reduction in sampling error created by more learners gaining experience. The faster accrual of experience as a group should allow for more quickly discarding new policies when they are poor, and a decreased likelihood of incorrectly discarding a policy when it is good based on initial random error that could underestimate policy value.

The Formal Transfer Model with Social Learning

To expand on the formal version of the LTM to include social learning, three essential changes must be made to the model: additional learning agents, a way to pool experiences of the agents, and a way to weight the importance of group experiences against those of the target agent. The first change is simply conceptual, instead of assuming we are only interested in one agent engaged in the transfer process, multiple agents engage in this process simultaneously. To build on the previous reinforcement learning approach from computer science and account for all agent experiences, it could be possible to draw on algorithms that are designed for multiple agents (Sutton & Barto, 2018). However, those algorithms are designed for multiple agents attempting to solve a single problem, giving each agent a chance to explore more possible solutions. Such approaches are not necessarily the best fit for a model of transfer where multiple learners are trying to solve a single "problem", but only have limited solutions which they could apply. Future extensions of the present model could explore other options along these lines, but they are not the most parsimonious potential approach, which is a primary goal of theorizing (Box, 1976).

A simpler approach is to pool the experiences of multiple learning agents to affect the value estimates of each individual agent and thereby affect application decisions. The easiest way to pool experiences of other agents in a target agent's environment would be to average the other agent's value estimates. The average value then of other agents, for the *j*th agent in the model, can be defined as:

$$G_{tj}(a) = \frac{\sum_1^N Q_t(a)_i}{N}$$

Algorithm 5. Other Agent Value Estimation

where N is the number of other agents in the model, and $Q_t(a)_i$ is the value estimate of the *i*th other agent for Policy A. Calculating at the value estimate level avoids an assumption that the target agent knows the outcomes of the individual attempts of the other agents, and only assumes they have an accurate view of those agent's evaluation of each policy. However, a simple averaging of the value estimates of the other agents does assume an equal weighting of the opinions of all other agents, so does not account for network effects and varying strengths of ties to those agents. Exploring the effects of networks will be an interesting avenue for future work.

Regardless of the assumptions, the value estimates of the group must be combined with the estimate of the target agent in some way. The LTM proposes a weighting approach that can vary the degree to which the target agent weights their own value estimate over that of the others in their group. This approach results in an ability to vary the degree of connectedness between the target agent and the rest of the group, making the target agent and their group a type of loosely coupled system (e.g., Weick, 1976). Let this level of connectedness be defined by *C*. The variable *C* represents a weighting factor such that when levels of connectedness are high the value estimate of the group will be weighted more heavily than that of the individual. Thus, we can calculate the target agent's value estimate accounting for the estimate of the group as:

$$wQ_t(a) = (1 - C)Q_t(a) + CG_t(a)$$

Algorithm 6. Weighted Value Estimate

However, since $G_t(a)$ is only calculated when there are multiple agents in the model, when only a target agent exists $wQ_t(a) = Q_t(a)$. Variables and equations introduced in this section can be found in Tables 7 and 8 respectively.

Study 2A: Method, Simulation and Results

As with the first model, the extension of the LTM was instantiated in a computational model by expanding on the model from Study 1 in NetLogo. A visual of the modeling environment and associated code can be found in Appendix B. Otherwise, methods outlined for Study 1 apply to this model as well.

Virtual Experimentation

A primary goal of model exploration at this stage was to ascertain the effects of having multiple agents learning simultaneously, and the degree of influence of those agents on each other in determining transfer outcomes. To explore these effects, two simple verification checks were made, then three experiments were run to simultaneously check for generative sufficiency, sensitivity, and robustness.

Model verification

The two primary changes in this model are the addition of trainees to the modeling environment, and the mechanism for combining effects of experience from multiple agents for use by each individual agent. The check to ensure multiple agents are populated into the environment is simply visual in NetLogo, and it was affirmed that the proper number of agents were generated as specified. To check the pooling procedure, levels of connectedness were set at 0 and 1 and the model run for 500 time steps with 20 agents. When connectedness is 1, the pooled estimate for each agent should be equal to the pooled estimate of all other agents in the model. For example, for agent 1, when all other agents have an average estimate for Policy B of .73, a fully connected model should have a pooled estimate of .73 for agent 1. This is indeed the case. On the other hand, when connectedness is 0, the pooled estimate for Policy B for agent 1 should be equivalent to agent 1's own value estimate for that policy. For example, if that agent estimates the value is .70, the pooled estimate should also be 0. This is also indeed the case in testing the model. Therefore, the model appears to be operating as planned.

Number of Trainees

To initially understand the effect of the number of learners/trainees in the transfer environment a series of simulations were completed manipulating the number of trainees from 1 to 20, 500 replications each. Other variables held at levels decided upon in the first model (type 2 likelihood at .80, initial policy estimates at .50, burn in time 100, transfer time 500, no practice, no implementation intentions, true Policy A reward .70, change in policy value .05). Additionally, the new connectedness variable was set at .50. It was expected that more agents in the model will improve transfer outcomes by improving the value estimates of each agent through a more rapid increase in sample stability.

In examining these results, it was found that there was almost no relationship at the replication level between number of trainees and either behavioral transfer ($r(10000) = .007$, $p = .484$) or post training performance ($r(10000) = .005$, $p = .617$). At the condition level there was a slight positive effect of the number of trainees on both behavior ($r(20) = .14$, $p = .556$) and performance ($r(20) = .13$, $p = .585$). This condition-level effect can be further examined by looking at the condition-level results of behavior and pre-post performance change (in Cohen's *d*) in Table 9. Despite the small positive correlation between the number of trainees and behavioral transfer, upon examination of the descriptive statistics in Table 9 it is obvious this effect is of little consequence, with transfer only increase from 43% to 44% as the number of trainees increases from 1 to 20. On the other hand, there is a substantial impact on observed Cohen's *d* for pre-post performance change whereas the number of trainees increases from 1 to 20 the observed effect size increases from .28 to 1.43.

Connectedness

A second set of simulations were run to explore the potential effects of the connectedness parameter. The expected effects of manipulating the degree of the connection between the individual and group were less clear than for the number of trainees. Assuming having other trainees in the model is beneficial to the agent, it would be expected that more connectedness would also be beneficial as the potential detrimental effects of sampling error leading an individual agent down a sub-optimal path should be diluted the more they take into account the experiences of other agents.

To test this, the connectedness parameter was swept from 0 to 1.0 in .05 increments, with 10 agents simulated, and holding all other parameters constant at the same levels in the above simulation. Unfortunately, results for this simulation were even less impressive than those for the number of trainees. Relationships between connectedness and behavioral transfer and post training performance were nonexistent ($r(10500) = .009$, $p = .356$, and r10500) = .001, $p = .918$, respectively) at the replication level, and were mixed at the condition level ($r(10500) = .217$, $p = .345$, with behavior, $r(10500) = -.031$, $p = .894$) with post training performance, and $r(10500) = -.078$, $p = .737$) with pre-post performance change). Table 10 displays the condition-level outcomes for behavioral transfer and pre-post performance change effect size. Examination of this data confirms no effect of note as behavioral transfer remained ~44% regardless of condition, and pre-post performance change was around $d = 1.00$ with some random error.

Interaction between Trainees and Connectedness

Given the results of the above simulations it was not expected that an interaction effect would be enlightening. However, such a model was proposed for this project, and given the intricacies of computational models such as these it is possible traditional analyses obscure meaningful relationships. Thus, the potential interactive effect of trainees and connectedness on transfer outcomes was still explored. It was predicted that a positive effect of having multiple agents in the model would increase as the degree of connectedness increased. This effect was expected as the agent should benefit from taking advantage of the extra experiences of their colleagues through greater weighting of those experiences and their combined increased sampling rate. To test for this, the number of trainees was swept from 1 to 20, and connectedness from 0 to 1.0 in .05 steps, fully crossed, while holding all other variables constant at the levels chosen from Model 1. Moderated multiple regression was used to examine the effects of the number of trainees and connectedness on behavioral transfer and post training success. In alignment with the previous results from this model, there are no discernable main or interaction effects from this experiment. In predicting behavioral transfer neither the number of trainees (b_0 = .436; b_1 = -.00004) or connectedness (b_2 = .001), nor their interaction (b_3 = -.00006) demonstrated substantial effects. Similar results were found in predicting post-training performance (b_0 = .722; b_1 = -.000007; b_2 = *.00004;* b_3 = .000007). In accordance with other analyses in this paper, heatmaps were also generated to examine potential effects missed by more standard analyses. These reconfirmed no substantial effects, with differences between conditions largely attributable to noise. An example of this can be seen in Figure 19.

Study 2A: Discussion and Conclusion

The goal of this iteration of the LTM was to account for general effects of social learning in a training transfer environment in a parsimonious matter. The initial modeling discussed here suggests this attempt was a general failure as the expected effects of the primary variables failed to emerge.

Specifically, it was expected that the number of trainees would improve transfer outcomes as the greater numbers would essentially smooth out sampling errors for single agents which could lead to suboptimal transfer. This prediction does not appear to have quite been the case. At best, there appears to only be a slight improvement in behavioral transfer and post training performance as the number of agents increase, and nothing like the strong effects expected based on SLT (Bandura, 1977) or existing meta-analytic social effects in training transfer (Blume et al., 2010). A misleading exception to this failure lies in the observed effect sizes comparing pre and post-training performance, which range from d = .28 to 1.43. However, given the lack of improvement in behavioral transfer and slight relationships between the number of agents and performance, this improvement in Cohen's d appears to be an artifact of its calculation. Specifically, Cohen's d is in part calculated using the pooled standard deviation of the two groups being compared. When the number of agents increases the observed standard deviations of performance within those groups decreases as sampling error becomes less problematic. Then, when the effects are compared the pooled standard deviation utilized is smaller when there are more agents, making the effect size appear artificially large. Thus, there is an effect of more agents in the model due to their effect on sampling error as was predicted, but the effect is not the one which was expected. Further, the results for connectedness effects were even more disappointing. It was hoped that connectedness would be a simple way to recreate the social support which has a meta-analytic effect of .21 on transfer (Blume et al., 2010), but this appears to clearly not be the case. More sophisticated forms of social learning will need to be explored to see if they can account for such social effects.

Given the clear failure of this integration of social learning into the LTM, we are forced to explore other options. It is to the exploration of these other options we shall now turn.

Study 2B and 2C: Rethinking Social Learning Model

Unfortunately, the initial attempt to include a social learning mechanism in the LTM failed. In order to assess other potential options for social learning in the present context, a search was conducted for more existing models of social learning mechanisms. Multiple potential mechanisms have previously been modeled to answer various research questions, such as the use of genetic algorithms (e.g., Yeh & Chen, 2001), imitation (Richerson & Boyd, 2005), and emulation (Lopes, Melo, Kenward, & Santos-Victor, 2009), some of which have been applied to organizational research such as coordination within teams (e.g., Singh, Dong, & Gero, 2013). Although all these approaches, and likely others, could prove fruitful, the present modeling will focus on imitation.

The choice to focus on imitation lies in its use in studying the mutual development of culture and genetics in human populations. Richerson and Boyd (2005) described the process by which human culture and genetics mutually reinforced each other over thousands of years to produce societies, and the actual humans within it, that we know today. A primary mechanism within models on this relationship is social learning as the actions of groups over time are largely dictated by the pressures exerted upon the individuals within those groups by the other people around them. Over time, these pressures lead to the success and spread of certain cultural artifacts and the elimination of others and the ability to acquire novel behaviors via social learning is a prerequisite for cumulative change. We can see the outcomes of hundreds of generations of such pressures in the emergence of complex cultures representing the sum of the socially selected actions, beliefs, values, etc. that were adaptive for success in the social groups within which they emerged.

I argue this view is particularly relevant to examining training in organizations. As has been discussed above, social effects are one of our most important factors in the success or failure of training and subsequent transfer. These social effects come in several guises, such as manager and coworker support, climate for transfer, and organizational culture (e.g., Blume et al., 2010). Much as Richerson and Boyd (2005) view social learning as a mechanism through which culture is developed and reinforced, we can view the social effects within training and transfer as a form of culture that both affects and is reinforced by the actions of the individuals within the organization. In our first attempt here at a social learning mechanism we can already see this form of mutual causality. Namely, agents within the model have experiences with their task and their experiences combine to form a collective view of the task which is an emergent property of the simulated work group. This emerged view then acts as the cultural context within which the agents act, and this context impacts the decisions the agents make. These mutually causal, simultaneous top-down and bottom-up (Kozlowski & Klein, 2000) effects then dynamically play out over time. Thus, the underlying causal relationship modeled in the prior iteration of the LTM appears to be in line with other models of culture in the scientific literature, but the actual social learning mechanism too simplistic to have the expected effects. To rectify this shortcoming, two models were built and explored to study the potential effects of imitation.

In Richerson and Boyd (2005) imitation occurs when organisms copy others in their environment as a way to navigate that environment. As in SLT (Bandura, 1977), imitation allows organisms, in this case humans, to learn new behaviors and the consequences of those behaviors through observation. This observation improves the rate and outcomes of individual learning, all else being equal. Two predominant forms of imitation are pertinent here. The first form is *imitation of the successful* where individuals will tend to do the actions they see successful individuals around them doing on the same tasks. Such a mechanism can be seen throughout our modern world. For example, many people play and watch major sports with a dream of someday being as good as the professionals they see on television. To improve at their own games, a common approach is for individuals to attempt to emulate the athletes they see succeeding at the same game. Simple searches for tips on golf, for example, return hits on how to drive the ball like Rory McIlroy, or putt like Phil Mickelson.

Richerson and Boyd's (2005) second form of social learning occurs through a *frequency bias* where learners tend to do the things that the majority of their peers are doing. That is, when in a group, people will tend to do the things that the individuals around them are doing, using a "When in Rome…" approach to their decision making and learning. This approach is especially adaptive to learners in unfamiliar situations in that they can take cues from those around them on how to navigate the novel environment. Richerson and Boyd's (2005) modeling suggest strong conformist biases benefit social groups over many generations in improving the group's ability to survive in their environment.

In either case, Richerson and Boyd (2005) argue these social learning mechanisms are fast and frugal forms of learning which offload the burden of learning much information through direct experience. In addition, these learning mechanisms are biases in favor of following the successful or the lead of their social groups. Thus, their framing of the benefit of these social learning mechanisms also fits well with this paper's framing within research on dual process cognition (e.g., Kahneman, 2011). However, for clarity we need to find a way to readily distinguish between the two types of social learning Richerson and Boyd (2005) describe. In their description conformity is more about a form of coercion than it is vicarious learning in the form of SLT (Bandura, 1977). Therefore, in our nomenclature it does not seem quite correct to label it directly as a form of imitation. On the other hand, the imitation of the successful does fit well with SLT. Thus, it seems proper to let imitation mean specifically *imitation of the successful*, while relabeling the coercive form as *conformity*. Therefore, the rest of this paper will use *imitation* and *conformity* to refer to these types instead of imitation alone.

To examine the potential effects of these mechanisms on the LTM, two independent iterations of the theory and associated computational model were made. The first, referred to as Model 2B, focused on imitation, while Model 2C focused on conformity.

Model 2B Overview

This iteration of the LTM explores the effects of social learning through a tendency for learners to imitate other successful learners in their environment. For this mechanism, it is proposed that learners observe others in their environment, judge their performance and their behavior, and have some degree of likelihood of following the same behavior that high performing other agents are enacting. To include this mechanism in the formal version of the LTM and its associated computational model, we need to make two adjustments to Model 2A.

The first change lies in the observational and pooling procedure originally proposed. Instead of pooling the value estimates of all other learners in their environment, learners must instead track the actual performance of their fellow learners. From that set of learners, they must then judge which one exhibits the highest performance for them to make a judgement about how to imitate that high performer. This approach does assume that learners have a perfect ability to judge the performance of others that was not present in Model 2A, but this assumption can be relaxed and explored in future modeling endeavors.

Once performance and behavioral judgements have been made, a mechanism needs to be created for those observations to affect the behavioral choices of the observing agents. Within the dual processing framework, the decision to imitate other agents is proposed to fit more cleanly into type 2 processes due to the level of cognitive effort required to make accurate judgements of other's performance and behavioral decisions based on those judgements. It is possible that the mechanism could be placed in type 1 processes which would fit with future explorations where the assumption of perfect observation is relaxed, but for now we will assume the decision to imitate the successful is a more conscious and effortful one than a more automatic one. Thus, when individuals engage in their type 2 processes, they first must decide on whether to imitate someone else or not. Within the computational version of this model, the likelihood of imitating another agent is controlled with a parameter labeled *imitate*. If a learner does choose to imitate someone else, they must scan their environment for other learners and judge their performance to identify the one with the highest level of performance. Once identified, they must then observe the behavioral choices that learner is making and then apply the same behavior. In the computational model agents carry out this observation process and choose to apply the behavioral policy enacted by the most successful other agent in their environment *on the previous task attempt.* Therefore, the behavior of any agent i in the model at time $t + 1$ is the behavior of the highest performing other agent in the model at time t, when agent i chooses to imitate during task attempt $t + 1$.

Model 2C Overview

Model 2C represents a modification on the theory and model in 2B to change the social learning mechanism from imitation to conformity. Thus, instead of a tendency to do the same behaviors as successful learners around them, under this model learners tend to do the behaviors that the majority of the learners around them are doing. In this case, individuals do not need to track the performance of others around them, only their behavioral choices. From these observations the learner can use a simple voting procedure to determine which behavioral choice is the most common among their group. In the computational version of this proposal, when each agent decides to conform to their group their behavior on the task at time $t + 1$ is equivalent to the behaviors displayed by the majority of the other agents in the environment at time t. The tendency to conform to the group on any given task attempt is controlled via a *conform* parameter.

Study 2B: Method, Simulation and Results

The described theoretical additions to the LTM regarding the use of imitation as a social learning mechanism were instantiated in a computational model, expanding on the base model explored in Study 1. A screen shot of the modeling environment and copy of the associated simulation code in NetLogo can be found in Appendix C. The new mechanisms in this model were verified through examining the tracking mechanisms of agents to ensure they were correctly identifying the performance and behavior of the top performing other agents in their environment, and that they would follow the indicated behavior under conditions where they always engage in type 2 processes and always imitate the best performers. Once this was completed, a small experiment was run to study the effects the number of trainees in the model and level of imitation have on our outcomes of behavioral transfer and task performance. It was expected that both the number of trainees and level of imitation would improve transfer outcomes as agents would benefit from the increased sampling rate of the agents around them, making it more likely that at least one agent discovers that Policy B is indeed the better policy and that finding would propagate through the rest of the agents.

Trainees Versus Imitation Experiment

To study the effects of the number of trainees and level of imitation in this version of the LTM, a simulation was conducted crossing the number of trainees, swept from 1 to 20, with level of imitation rate swept from 0 to 1 in .05 increments. Other variables were held at the levels chosen in Model 1: 100 pre-training time steps, 500 transfer time steps, type 2 likelihood set at .80, initial policy estimates were .50, true value of Policy A was .70 with a change in value of .05, and 500 replications of each condition.

In examining the results of this simulation, we see the direction of relationships expected both within this model, and more broadly as guided by social effects in the training literature. Namely, at the replication level, the number of trainees in the model was positively related to both behavioral transfer ($r(210000) = .103$, $p < .001$), and post training performance ($r(210000) = .156$, $p < .001$). In addition, the level of imitation was also positively related to both behavioral transfer ($r(210000) = .495$, $p < .001$), and post training performance ($r(210000) = .730$, $p < .001$). Then, a multiple regression analysis was performed to test the joint effects of the number of trainees and imitation level on the transfer outcomes. In predicting behavioral transfer, both the number of trainees ($F(3, 209996) = 89557.07$, $p < .001$, $R^2 = .75$; $b_0 = .728$, $t = 3096.22$, $p < .001$; $b_1 = .004$, $\beta = .156$, $t = 107.61$, $p < .001$) and imitation rates ($b_2 = .392$, $\beta = .730$, $t = 505.13$, $p < .001$) show positive relationships, and a positive interaction ($b_3 = .006$, $\beta = .064$, $t = 44.04$, $p < .001$). In predicting post training performance, both the number of trainees ($F(3, 209996) = 24351.15$, $p < .001$, $R^2 = .51$; $b_0 = .736$, $t = 32701.20$, $p < .001$; $b_1 = .0002$, $\beta = .106$, $t = 54.78$, $p < .001$) and imitation rates ($b_2 = .020$, $\beta = .495$, $t = 263.55$, $p < .001$) show positive relationships, and a positive interaction ($b_3 = .0003$, $\beta = .046$, $t = 24.36$, $p < .001$). Finally, predicting pre-post training performance change (in Cohen's d) across conditions, both the number of trainees ($F(3, 416) = 1816.19$, $p < .001$, $R^2 = .96$; $b_0 = 2.346$, $t = 175.72$, $p < .001$; $b_1 = .137$, $\beta = .770$, $t = 59.00$, $p < .001$) and imitation rates ($b_2 = 1.849$, $\beta = .548$, $t = 41.93$, $p < .001$) show positive relationships, and a positive interaction ($b_3 = .111$, $\beta = .189$, $t = 14.47$, $p < .001$). These interaction effects, displaying an augmenting effect between the number of trainees and imitation rates, were graphed in Figures 20-22. In addition, to better understand the nuances of these effects, heat maps were generated and can be found in Figures 23-25. In examining these heat maps, we see almost no actual effect of the number of trainees beyond that gained by adding even one agent to the model. On the other hand, we see a steady improvement in behavioral transfer and post training performance as imitation rates increase. In the calculation of pre-post performance effects, we see the highest effect sizes when trainees and imitation are high, but this is again likely an artifact of the calculation of Cohen's *d*.

Study 2C: Method, Simulation and Results

The theoretical additions to the LTM described above regarding the use of conformity as a social learning mechanism were instantiated in a computational model, expanding on the base model introduced in Study 1. A screen shot of the modeling environment and copy of the associated simulation code in NetLogo can be found in Appendix D. The new mechanisms in this model were verified through examining the tracking mechanisms of agents to ensure they were correctly identifying the behaviors of their fellow agents, and that they would follow the indicated behavior of the majority under conditions where they always engage in type 2 processes and always conform. Once this was completed, an experiment was run to study the effects of the number of trainees in the model and level of conformity have on our outcomes of behavioral transfer and task performance. It was expected that both the number of trainees and level of conformity would again improve transfer outcomes as agents would benefit from the increased sampling rate of the agents around them, making it more likely that at least one agent discovers that Policy B is indeed the better policy and that finding would spread through the rest of the agents. Thus, as with the imitation model, it was expected that we would see positive relationships between the number of trainees, level of conformity, and the transfer outcomes.

Trainees Versus Conformity Experiment

To study the effects of the number of trainees and level of conformity in this version of the LTM, a simulation was conducted crossing the number of trainees, swept from 1 to 20, with level of conformity swept from 0 to 1 in .05 increments. Other variables were held at the levels chosen in Model 1: 100 pre-training time steps, 500 transfer time steps, type 2 likelihood set at .80, initial policy estimates were .50, true value of Policy A was .70 with a change in value of .05, and 500 replications of each condition.

In examining the results of this simulation, we see the opposite of the general relationships expected within this model. At the replication level, the number of trainees in the model was negatively related to both behavioral transfer ($r(210000) = -.203$, $p < .001$), and post training performance ($r(210000) = -.144$, $p < .001$). In addition, the level of conformity was also negatively related to both behavioral transfer ($r(210000) = -.742$, $p < .001$), and post training performance ($r(210000) = -.528$, $p < .001$). A multiple regression analysis was performed to test the joint effects of the number of trainees and conformity level on the transfer outcomes. In predicting behavioral transfer, both the number of trainees ($F(3, 299996) = 103802.16$, $p < .001$, $R^2 = .77$; $b_0 = .232$, $t = 907.59$, $p < .001$; $b_1 = -.006$, $\beta = -.203$, $t = -146.30$, $p < .001$) and conformity rates ($b_2 = -.452$, $\beta = -.742$, $t = -536.13$, $p < .001$) show positive relationships, and a negative interaction ($b_3 = -.007$, $\beta = -.070$, $t = -50.73$, $p < .001$). In predicting post training performance, both the number of trainees ($F(3, 299996) = 30371.68$, $p < .001$, $R^2 = .55$; $b_0 = .712$, $t = 30279.36$, $p < .001$; $b_1 = -.0003$, $\beta = -.144$, $t = -78.84$, $p < .001$) and conformity rates ($b_2 = -.023$, $\beta = -.528$, $t = -289.96$, $p < .001$) show negative relationships, and a negative interaction ($b_3 = -.0004$, $\beta = -.052$, $t = -28.63$, $p < .001$). Finally, predicting pre-post training performance change (in Cohen's *d*) across conditions, both the number of trainees ($F(3, 416) = 3150.12$, $p < .001$, $R^2 = .55$; $b_0 = .058$, $t = 8.78$, $p < .001$; $b_1 = -.012$, $\beta = -.104$, $t = -10.28$, $p < .001$) and conformity rates ($b_2 = -1.979$, $\beta = -.918$, $t = -91.21$, $p < .001$) show negative relationships, and a negative interaction ($b_3 = -.121$, $\beta = -.323$, $t = -32.03$, $p < .001$). These interaction effects, displaying a depressive effect between the number of trainees and conformity rates, were graphed in Figures 26-28. In addition, to better understand the nuances of these effects, heat maps were generated and can be found in Figures 29-31. These heat maps show that the best outcomes indeed occur when the number of trainees and rates of conformity are low. In addition, they also show a clear sensitive area in the model and some non-linear effects. Specifically, outcomes suddenly improve as conformity rates drop below about .45. However, the level this change occurs depends on the number of trainees in the model, such that the transition occurs at higher levels of conformity when there are fewer trainees in the model. An interesting pattern also emerges comparing when there is an odd number of trainees in the model versus an even number such that the transition point to better outcomes occurs at a higher level of conformity for even numbers of trainees than for the odd numbers around them. This is likely a statistical artifact of the voting process in the model rather than something of great significance.

Study 2B and 2C: Discussion and Conclusion

The models introduced in Studies 2B and 2C were meant to explore other potential mechanisms to account for social effects in training transfer studies which the initially proposed model failed to do. From the initial experimenting outlined here, it appears that either have the ability to provide interesting insights into the transfer process, at least beyond that obtained in the original theory. Here, let us briefly discuss the implications of these models for both theory and practice.

Implications for Theory

The primary effects on transfer from a social standpoint lie in the environmental factors of perceived support and climate for transfer. Corrected meta-analytic estimates for the effects of these on transfer are .21 and .27, respectively (Blume et al., 2010). In Model 2B we studied the potential effects of imitation on transfer, which is the tendency to engage in the behaviors other successful learners are engaged in. In Model 2C we viewed the social learning mechanism from a standpoint of conforming to the behavioral tendencies of the majority of the other learners around a target learner. These mechanisms could be argued to fit conceptually with what is occurring in producing the effects we see for support and climate. Namely, these two effects are based on the perceptions of learners of the actions of those around them for providing the necessary social and physical conditions in which they can transfer their training. In both present models, as argued above, we have created an emergent environment from the actual behaviors of agents which in turn produce a social context in which the agents must act. When the social environment created by the agents is such that it promotes the transfer of the trained policy, this is akin to the agents perceiving an environment supportive of their transfer attempts.

In addition, the two models might fit conceptually a little better with specific effects. For example, Model 2B relies on target agents seeing other successful agents apply a behavior that the target agent can then mimic. This is more of a one-on-one interaction where the successful agent essentially either supports or does not the target agent's transfer by providing a positive role model for that behavior or not. This conceptually fits with recommendations for managers to promote transfer by modeling desired behaviors to their employees (e.g., Lancaster, Di Milia, & Cameron, 2013). Although not labeled as managers in the model here, managers could be seen as successful employees whom their followers are likely to view as role models as occurs in the outlined mechanism in Model 2B. To that end, examination of Model 2B shows an ability to create relationships in the desired direction, but the effects of the actual imitation mechanism are of a substantially larger magnitude than observed in meta-analyses of support effects on transfer (Blume et al., 2010). A reason for the increased effect sizes observed in the present model being so much larger than target effect sizes may be the current ability of the agents to perfectly view the performance and behaviors of their models. Adding noise to their observations to reflect imperfect observation in real life may correct this deficiency and bring findings more in line with research. For now, this model appears to be a great step forward over the previous iteration of the model that fits better with existing research both within our field, and with scientific efforts around social learning in general.

The case surrounding Model 2C is considerably more complicated. Conceptually, Model 2C fits more cleanly with effects of transfer climate because the underlying mechanism in this model, conformity, is no longer a one-on-one modeling case, rather it is truly a group-level consideration. That is, here the agents form a social environment regarding their collective use of the behaviors available to them. That social tendency to use either behavior available to them or not could be considered the climate for the use of that behavior. Therefore, if the group tends to use the trained Policy B, the climate would be one that is positive for transfer. If the group tends to use the pre-existing Policy A, there would be a negative climate for transfer. Initially it was expected that conformity would be positively related to transfer as the group would be more likely to collectively realize the trained policy is beneficial for their use and therefore create that positive climate. However, that is not what we find in simulation. Instead, we find that conformity has a substantial negative effect on transfer, although the absolute magnitudes of those effects are closer to the meta-analytic effects of climate than observed for support in Model 2B (Blume et al., 2010). This finding was initially surprising. However, in retrospect, because of the dynamic nature of the mechanism the negative relationship perhaps should be expected. Specifically, the behaviors of agents at any time $t + 1$ are a function of the behaviors of the group at time t. This in effect creates a heavy bias, especially when conformity is high, towards the continued use of Policy A because when the agent begins the transfer period the behavior for all agents at the previous time point was Policy A. To overcome this fact, it takes the agents independently choosing to apply Policy B and slowly changing the balance of the group until a majority are applying Policy B. This process is harder and takes longer when there are more agents in the environment, which also accounts for the now negative effect of the number of trainees in the model.

Therefore, I argue that initially low pressure to conform is actually akin to a climate allowing transfer because the agents are free to explore the benefits of their training rather than being pressured to avoid doing so and the inverse directional relationship with how we operationalize this effect in the literature should be expected. Within actual work groups there can be substantial pressure placed on workers not to comply with organizational interventions such as training and therefore create a negative climate for transfer, whereas the absence of such pressure can be interpreted as positive climates for transfer. Over time, as more individuals pick up the new behavior that pressure could reverse to lead to improved transfer. This essential switch in pressure could be a reason why we see the sudden improvement in transfer outcomes when conformity drops below .45 in the present model. Early in transfer attempts the agents need that freedom to do their own exploring, but as transfer goes on there is some benefit to social pressure helping bring late adopters over to begin transferring at a greater rate. Thus, it appears that with some reconceptualization of what the parameters represent, this model also provides a potential window into existing effects in the transfer literature, though more work will be required to explore those effects and verify the mechanisms are correct. For the time being, given the closer absolute magnitude of the effects in this model and the more nuanced relationships observed with the parameters in this model with the outcomes of interest, this model may provide more potential insights than Model 2B for future work. It is for this reason that this model is chosen to provide the basis for the next round of modeling outlined in Study 3, although it is acknowledged much work remains to verify this model for long term use.

Future modeling of social learning

As mentioned in the introduction to these two models, there are other versions of social learning which could be interesting to explore in future efforts to model the employee learning and transfer process. For example, the use of a genetic algorithm approach (e.g., Yeh & Chen, 2001) could provide interesting insights for skill development if we were interested in how employees might generate their own novel solutions to work tasks and then propagate their discoveries to their work groups and the organization at large. Furthermore, existing models sometimes model tradeoffs in several dimensions of preference at the same time. Along these lines, Lopes et al. (2009) modeled tradeoffs between making decisions based on individual preferences, versus making those choices based on social pressures from either imitation or emulation. In their modeling, momentary choices are based on a tripartite tradeoff between these three pressures, where increased weight of any one type lowers the weight of the other two. In the present modeling we focused on only a single type of social learning, imitation or conformity, and a tradeoff with choices based on personal experience. Future iterations should explore the potential simultaneous effects of these pressures.

In addition, models where imitation and conformity are placed in type 1 processes instead of type 2 processes should be explored. The argument for why they were placed in type 2 processes for this model was laid out above, and it seems likely that placing these mechanisms in type 1 processes would only exacerbate the overly strong relationships they displayed with transfer and performance compared to meta-analytic estimates (Blume et al, 2010), although doing so would potentially fit with Richerson and Boyd's (2005) description of these mechanisms as fast and frugal, though it is not clear they are discussing cognitive load rather than general effect. This expectation is due it furthering their ability to override type 2 processes prior to being able to make any further conscious judgement. On the other hand, the way they were included in the model may de facto place them somewhere between the clean separation of type 1 and type 2 processes otherwise followed in this paper, as they are only enacted when type 2 processes are called upon but do not make the same level of logical judgement as other type 2 processes modeled and instead override those processes to automatically imitate or conform. Thus, the decision to imitate or conform is treated as conscious, but its execution is more automatic in nature. The middle ground occupied in the present model by the imitation and conformity mechanism does potentially fit with the view of cognitive systems on a continuum from conscious and effortful to automatic (Evans & Stanovich, 2013). Regardless, it might be more fruitful to relax the assumptions of perfect observation of the learner's fellows in the model, which would add noise to the imitation and conformity decisions and should therefore lower the observed relationships to more closely match existing meta-analytic effects. These are future explorations which should be undertaken to refine the present model.

Other modeling possibilities

It is also possible that the models here could be combined with other theories to study group effects on training transfer. One intriguing example would be to incorporate Diffusion of Innovation Theory (e.g., Rogers, 2003) to study the propagation of transfer through a work group. As discussed regarding the reasons for the unexpected negative relationship between conformity and outcomes in Model 2C, a key to overcoming the momentum of the group is for individual agents to adopt the target behavior and slowly bring other agents on board. Eventually, a tipping point is reached where it becomes acceptable for the group to use the trained behavior where a critical mass will do so, then over time straggling agents will be pressured to adopt the new behavior instead of clinging to the old one. Conceptually this fits quite well with the diffusion of innovation, where research suggests new innovations are adopted in stages across populations. Initially, new innovations are adopted by only a few individuals, called innovators. Some individuals will follow these first adopters eagerly, representing early adopters but still making up a minority of the population. Once this group has opened the door, more and more individuals pick up the innovation in rapid succession as a critical mass is reached and soon most of the population uses the innovation. Eventually, only a few holdouts remain, representing the laggards in the population (see Rogers, 2003, for a broader overview). The conformity model here may inadvertently speak to this process in a transfer environment where the innovation is the newly introduced behavioral possibilities. Expanding on the proposed mechanisms and exploring others from the Diffusion of Innovation literature could provide new and useful insights into the social pressures governing transfer rates in organizations.

In addition, it would be interesting to pair this approach with network effects to understand how training adaptation might propagate across a network of employees. Social networks have gained much traction in organizational psychology of late (e.g., Soltis, Brass, & Lepak, 2018), but networks have long been of interest in related fields. One of the benefits of using NetLogo as the base for the present simulation is the existence of easily accessible network models which could be integrated with the present theory. For example, NetLogo comes with a model to study diffusion of information across directed networks (Stonedahl & Wilensky, 2008). Further, related theories could be drawn on to more broadly study the long run development of employees as suggested above, but in the ecological context of their work group as has previously been suggested with Ecological Systems Theory (Bronfrenbrenner, 1977, 1979) in understanding child development (Neal & Neal, 2013). By drawing on such existing modeling efforts we could greatly enrich our understanding of the social effects on the transfer of training. Such work would also provide potential extensions of recent research on vicarious learning mechanisms within teams (Myers, in press).

Implications for Practice

The combined effects of the above discussions and modeling here have some implications for practice as well. When it comes to promoting transfer, it is especially important to promote a climate wherein learners are free to explore their training and not succumb to group pressure to revert to pre-training behaviors. This is important to have established at the very end of training so that the dynamic effect of observing continued use of old behaviors and the pressure that use can create is broken to allow greater time for exploration. Once that climate is established, practitioners should encourage learners to observe the most effective individuals in their work groups and follow their lead. Through this sequence, one may be able to unlock the benefits of both types of social learning explored in the present models.

Conclusion

It seems apparent that the models explored in studies 2B and 2C are substantial improvements upon the LTM over both the baseline model, and the first attempt to include a social learning mechanism. These models have a stronger basis in the broader scientific literature and appear to conceptually fit with ways we think about groups in the organizational sciences. However, much work will remain to establish which of these mechanisms, or which mix of them, best accounts for transfer effects. For the time being, they appear to offer potential theoretical and practical insights and more work along these lines is encouraged.

Study 3A: Adding Self-Regulation to the Transfer Process Model

Unfortunately, the originally proposed social learning mechanism for the LTM did not operate as expected. However, it appears that the alternate models exploring conformity and imitation show greater promise for examining the effects of social groups in the transfer process. It was also argued that the conformity model fit better conceptually with the social environment effect we study through transfer climate, despite having the opposite direction effect initially expected. Therefore, the conformity model was used as a basis for a third iteration of the LTM which integrates a perspective that no within-person process model could, or at least should, avoid addressing – Self-Regulation (e.g., Vancouver, 2008).

Self-Regulation

Self-regulation is the dominant theory of motivation in organizational psychology (Vancouver & Day, 2005) and describes how individuals guide their actions towards goals over time (Karoly, 1993).

Hierarchical goal pursuit

Goals are internally represented desired states of being held by individuals (Lord, Diefendorff, Schmidt, & Hall, 2010) and are the central construct in self-regulatory systems. Goals exist in a hierarchy, such that individuals possess both short and long-term goals with short-term goals being nested underneath longer-term goals. As lower level goals are completed, individuals move closer to attaining higher-level goals. This hierarchical system is the basis of theories of self-regulation, and the application of self-regulation to organizational phenomena such as training transfer (e.g., Carver & Sheier, 1998; Powers, 1973; Blume et al., 2019). At each level of the goal hierarchy a self-regulatory system monitors goal progress and adjusts system outputs to maintain the desired goal level (Vancouver & Day, 2005).

Self-regulatory negative feedback systems

Self-regulatory systems are based around negative feedback loops which attempt to minimize discrepancies between the goal of the system and perceived progress towards it (Vancouver & Day, 2005). Two major versions of self-regulation exist, Social Cognitive Theory (SCT; Bandura, 1991; Schunk & Usher, 2012) which grew out of Bandura's (1977) Social Learning Theory (SLT), and Control Theory (CT; Powers, 1973; Carver & Sheier, 1998). Social Cognitive and Control versions of self-regulation do not differ all that substantially and make diverging predictions only in specific circumstances (Vancouver, Gullekson, Morse, & Warren, 2014). However, CT has the distinct advantage of relying on more formal forms of logic than SCT does, which is reliant on the narrative approach to theorizing while CT is more computational in nature due to its historical roots. As discussed previously, narrative theory is useful for conveying ideas, but suffers in making formal predictions (Vancouver, 2012; Adner, Polos, Ryall & Sorenson, 2009). Thus, because this paper is building a formal model of transfer and CT is a more formal version of self-regulation, CT will provide the basis of the present theorizing.

Many specific versions of CT exist (e.g., Powers, 1973; Campion & Lord, 1982; Carver & Sheier, 1998; Vancouver, 2008), but all have crucial elements in common. Broadly they all consider the nested goal hierarchy previously mentioned, but more specifically the basic negative feedback system of regulation relies on just a few key ideas. Lord and Hanges (1987) argue that the negative feedback systems of CT all have five elements: 1) some standard (goal) which the system seeks to maintain, 2) a sensor which monitors the state of the environment, 3) a comparator which compares the standard to the sensed environment, 4) a decision mechanism to decide whether something should be done to reduce any perceived discrepancy, and 5) an effector mechanism which produces some behavior meant to reduce the perceived discrepancy. As goal striving unfolds over time the regulatory system monitors progress towards that goal and works to lower discrepancies through multiple mechanisms. The two primary pathways to reducing discrepancies are 1) changing output behaviors that affect the perceived environment, such as increasing effort, and 2) changing the set-point of the goal in question (Campion & Lord, 1982). Within the CT literature it has been a point of contention on which option is most likely to occur, with some theorists arguing that goals are more easily adjusted, and that behavior is typically only changed when changing one's goal is not feasible (Lord & Hanges, 1987).

Self-Efficacy

Arising from the Social Learning and Social Cognitive perspective of Bandura (1977, 1991), self-efficacy represents the other central variable in self-regulation. Self-efficacy is the belief individuals hold regarding their ability to execute desired behaviors in the pursuit of some outcome (Bandura, 1977), and is the primary mechanism through which individuals exert agency (Bandura, 1977, 1989, 1991). That exertion of agency being represented in which environments people choose to enter (Bandura, 1989), and in choosing the tasks with which they will engage (Bandura & Cervone, 1987). Control theorists generally agree that efficacy is an important construct (Vancouver, 2012), but they dispute its nature, which complicates how efficacy will be instantiated in the present modeling.

Much research on self-efficacy followed the predictions of Bandura (1977, 1989, 1991), who describes efficacy as having a (nearly) uniform positive effect on performance. Through its influence on task and environmental choices, efficacy has a .38 meta-analytic relationship with performance between individuals (Stajkovic & Luthens, 1998). Theory and research on the within-person nature of efficacy is less uniform in its findings. In explicating their theories, multiple observers, including Bandura (1977), have discussed how extremely low levels of efficacy will lead to complete task disengagement as a cognitive defense mechanism (e.g., Lindsley, Brass, & Thomas, 1995). Additionally, Vancouver and colleagues have argued that the relationship between efficacy and performance is not always positive (Vancouver et al., 2014), and in a series of experiments have suggested that the relationship between efficacy and task engagement is discontinuous in nature. Specifically, at very low levels of efficacy individuals are unlikely to engage in a task at all, but as their efficacy increases, they will suddenly choose to engage in the task but will need to put forth maximum effort in order to succeed. Then, as efficacy continues to increase the individual will reduce effort as they do not feel full effort is required to ensure success, conserving those resources for when they are more necessary (Vancouver, Moore, & Yoder, 2008). The location of that discontinuity can be moderated by the nature of the task in question as well, such as by manipulating the value attached to said task (Sun, Vancouver, & Weinhardt, 2014).

The LTM with Self-Regulation

In some ways, motivational aspects of goal pursuit are already a part of the LTM. Sutton and Barto (2018) explain that internal state components of learning agents correspond to animal motivational states. In addition, learning agents have a built-in motivation to "ascend the gradient of its value function, that is, to select actions expected to lead to the most highly-valued next states" (Sutton & Barto, 2018, p. 361). In a normal reinforcement learning problem, the maximum attainable goal is defined by the environment and the ways in which the programmer encodes rewards. In a basic reinforcement problem, the agent will continuously attempt to improve its value states because it does not know what the maximum is. Humans, on the other hand, can decide that they have reached a goal and stop pursuing higher levels of attainment.

This ability to decide an acceptable level of performance has been reached and voluntarily avoid further improvement can be achieved through self-regulatory systems. There are approaches to modeling goal directed behaviors in learning agents seeking to navigate an environment in a typical reinforcement learning problem (Sutton & Barto, 2018), but those approaches are beyond what is required for the present purposes and the framework of a simple 2-armed bandit problem.

How can we account for specific goal directed behavior within the LTM? First, we must define a performance goal for the agent, *T*. To understand the relationship between performance and the agent's goal, we must define and track performance. Performance in this model will be represented by the variable *Y* and be calculated as the average performance across all task attempts, regardless of policy applied. In the present model, where any successful attempt is rewarded with 1 point, the average performance as percentage of successful attempts is equivalent to the average reward received on task attempts. Thus, *Y* may be calculated as:

$$Y_{t+1} = \frac{\sum_1^t R_i}{t}$$

Algorithm 7. Agent Performance

where *Y* at time $t + 1$ is equivalent to the average of all rewards received by the agent.

A comparison must then be made between the level of performance and the stated goal. This can be done through a simple difference variable we shall define as *D*, calculated as:

$$D_t = T - Y_t$$

Algorithm 8. Goal Discrepancy

The learner must then decide if they are short of their performance goal or not. This decision is defined by a variable *J*, equivalent to 0 if the goal is met, and 1 if it is not. That decision then feeds into an effector mechanism where if performance is short of the goal the agent chooses to change their behavior to reach it.

In other computational models of the self-regulatory system, the effector mechanism appears to be largely tailored to the task being modeled. In Vancouver's (2008) dynamic model of the self-regulatory system he defines the effector mechanism in terms of the actions taken to close the perceived gap between goals and perceptions where the acts taken are relevant to the goals being monitored. For example, if an overarching goal is to write a paper, the acts could be a series of steps such as doing research, outlining, etc., that are all governed by their own regulatory system. Given the undefined nature of the tasks modeled in this paper, it makes most sense then to define the actions an effector mechanism may take in terms of the current actions the learner/agents may take. Thus, the LTM hypothesizes that agents are more likely to explore their policy options if they are currently short of their performance goals. To account for this effect, let *F* represent the degree to which they are more likely to explore on a given task attempt. The variable *F* then modulates the exploration parameter *E* as a function of whether the agent is reaching their goal or not, resulting in the calculation of *E* as:

$$E_{t+1} = E_0 + FJ_t$$

Algorithm 9. Effector Mechanism 1

The newly added variables and equations for all of Study 3 can be found in Tables 11 and 12.

Study 3A: Method, Simulation, and Results

The outlined model for including self-regulatory mechanisms was instantiated in a new model in NetLogo. A screen capture of the modeling environment and code for this model can be found in Appendix E.

Virtual Experimentation

Although some key findings connected broadly to self-regulatory effects underlie transfer findings were already explored in experiments above, two key effects require illumination here. Namely the effects of goal setting and efficacy.

Following the nature of self-regulatory systems, it stands to reason that the higher the level goal set in a system, the higher we would expect the performance outputs of that system to be. A higher set goal creates a larger discrepancy between that goal and perceived reality. When individuals sense this discrepancy, they act to reduce it (Carver & Sheier, 1998). This basic finding, that higher goals lead to higher performance, is the essence of goal theory as outlined by Locke and his colleagues (Locke, 1968, 1975; Locke & Latham, 1990). Although it is often suggested that goals are specific and challenging yet attainable, even especially difficult goals can enhance performance if feedback regarding that performance is provided (Campion & Lord, 1982). Within the training literature, goal choice plays a key motivational role in the post-learning transfer phase (e.g., Beier & Kanfer, 2010). Post-training goals show a small meta-analytic effect on transfer outcomes ($\rho = .08$; Blume et al., 2010). To test for the effect of goal setting in the LTM, *T* was systematically manipulated. In addition, any discrepancies between observed performance and the set goal have effects on behavior in the present model through the level of exploration the agent is willing to engage in. There was no *a priori* expectation of which levels of change in the baseline exploration rate would result in expected relationships between goals and outcomes, therefore the level of exploration change, *F*, when short of the goal was simultaneously explored.

To explore the joint effects of goal level and change in exploration rate, a simulation was executed that crossed goal level, swept from 0 to 1.0 in .05 increments, against change in exploration from -.1 to 1.0 in .05 increments. *F* began at -.10 because the baseline exploration was held at .10, as established in prior simulations, and it was decided to cover the full range of final exploration rates. All other variables were held constant: likelihood of type 2 processing at .80, value of Policy A at .70, change in value at .05, initial policy estimates at .50, 100 pre-training time steps, and 500 transfer time steps.

Initial results of this simulation were surprising as the relationship between goal level and the outcomes of behavioral transfer ($r(241500) = -.126$, $p < .001$) and post training performance ($r(241500) = -.065$, $p < .001$) were negative at the replication level and the condition level ($r(483) = -.335$, $p < .001$, and $r(483) = -.326$, $p < .001$, respectively). Similarly, relationships between exploration rate change and behavioral transfer ($r(241500) = -.097$, $p < .001$) and post training performance ($r(241500)= -.050$, $p < .001$) were also negative at the replication and condition ($r(483) = -.249$, $p < .001$, and $r(483) = -.257$, $p < .001$) levels. Additionally, at the condition level, both goal level ($r(483) = -.259$, $p < .001$) and exploration rate change ($r(483) = -.234$, $p < .001$) were negatively related to the effect size of pre-post performance change. Given the surprising nature of these relationships, further analyses were completed in an effort to better understand them. Moderated multiple regression analyses showed the combined effects of goal level ($F(3, 241496) = 3503.46$, $p < .001$, $R^2 = .20$; $b_0 = .411$, $t = 783.48$, $p < .001$; $b_1 = -.110$, $\beta_1 = -.126$, $t = -63.48$, $p < .001$) and exploration rate change ($b_2 = -.077$, $\beta_2 = -.097$, $t = -48.69$, $p < .001$) had negative main effects on behavioral transfer, and a negative interaction ($b_3 = -.335$, β_3

= -.128, t = -64.11, p < .001). Similarly, in predicting post training performance, goal level (F(3, 241496) = 61.56, p < .001, R^2 = .53; b_0 = .721, t = 3725.10, p < .001; b_1 = -.005, β_1 = -.065, t = -8.40, p < .001) and exploration rate change (b_2 = -.004, β_2 = -.050, t = -6.41, p < .001) had negative main effects, and a negative interaction (b_3 = -.016, β_3 = -.067, t = -8.55, p < .001). Finally, in predicting pre-post performance change, goal level (F(3, 479) = 43.54, p < .001, R^2 = .46; b_0 = .289, t = 46.86, p < .001; b_1 = -.130, β_1 = -.259, t = -6.40, p < .001) and exploration rate change (b_2 = -.107, β_2 = -.234, t = -5.78, p < .001) had negative main effects, and a negative interaction (b_3 = -.460, β_3 = -.304, t = -7.50, p < .001). These interaction effects, showing the mutually depressive effects of these variables on our outcomes, have been graphed in Figures 33-35.

For further analysis, heat maps of these results were generated and can be found in Figures 36-38. These heat maps are especially informative as they show that traditional statistics are unable to fully describe the underlying data pattern. In these simulations, the heat maps suggest that goal level had no effect when it was below .70, that is when the goal was below the true value of Policy A. However, when the goal level was above .70, the effects of goals on the outcomes depended on the rate of exploration change. When the exploration change was very low, or very high, outcomes were worse than when exploration changes were low to moderate in magnitude. Thus, when short of their goal, it was beneficial for agents to explore to some degree, but not too much. The extreme negative effects when exploration changes were very low or very high may be masking the expected positive effect of goals. To test this, correlations between goal level and outcomes only across replications where the change in exploration rate was .10 were calculated. These relationships were indeed positive (r(21) = .105, p = .651, for behavior, r(21) =

.061, p = .793, for post training performance), which at least matches the expected direction of the effect of goal set point.

To explore the possibility that this model works within certain parameter ranges, a second simulation was run holding the change in exploration rate to .10 while sweeping goal level from 0 to 1.0 in .01 increments, 500 replications each. These results reveal a similar $r(21) = .095$, p = .682,relationship between goals and behavioral transfer, and an $r(21) = .05$, p = .830, relationship between goals and post training performance.

Study 3A: Discussion

The results of the initial exploration of the effects of goal level on the LTM did not match expectations. Instead of the small positive relationships expected between goals and transfer outcomes (Blume et al., 2010) the overall observed relationship was negative. However, it appears this negative effect is driven by especially bad outcomes when the change in exploration rate caused by being short of one's goal is especially high or low. The finding that such levels of change are detrimental does fit with previous findings in this paper in that especially high levels of exploration do not allow the agent to exploit the policies they do happen to find as more productive. On the other hand, especially low, and in this simulation negative, changes in the exploration rate represent a degree of disengagement from trying to transfer, and therefor would not result in positive transfer outcomes because the agent has in effect stopped trying to do so. When investigated further, it may be that the effects of goals in the model only work as expected within certain ranges of the change in exploration rate. This was shown as plausible in that the relationship between goal and behavioral transfer approximates the meta-analytic effect (Blume et al., 2010) when only examining the effect where the change in exploration rate is .10.

Given these findings, overall, it was concluded this is a plausible model provided parameters are held within certain ranges. However, prior to exploring the model further it was decided to investigate other potential mechanisms for translating falling short of one's goal into action effects to ascertain which may be more broadly applicable to the transfer environment.

Study 3B: Tweaking Goal Seeking

Prior to fully accepting Model 3A it was decided that at two other potential implementations of the self-regulatory system should be explored for the LTM. Model 3A relied on an effector mechanism that blindly makes the same adjustment to behavior regardless of the degree to which one is short of the desired goal. However, this does not necessarily align completely with reality. One other potential interpretation is that when individuals are further from their desired goal, they will take more drastic actions to close that gap. Leaving aside, for the time being, the issue of disengagement in the face of extreme deficits between goals and current states, an increase in motivation generally fits with the CT view of self-regulation where as an individual approaches their goal motivation would only be maintained through the increase of their goal level in order to maintain a deficit, or motivation would be redirected towards the completion of other goals (Carver & Sheier, 1998).

A second potential effector mechanism would be one which raises exploration as an individual nears their goal. Thus, when an individual is close to their goal but not quite there, they may work harder to find a way to push their current state to finally come in line with their desired one instead of backing off. This would imply an inverse relationship between the difference between one's current state and motivation instead of a positive relationship. Such a view fits with other theories such as the Temporal Motivation Theory (Steel & Konig, 2006) which states that the expected value of engaging in a task increases as the temporal distance to that task decreases, raising the motivation of the individual to engage in that task. Additionally, some research has shown that levels of motivation increase as subjective judgement of how close one is to their goals increases, and that close goals specifically increase focus on the process of meeting that goal (e.g., Peetz, Wilson, & Strahan, 2009). Such a relationship would fit with an effector mechanism that increases exploration when goals are close but unreached to a greater extent than when those goals are further away.

Given these two other possibilities for effects of goal deficits, two alternative effector mechanisms were explored for the LTM.

Model 3B-1

The first alternate effector mechanism proposes a direct link between the perceived difference between an individual's current state and their desired goal, and the degree to which they are willing to explore their behavioral options. Specifically, the degree to which they are short of their goal increases their desire to explore to the extent they see themselves as short of that goal. Mathematically this makes the variable *F* outlined in Model 3B to be a dynamic variable instead of a static one. Now, *F* will be calculated as:

$$F_{t+1} = D_t$$

Algorithm 10. Effector Mechanism 2

Stating that *F* at time $t + 1$ is equal to the observed difference, *D*, between one's goal and observed state at time *t*.

Model 3B-2

The second alternate effector mechanism proposes an inverse relationship such that exploration will be greatest when one is just short of the set goal, and that rate will taper off as the distance to the goal increases. The simplest way to create such a relationship is to simply adjust *F* to be

$$F_{t+1} = 1 - D_t$$

Algorithm 11. Effector Mechanism 3

However, this will create extremely large relative values of *F*, making exploration essentially 1 every time an individual agent is short of their goal. As seen in the simulations for Model 3A, this is not ideal or realistic. To place an upper limit on that change in exploration then it was chosen to calculate *F* as

$$F_{t+1} = .5 - D_t$$

Algorithm 12. Effector Mechanism 4

which will limit exploration rates to .5 + baseline defined exploration rate, which we typically are setting at .1, when individual agents are just short of their goals.

Study 3B: Methods, Simulation, and Results

The two mechanisms outlined above were instantiated into two mirrored computational models in NetLogo where the only difference is the calculation of *F*. A snapshot of the modeling environment and copies of the simulation code for each can be found in Appendix F.

Model 3B-1

To explore the effect of treating *F* as a direct positive function of the difference between one's goal and perceived state, an initial simulation swept the performance goal variable from 0 to 1.0 in .01 steps. Other variables were held constant: type 2 likelihood at .80, Policy A value at .70, change in value at .05, baseline exploration rate at .10, initial policy estimates at .50, 1 trainee, 100 pre-training time points and 500 post-training time points. Initial results suggest correlations between goals and behavioral transfer ($r(101) = .701$, $p < .001$) and post training performance ($r(101) = .392$, $p < .001$) are positive, as would be expected. To better understand the nature of the effect, graphic depictions were created of the mean observed behavior, post training performance, and pre-post performance improvement in Figures 39-41. These results show the relationship between goals and these outcomes is not uniform. Instead, goals have no real effect when goals are well below the set value of Policy A. Then, as goals approach and pass the value of Policy A, outcomes rapidly improve until leveling out once goals reach a level just higher than the value of Policy A. For pre-post performance change specifically, positive outcomes begin to occur around a goal level of .60.

Given the observed positive relationship between goals and outcomes in this model, and the potentially interesting effects of the semi-discontinuous nature of their effects, a small experiment was run to explore these effects further. Specifically, goal level (varied from 0 to 1.0 in .05 increments) was crossed with changes in value from Policy A to Policy B (varied from -

1.0 to 1.0 in .05 increments) to study the effect of changing goals against changing behavioral options. In this simulation, it was found that goals again had a positive effect on both behavioral transfer ($r(430500) = .638$, $p < .001$) and post training performance ($r(430500) = .032$, $p < .001$), as well as pre-post performance change ($r(861) = .094$, $p = .006$). In addition, value change also had positive effects on behavior ($r(430500) = .336$, $p < .001$), post training performance ($r(430500) = .589$, $p < .001$), and pre-post performance change ($r(861) = .611$, $p < .001$) as would be expected. A moderated multiple regression analysis was then completed. In predicting behavioral transfer it was found that goal level ($F(3, 430496) = 181108.91$, $p < .001$, $R^2 = .75$; $b_0 = .224$, $t = 688.04$, $p < .001$; $b_1 = .183$, $\beta_1 = .336$, $t = 331.21$, $p < .001$), and value change ($b_2 = .678$, $\beta_2 = .638$, $t = 629.60$, $p < .001$) had positive main effects, and a positive interaction ($b_3 = .351$, $\beta_3 = .196$, $t = 192.95$, $p < .001$). Similarly, in predicting post training performance it was found that goal level ($F(3, 430496) = 301747.87$, $p < .001$, $R^2 = .82$; $b_0 = .720$, $t = 6484.78$, $p < .001$; $b_1 = .128$, $\beta_1 = .589$, $t = 681.17$, $p < .001$), and value change ($b_2 = .013$, $\beta_2 = .032$, $t = 36.73$, $p < .001$) had positive main effects, although the effect of value change was very small after controlling for the effect of goal level, and a positive interaction ($b_3 = .411$, $\beta_3 = .574$, $t = 663.25$, $p < .001$). Finally, pre-post performance change displayed similar positive relationships with goals ($F(3, 857) = 908.08$, $p < .001$, $R^2 = .87$; $b_0 = .331$, $t = 6.32$, $p < .001$; $b_1 = 3.236$, $\beta_1 = .611$, $t = 36.57$, $p < .001$), and value change ($b_2 = .972$, $\beta_2 = .094$, $t = 5.62$, $p < .001$), and a positive interaction ($b_3 = 10.761$, $\beta_3 = .615$, $t = 36.82$, $p < .001$). These positive interactions are depicted in Figures 42-44. For further investigation, heat maps of these effects are depicted in Figure 45-47. As with the simulation of goals alone for this model, goal level had basically no effect on either behavioral transfer, performance, or performance improvement when goals were below about .60. When goals reach .60, there is a sudden and rapid change in the pattern of results where the best outcomes occur when goals are slightly above the baseline value of Policy A, and the change in value of the policies is moderately positive. If goals become too high, outcomes become worse as the agent begins to search for a better option than those available instead of exploiting the available options. In addition, outcomes are only especially bad when goals are high, and the new policy is substantially worse than the existing policy.

Model 3B-2

To explore the effect of treating *F* as a negative function of the difference between one's goal and perceived state, as with Model 3B-1, an initial simulation swept the performance goal variable from 0 to 1.0 in .01 steps. Other variables were held constant: type 2 likelihood at .80, Policy A value at .70, change in value at .05, baseline exploration rate at .10, initial policy estimates at .50, 1 trainee, 100 pre-training time points and 500 post-training time points. Initial results suggest correlations between goals and behavioral transfer ($r(50500) = .802$, $p < .001$) post training performance ($r(50500) = .307$, $p < .001$) are positive, as previously observed. Visuals were created of the mean observed behavior, post training performance, and pre-post performance improvement in Figures 48-50. These results also show the relationship between goals and these outcomes is not uniform but in a different way than with Model 3B-1. Here, goals still do not have a noticeable effect at extremely low levels, but they begin to impact outcomes at a lower level than in Model 3B-1. Additionally, their effect on behavior and performance does not come so suddenly and drastically. Instead, as goals increase behavioral transfer and post training performance gradually increase until leveling out around .50 and .73, respectively. For pre-post performance change we only begin to observe positive effects when goals reach at least .40.

Having found a positive relationship between goals and outcomes in this model, the same experiment run for Model 3B-1 was executed for this model as well. It was found that goals again had a positive effect on both behavioral transfer ($r(430500) = .123$, $p < .001$) and post training performance ($r(430500) = .315$, $p < .001$), as well as pre-post performance change ($r(861) = .094$, $p = .006$). In addition, value change also had post training performance ($r(430500) = .827$, $p < .001$), and pre-post performance change ($r(430500) = .611$, $p < .001$) as would be expected, but a negative effect on behavioral transfer positive effects on behavior ($r(861) = -.525$, $p < .001$). A moderated multiple regression analysis was then completed. In predicting behavioral transfer it was found that goal level ($F(3, 430496) = 218770.33$, $p < .001$, $R^2 = .78$; $b_0 = .347$, $t = 1372.82$, $p < .001$; $b_1 = .107$, $\beta_1 = .123$, $t = -547.17$, $p < .001$) had a positive main effect, but value change ($b_2 = -.233$, $\beta_2 = -.525$, $t = 127.98$, $p < .001$) had a negative main effect, and a positive interaction ($b_3 = .822$, $\beta_3 = .560$, $t = 583.56$, $p < .001$). In predicting post training performance it was found that goal level ($F(3, 430496) = 567084.97$, $p < .001$, $R^2 = .89$; $b_0 = .607$, $t = 4698.99$, $p < .001$; $b_1 = .196$, $\beta_1 = .315$, $t = 1208.04$, $p < .001$), and value change ($b_2 = .264$, $\beta_2 = .827$, $t = 459.89$, $p < .001$) had positive main effects, and a negative interaction ($b_3 = -.126$, $\beta_3 = -.119$, $t = -174.33$, $p < .001$). Finally, pre-post performance change displayed positive relationships with goals ($F(3, 857) = 844.33$, $p < .001$, $R^2 = .86$; $b_0 = -2.523$, $t = -29.02$, $p < .001$; $b_1 = 4.081$, $\beta_1 = .244$, $t = 48.20$, $p < .001$), and value change ($b_2 = 7.083$, $\beta_2 = .828$, $t = 14.21$, $p < .001$), and a negative interaction ($b_3 = -1.397$, $\beta_3 = -.049$, $t = -2.88$, $p = .004$). These interactions are depicted in Figures 51-53. For further investigation, heat maps of these effects are depicted in Figure 54-56. Interestingly, these analyses show that the best performance outcomes occur when value changes and goals are both high, which we would expect. However, a greater degree of behavioral transfer occurs when goals and value changes are low, counter to expectations.

Study 3B: Discussion

Models 3B-1 and 3B-2 were meant to explore other potential effector mechanisms within the self-regulatory processes of the LTM which are consistent with existing theory and research findings. This was undertaken after finding the originally proposed mechanism explored in Model 3A may only approximate meta-analytic estimates in the transfer literature under a limited range of parameters. The present models changed the value of *F* from an *a priori* set effect of being short of one's goal on one's rate of exploration to more dynamic conceptualizations based on perceived differences between one's goal and present state. Initial simulation results for these models are mixed.

First, Model 3B-1 did display the expected overall positive relationships between goal level and transfer outcomes of behavior and performance. This represents an improvement over the overall model of 3A, where initial results suggested overall negative effects of goals instead of positive ones. However, the magnitude of the goal effects in Model 3B-1 are much larger than the suggested .08 in transfer research (Blume et al., 2010). The same can be said of Model 3B-2, where relationships were in the expected directions, but of abnormally large magnitude.

Even so, there are some potentially intriguing results. For example, the finding that behavioral transfer rates actually reverse at very high goal levels in this model may be a sign of agents finding that a roughly 50-percent exploration rate is optimal given two behavioral choices as they desperately search for an option which may complete their goal. The combined lack of transfer at low goal levels, and this upper limit on transfer in this case may provide an explanation for low transfer rates commonly cited in the literature (Ford et al., 2010). For workers with low goals for their personal performance, all these models (3A, 3B-1, and 3B-2) suggest we will not see high degrees of behavioral transfer, although the exact amount differs by model. Further, when goals are very much higher than the achievable performance through available means transfer does not occur to an extreme extent because the agent does not just settle and acquiesce to use the best available policy, but keeps searching for an option which will fulfill their goals. For real world employees, the same calculus could be in play where among workers with high goals a failure to directly transfer received training may not be out of a failure to recognize the improvement of the training over whatever their old approach is, but represent a recognition that the training is not good enough and a result of their personal pursuit of other, not necessarily organizationally directed, options to achieve their goals. Such an insight could provide guidance to future research projects.

Further, this set of models may provide practical guidance on the post training setting of goals. Following goal theory (Locke & Latham, 1990), goals for transfer are set following training to, ideally, be specific, challenging, and attainable. The relationship between those goals and actual transfer could be said to be disappointing given the weak meta-analytic relationship between goals and transfer (Blume et al., 2010). The findings here suggest that we may need to focus more on those post training goals being attainable to keep them in the range where they can have a substantial effect on later transfer. Further, on the research side, when we study those goals, we may need to change the way we analyze their effects. We traditionally rely on ordinary least squares regression and correlational approaches to study these effects, but it has been suggested that more advanced analytic techniques could improve our understanding of transfer (Olenick et al., in press), and the effect of goals on that transfer is a good example. In selection research it has recently been shown that taking a fit approach and associated polynomial regression techniques can greatly improve the ability of interests to predict work performance (Nye, Prasad, Bradburn, & Elizondo, 2018). Similarly, given the interplay between goal levels and the value of the received training indicating an interplay where different goals may work better with different levels of value for the trained behavior, we could study the congruence between set goals and the value of the training. In this way we may better estimate the value of goal setting within the training and transfer field.

Despite the potential insights gained from these models in total, the results of the simulations run for this paper in studies 3A and 3B suggest the best model for use in transfer may be that proposed in Model 3A. This conclusion results from the very close match between simulated effects of goal level, provided the model is within certain ranges of parameters, for Model 3A while the effects observed in Models 3B-1 and 3B-2 are larger than the meta-analytic effect of .08 (Blume et al., 2010). Further, the mechanism tested in Model 3A is more parsimonious than those tested in 3B-1 and 3B-2, and it provides a degree of control for further simulation. The approach Model 3A takes is one more akin to treating not just goals, but the effects those goals have on decisions as an individual difference within the transfer environment, adding to existing studies of individual differences such as personality, goal orientations, need for cognition, and implicit theories of learning (e.g., Jaeggi, Buschkuehl, Shah, & Jonides, 2014). Given the results showing Model 3A replicates the effects of goals on transfer closely, provided F is set to plausible ranges, and the potential it implies for future research, Model 3A was retained for use in the full LTM. However, it is acknowledged that future work will be required to explore this and other effector mechanisms, especially in regard to applying the model to any particular task of interest.

Study 3C: Engagement Thresholds

Having established that the originally proposed self-regulatory system in Model 3A generally outperforms two other alternatives, shown in Models 2B-1 and 2B-2, in recreating regulatory effects in transfer research, another set of self-regulation findings and implications was explored. Thus far in exploring self-regulation in training transfer we have focused on the effects of goals. Now, we must explore the effect of self-efficacy, which has long been a central variable in self-regulatory models.

Self-efficacy, the belief in one's abilities to accomplish a given task (e.g., Bandura, 1977), has been argued as the central motivational variable by which individuals have agency over their environments (Bandura, 1989). Decades of research have established a clear general pattern of higher efficacy relating to higher task performance (Stajkovic & Luthans, 1998), and this effect has been meta-analytically established in training transfer where post training efficacy has a corrected relationship with transfer of .22 (Blume et al., 2010). However, in the last decade, some minor but important disagreements have arisen over the nature of self-efficacy. Importantly, it has been argued that when studied in a causal manner self-efficacy is a product of performance, and not necessarily the other way around. In this case, Sitzmann and Yeo (2013) found that within individuals performance predicted self-efficacy at $\rho = .30$ when controlling for linear trajectories, but self-efficacy only predicted performance at $\rho = .06$ under the same conditions. Thus, it would be beneficial for the present model to display a general positive relationship between efficacy and transfer, and performance, but also replicate the differences in the causal strength of the efficacy-performance relationship.

Further work by Vancouver and colleagues has challenged the traditional view of self-efficacy having a monotonously positive effect on important outcomes such as performance and task engagement. Over various studies, they have found that self-efficacy can have negative effects in learning tasks under some conditions (Vancouver, Gullekson et al., 2014), and that the relationship between efficacy and task engagement is actually discontinuous in nature (Vancouver et al., 2008; Sun et al., 2014). This discontinuous relationship suggests that at very low levels of efficacy for a task, individuals will refrain from engaging in that task and instead conserve their resources for tasks they are more confident in. As efficacy levels for a task increase, eventually a threshold is passed where suddenly those individuals will choose to engage in the task and will outlay substantial resources in order to improve their odds of success. In the transfer environment it is possible that learners would not even attempt to transfer their learning if they do not believe they can succeed at the application of that learning, which would drastically reduce transfer rates and provide another potential explanation for the common belief that transfer rates are disappointingly low. To the author's knowledge no studies have examined the effect of efficacy on training transfer from a discontinuous perspective. Therefore, the present model will explore the potential effects of a discontinuous model of efficacy on transfer to guide future research.

Discontinuous Self-Efficacy in the LTM

As currently conceived, the LTM and its computational equivalent does not directly incorporate a variable labeled efficacy. However, since efficacy is a perception of the individual regarding their ability to complete a task (Bandura, 1977), and that efficacy is the product of past performance (Sitzmann & Yeo, 2013), the equivalent of an efficacy evaluation is already present within the LTM. The underlying value of each policy which the learner may apply to their encountered situation is the percentage probability of that policy succeeding in that situation. The learner develops an estimate of the policy's value as they attempt to apply that policy and receive feedback to inform that estimate. Thus, their estimate of the policy's value is their efficacy because it is their estimate of the likelihood of their succeeding at applying the policy. Therefore, nothing needs to be directly changed in the existing model to incorporate efficacy as a construct.

However, as discussed, the relationship between efficacy and task engagement is not actually linear (Vancouver et al., 2008; Sun et al., 2014). As it exists in the present model, the likelihood of using any policy available to the learner is a positively linear function of the estimated value of that policy. That is, even though the exact choices made by an individual on a given task attempt is dependent on several dynamic variables, the underlying relationship is that as the estimated value of a policy increases the likelihood of using that policy will increase. If the relationship between efficacy and engagement is non-linear, then the existing underlying relationship is incorrect. To remedy this, a single variable needs to be added to our overall model. We will call this variable the *engagement threshold*, labeled V, and will represent the value estimate below which the learner will not choose to implement that policy and will instead opt for the other policy available to them. As tasks are encountered and policy decisions are made by the agents, each learning agent in the model independently compares their value estimate of that policy to the cutoff level defined by V. If that policy has a lower value estimate than that threshold level the agent will choose the other policy, but only if the other policy option lies above the threshold, otherwise the original policy choice will be implemented.

Study 3C: Methods, Simulation, and Results

The addition of an engagement threshold and necessary code to ensure agents only applied behaviors above that threshold when possible was made to the expanding computational model of the LTM. A screen capture of the modeling environment and associated code can be found in Appendix G.

Causal Effects of Self-Efficacy on Transfer and Performance

Since efficacy is a value in this model which develops of its own accord, the effects of efficacy were not explored via direct manipulation. Instead a model was executed which tracked the level of the estimated value of Policy B over time to attain a measure of the agent's efficacy for that policy. Given the mechanics of the model as presented in this paper, it was expected that the value estimate will be related to outcomes of interest in the way efficacy is found to be in the psychological literature. Specifically, the estimated value of a policy will be positively related to the likelihood that the learner will choose that policy on a given task attempt and thus transfer it to the task from their theoretical learning environment, and the perceived value of the policy will be positively related to task performance (Blume et al., 2010; Sitzmann & Yeo, 2013; Stajkovic & Luthans, 1998).

To study the dynamic relationships between efficacy, performance, and transfer, 1000 replications of a model with one agent were run for the established 500 time point transfer length. At each time point, the value estimate of Policy B, whether or not Policy B was applied at that time point, and the reward (representing task success or failure) for that time point were saved. All other variables were held constant as before, with type 2 likelihood at .80, value of Policy A at .70, change in policy value of .05, exploration rate of .10, and starting policy value estimates of .50.

To analyze this data, correlations were computed between the saved variables, adjusting the data set to account for causal ordering. Only time points within the transfer period were analyzed to remove any biasing effects of the data regarding Policy B during the pretraining phase when those values were not affecting the behavior of the agent. Additionally, only the value of Policy B is of interest as it is the target of transfer and therefore the subject of the efficacy measurements typically taken at the end of training. First, the estimated value of Policy B at each time is causally related to performance at that time point. The relationship between these two variables was found to be $r(500000) = .065$, $p < .001$. This relationship nearly perfectly replicates the relationship found by Sitzmann and Yeo (2013) for the same effect. Next, the value of Policy B was related to the tendency to choose Policy B at that time point, representing behavioral transfer. This relationship was found to be $r(500000) = .369$, $p < .001$, which is in the correct direction for transfer as found by Blume et al. (2010).

Finally, a lag variable was required to test the effect of performance to align performance on one task attempt with the value estimate of Policy B on the next attempt. When Policy B is chosen at time t, the resulting performance at that time point should have a causal relationship with the value estimate of Policy B at time $t + 1$. To isolate these effects, only time points where Policy B was applied in the transfer environment were analyzed. Among these time points, the relationship between performance and the value estimate of Policy B was $r(207936) = .048$, $p < .001$, which is in the expected direction according to meta-analytic estimates, but substantially less in magnitude (Sitzmann & Yeo, 2013). It is possible that the length of the transfer run obfuscates the relationship between these two variables as the value estimate of Policy B stabilizes over time and therefore would not be greatly affected by a single performance. Within most research studies we are unable to study any length of time close to 500 data points long and instead the dynamic relationship between efficacy and performance is based on a much shorter time period. To test if a shorter time period would better approximate the expected relationship, the correlation between performance and the value estimate of Policy B on the next time point was estimated for both the first 100 transfer attempts, and the first 25 transfer attempts. In the first 100 transfer attempts the relationship was $r(41587) = .064$, $p < .001$, and was $r(5199) = .056$, $p < .001$, in the first 25. These relationships suggest it is not merely the time period examined which accounts for the difference between the meta-analytic relationships and those generated by the present model.

Effects of Engagement Threshold

Unlike with the general effects of our efficacy stand-in, the policy value estimate, an experiment was completed to explore the effects of the discontinuous model of efficacy as it applies to transfer. To test the effect of the *engage* variable, V, a simulation swept the parameter from 0 to 1.0 in .01 increments. It was expected that transfer would diminish as the threshold level of *engage* increases. The logic of that relationship being that not only will it be more likely for the value estimation of the target policy to fall below that threshold overall, that effect is exacerbated by the instability of small samples where even high true values for policies will often have lower estimates of that value in the initial stages of transfer only because of sampling error. This incorrect early judgement would sometimes result in the abandonment of a policy before its true value is revealed to the learner. Such an effect would seem logically consistent with experience given that some learners will not apply their new KSAO because they feel it is too difficult. As such, this also represents an initial relaxation of the assumption that learners enter the transfer environment with the ability to successfully apply their new KSAO.

In running the full parameter sweep of the engagement threshold, all other parameters were held constant at our established levels: type 2 likelihood was set to .80, .10 exploration rate, true value of Policy A .70, a .05 change in value to Policy B, 100 pre-training time points, 500 transfer time points, 500 replications each, with one agent in each model. However, unlike previously, the initial estimate for the value of Policy B was set to 1.0 instead of .50. This change was made to refrain from artificially limiting initial transfer attempts by a parameter which in this simulation was not our focus, instead allowing any reluctance from the agent in applying Policy B to arise from its own experience.

Initial examination of the results of this experiment confirmed expectations outlined above. The relationship between threshold level and behavioral transfer ($r(50500) = -.365$, $p < .001$), and post training performance ($r(50500) = -.214$, $p < .001$) were both negative. To further understand the relationship between the engagement threshold and transfer outcomes, mean results for each condition for behavioral transfer, post training performance, and the effect size of pre-post training performance change have been plotted in Figures 57-59. In addition, best fitting trend lines with a quadratic term were plotted to better visually illustrate the general pattern. The pattern of all these findings indicate that transfer outcomes are relatively high when the engagement threshold is low. However, when the threshold reaches about .50, transfer outcomes begin to deteriorate rapidly as they transition to a lower set point starting around .80 where those outcomes display essentially no transfer. In addition, performance change as expressed in Cohen's *d* becomes negative when threshold levels exceed about .60, which is well below the .75 true value of Policy B.

Study 3C: Discussion

The present study explored the effects of self-efficacy within the LTM. Specifically, it suggested that the value perceptions for the behavioral policy representing the targeted transfer behavior would display relationships with outcome variables that have been observed in the literature. Further, it explored the effects of the discontinuous model of self-efficacy (e.g., Vancouver et al., 2008) on transfer. Here I shall discuss the implications of these simulations for both theory and practice.

Theoretical and Research Implications

Overall, the effects of the value estimate for Policy B in the present model continue to be mixed. As you will recall, it was argued in a previous simulation that the effect of the initial value estimate for Policy B should approximate the effect of utility reactions we observe in the transfer literature (Blume et al., 2010). However, the expected relationship did not emerge at the replication level, but did to some degree at the condition level, leaving the support for the expected effect as plausible but needing some future refinement. Similarly, the results for the effect of and on Policy B value estimates were mixed in the present study. On the one hand, all the relationships between Policy B value estimates, transfer, and performance were in the expected direction. In addition, the magnitude of the causal effect of the policy estimate and performance was essentially identical to that observed in the research literature (Blume et al., 2010). Thus, it could be argued that the general pattern of results from this model fits that which was expected, and generative sufficiency has been achieved.

In addition, the general effects of the inclusion of a discontinuous effect for “efficacy” in the present model fit expectations. Overall, the effect of having a threshold for when to apply a given policy was such that high thresholds decreased behavioral transfer and performance outcomes. Unfortunately, there are no known studies to which the effect observed here can be directly compared, although the observed effect fits with general expectations from the work by Vancouver and colleagues on the nuanced effects of self-efficacy. However, we cannot directly compare the effect sizes observed here to theirs to enhance the claim of generative sufficiency as the tasks used in their work are not transfer related, nor do they collect data in a comparable way. For example, Vancouver et al. (2008) use a task called the Hurricane Game where participants must click on squares of various sizes, representing different levels of efficacy for doing so, as they randomly jump around a computer screen. There is no real learning component to this task, and they do not collect and report data on the behavioral strategies employed by their participants to compare how those strategies to each other. Therefore, future work is needed to apply the present simulation to more applicable learning and transfer related tasks which are designed to study the discontinuous nature of self-efficacy.

Along with applying the present theory to more directly comparable data, the nature of the discontinuous effect of self-efficacy in the present model needs to be further tuned and explored. As implemented in this version of the LTM, effort is assumed to be constant across all levels of self-efficacy if the agent has decided to engage in the targeted behavior. That is, the agents either fully engage with the behavior or they do not. The discontinuous model of self-efficacy (Vancouver et al., 2008) suggests that this is not quite the case. The discontinuous model does suggest that individuals completely disengage from tasks which are below that individual's threshold level, as modeled in the LTM, but above that threshold there is a negative relationship between efficacy and effort. In their studies on this phenomenon (Vancouver et al., 2008; Sun et al., 2014), Vancouver and colleagues use time allocation as a measure of effort applied to the task, but in the LTM it is currently assumed all resources are applied as long as the threshold is met. Future iterations of the present model should examine the effects of resource allocation to relax the assumption that individuals always fully engage or do not and explore the impact of a tapering off resource allocation by agents at high levels of efficacy. It could be the case that very high levels of efficacy are then detrimental to transfer while the highest levels of transfer occur when efficacy is just high enough to get a learner to engage. Such a finding would provide a potential explanation for the surprisingly low relationship found in the literature between efficacy and transfer (Blume et al, 2010) as the negative effect of high levels of efficacy would mask its overall benefits.

One surprising outcome of the discontinuous effect of the threshold model explored here is worth some discussion. Specifically, although expected to a lesser degree than was observed, it is surprising to see the threshold have negative effects on transfer at levels so far below the true value of Policy B. The reason for this likely has to do with sampling errors by the agents. In the early stages of transfer, the value estimate for Policy B can fluctuate quite wildly as the agent does not have much experience with that policy. On the other hand, even in early transfer attempts the same agent has at least 100 experiences with Policy A and therefore already has a relatively stable and accurate estimate of the value of Policy A. This results in a situation where in early transfer attempts the agent will have a good idea of the true value of Policy A, and therefore their theoretical efficacy for that behavior as it has been argued that the value of the policy and efficacy are equivalent in this model, and if that true value is above the threshold where they are willing to use that behavior. Simultaneously, they are unsure of the true value, and therefore their efficacy, for Policy B and just a couple poor experiences with Policy B can easily lead to their value estimate falling below the threshold and them discarding the policy before ever truly giving it a fair chance. It is worth noting that this discarding of Policy B based on these experiences again fits with general predictions of recent narrative theorizing around the transfer process (Blume et al., 2019). The negative effect of the threshold then occurs at a lower level than the true value of Policy B because even relatively lower thresholds will sometimes lead to the agent erroneously discarding Policy B based on few experiences. Combined with this effect, sometimes the value of Policy A will be overestimated based on pre-training experience, making it even less likely the agent will decide to transfer Policy B. Then, in the transfer environment, that agent discards Policy B only to potentially learn over time Policy A is not as valuable as it believed, and the overperformance of Policy A observed in the pre-training environment will tend to even out over the force of the extra time simulated in the transfer environment. This over-estimation then correction likely explains the observed negative effects seen in pre-post training performance comparisons here. Despite the initially surprising nature of this effect, it would again help explain the general belief in low levels of training transfer if individuals are giving up on that transfer in part due to a misreading of the benefits of their training compared to their personal willingness to employ that training.

Overall, given these results, the present model is potentially viable for studying the basic patterns of relationships we might expect in the transfer environment. Future research should fine tune the way in which the policy value estimates operate to better match real-world observations. Or, the model will require further exploration to understand under what parameter combinations the expected relationships may be reproduced. For example, it could be that when the difference in policy values from Policy A to B are even smaller than .05, the relationship between the value estimate and transfer may similarly decrease as the agent would erroneously choose to apply Policy A more often. However, this would also decrease the overall rate of transfer and potentially move the model out of acceptable ranges in other ways.

Practical Implications

The interesting finding that negative transfer outcomes begin at threshold levels well below the true value of a trained behavioral policy has significant implications for how we approach transfer in real organizations. In the present simulations we see that it is possible to exhibit negative training outcomes even when a trainee's personal threshold for trying a new behavior is well below that theoretically required for them to do so. Therefore, in our training interventions we should take extra care in ensuring trainees are willing to try their new training back on the job multiple times before judging whether to retain or discard it for future use. This could include measures taken within the training program itself, such as providing examples of the training working to provide evidence that it should be useful, or during the transfer phase such as check-ins on their progress and supervisor support *early in the transfer process* before the learner has a chance to discard the training as not being useful.

Conclusion

The model explored in this study represents the final iteration of the LTM for the present paper. Given the pattern of observed results it appears the model can be defensibly applied to the study of training transfer as it is able to largely reproduce expected patterns of relations and results. However, more work will need to be done in the future to fine tune aspects of the model to better fit existing data. For now, the model appears to be a useful first step towards accounting for transfer effects with a dynamic process theory, and that the model could provide potentially novel and useful insights for future research and practice.

Study 4: Exploring the Full LTM Model

Over the course of the present paper, we have explored several iterations of a process-oriented theory of learning transfer called the Learning Transfer Model. This evolving theory was instantiated in a series of computational models and explored to establish generative sufficiency for existing research findings in the transfer literature. Based on the simulations presented here, it appears that this process has largely, though not completely, been successful. However, work is not yet done. One strength of computational models is the ability to run novel experiments in a low-risk environment to provide insights for theory and practice that would not normally be feasible, if not completely impossible, in a traditional research environment. Therefore, the final study of this paper takes advantage of the developed modeling platform to demonstrate some of the types of experiments that can be conducted in this environment and discusses some of the implications of those findings. The experiments executed here were chosen *a priori* for the apparent potential novelty of the effects that we do not typically study in the transfer literature, as well as their ability to demonstrate effects we may not be able to easily study in real world environments without prior guidance.

Experiment 4A: Engagement Thresholds, Value Changes and Implementation Intentions

The first exploratory experiment pitted level of engagement threshold, value changes, and implementation intentions against each other. We saw in Study 3C that engagement thresholds have an overall negative effect on transfer outcomes with a rapid change in outcomes as those thresholds approach the values of the available behavioral policies. One possible implication of this finding is that thresholds for trainees need to be surprisingly low to ensure positive transfer outcomes given the trained KSAO. On the other hand, it might suggest that individuals with especially high thresholds for engagement would require especially valuable new KSAOs from a training event to ensure their successful transfer outcomes. In part, the present experiment explores the tradeoffs between these two factors in order to guide decisions regarding training for individuals based on their likely willingness to engage with the given task using their trained KSAO, and the theoretical performance value of that KSAO.

However, one way to overcome the reluctance to engage the task with the trained KSAO may be to pair that training with implementation intentions to make the response more automatic (Gollwitzer & Sheeran, 2006). The positive effects of implementation intentions were demonstrated in Study 1. It was expected that implementation intentions would reduce the negative effects of thresholds on transfer. It was further expected that implementation intentions would have a larger effect on transfer outcomes when engagement thresholds are high, but the value of improvement for the new policy is low. This was expected because when the value of the new policy is already high it should more often be able to overcome the threshold without the need for the extra intervention of implementation intentions.

Methods

To explore these effects, a three-way experiment was designed using the computational version of the LTM settled upon in Study 3C. To limit the number of runs required, the range of parameters simulated were limited more to ranges where effects were most salient in previous simulations. To this end, engagement threshold was limited to the range of .50 to 1.0, swept in .05 increments; implementation intentions were swept in .05 increments from 0 to .05; and value change to Policy B was limited to -.10 to .30, in .05 increments. Other variables were held constant as before, with the true value of Policy A being .70, type 2 likelihood of .80, pre-training time points of 100, with 500 transfer time points, but initial policy value estimates were again set to 1.0 to ensure no artificial limiting of transfer due to the threshold variable, one agent was simulated in each run. 500 replications for each condition were created.

Results

Initial analyses suggest the effects of all three variables explored here are in the expected direction across replications on our outcomes of interest. Implementation intentions had small but positive relationships with behavioral transfer ($r(297000) = .026$, $p < .001$) and post training performance ($r(297000) = .020$, $p < .001$), while changes in policy value had substantial positive relationships with both behavioral transfer ($r(297000) = .649$, $p < .001$) and post training performance ($r(297000) = .721$, $p < .001$). On the other hand, engagement thresholds are negatively related to both behavioral transfer ($r(297000) = -.283$, $p < .001$) and post training performance ($r(297000) = -.168$, $p < .001$). Given the nature of the present experiment, it is not advisable to interpret these correlations for their strength as the targeted conditions could be either enhancing or truncating them, but it is notable that they are in the expected directions. Next, moderated multiple regression analyses were completed predicting behavioral transfer, post training performance, and the effect size for pre-post performance change from the three-way interaction of implementation intentions, engagement thresholds and value change. The resulting parameters for these models can be found in Table 13, and graphs of the interactions in Figures 60-62. Heat maps were then generated at the condition level to explore these effects further and can be found in Figures 63-65. These analyses reveal that when the value of a policy is low, transfer is generally poor unless the threshold for engagement is low and implementation intentions are high. Such a pattern is okay though, because in the case of low value we generally actually do not want transfer to occur, unless there is a non-performance reason to do so, as it will reduce performance. When the new policy has a high value, the effect of threshold level dominates the rate of transfer such that low threshold levels are very beneficial and high levels are very detrimental. Beyond the effect of thresholds, having strong implementation intentions only have a noticeable affect when thresholds are already low. Patterns of results for both post training performance and pre-post training performance change are similar.

Discussion

The results for this experiment were somewhat surprising, especially when it came to the effect of implementation intentions. It was expected that implementation intentions would have a stronger effect when policy values were low, but threshold levels were high, essentially acting as a way to overcome the detrimental effects of high engagement thresholds. This was not the case. Instead, implementation intentions showed their strongest effects when engagement thresholds were already low, suggesting implementation intentions did not act as a way to overcome a reluctance to engage in the task, but rather enhance one's ability to engage in a task when one is already willing to do so. This is the type of surprising finding that a model such as this can put forth to guide future research and opens the model to falsification. If this unexpected finding holds up to further scrutiny it would suggest that in designing training events one should first focus on encouraging trainees to lower their engagement threshold before worrying about the use of implementation intentions. We know implementation intentions are generally effective additions to training events (Friedman & Ronen, 2015), but their use may be for naught if our learners are unwilling to engage with the trained KSAO anyways.

Experiment 4B: Number of Trainees, Conformity, and Goal Levels

A primary strength of the modeling platform built in this paper is the ability to explore social effects on transfer outcomes without requiring the hundreds, or even thousands, of individuals that would be required just to explore these ideas using real world data. This allows us to look for potential effects of interest from the theory and use that modeling to guide future targeted data collections and utilize our limited resources more judiciously. To this end, the rest of the exploratory simulations discussed here focus on the social effects of the conformity mechanism established in Study 2C. The simulations in Study 2C showed that high levels of conformity were extremely detrimental to transfer outcomes, especially after the number of agents reached about 3 or 4. One possible way to overcome the pressures of the group to conform is for individuals to have higher goals that will lead to them exploring behavioral possibilities more even in the face of that pressure. To test this possibility, the initial simulation from Study 2C crossing number of trainees with level of conformity was extended to include an effect of goals which was introduced in Study 3A. It was expected that conformity would still have a negative effect, especially as the number of trainees increased, but that negative effect would be tempered by increased goals.

Methods

The final model from Study 3C was again used to conduct this exploration. Trainees were swept from 1 to 20 in 1 trainee increments, conformity from 0 to 1.0 in .05 increments, and goals from 0 to 1.0 in .05 increments. Other variables were held constant as before, with the true value of Policy A being .70, type 2 likelihood of .80, pre-training time points of 100, with 500 transfer time points, and initial policy value estimates of .50. 500 replications were completed for each condition.

Results

Initial results largely produce expected relationships between variables of interest here and behavioral transfer and post training performance across all replications. The number of trainees in the model was negatively related to both transfer ($r(4410000) = -.199$, $p < .001$), and post training performance ($r(4410000) = -.145$, $p < .001$). The same was found for the relationships between conformity and transfer ($r(4410000) = -.766$, $p < .001$) and post training performance ($r(4410000) = -.556$, $p < .001$). However, goal levels were positively related to both transfer ($r(4410000) = .093$, $p < .001$) and post training performance ($r(4410000) = .068$, $p < .001$). To further understand these simulated effects, multiple moderated regression analyses were completed testing the three-way interaction of trainees, conformity, and goals on behavioral transfer and post training performance. Parameter estimates for these models can be found in Table 14. Additionally, graphic depictions of these interactions can be found in Figures 66 and 67. Additionally, heat maps of these results at the condition level are depicted in Figures 68 and 69. Due to the misleading results with changing numbers of trainees observed in previous simulations, effect sizes for pre-post performance change were not computed for this experiment. The general effect of conformity in this experiment is identical to that observed in Study 2C where conformity levels above about .45 largely eliminate the transfer of the new policy. However, we do see that goals have an effect where they essentially push this boundary slightly higher, such that it now occurs around .50 conformity. We also see an example of a potentially misleading result when relying on only traditional methods to examine these results, where the regression model and simple slopes analysis suggests an effect of the number of trainees such that fewer trainees is very detrimental when goals are low, but more trainees are detrimental when goals are high. When we examine the heat maps of the results instead, we see that the effect of the number of trainees across levels of goals is largely the same and this apparent interaction should not be over interpreted.

Discussion

In previous simulations in this paper, we saw that the social pressure of one's work group severely depressed transfer outcomes once the degree of conformity reaches about .45. The likely reason for this is that the default behavior is to not transfer, so the pressure to follow along at the next time step will tend to keep transfer low. Once conformity is low enough to allow exploration the agents are much more likely to explore and discover the benefits of their training and therefore begin to transfer. What we see that is new here is a tempering effect of high goals on the depressive effect of conformity. Specifically, it appears that high goals shift the sensitive area between failure to transfer and where transfer begins improving from a conformity level of about .45 to about .50. This is a small but potentially very important effect suggesting that good goal setting may help push some individuals who would otherwise be on the fence regarding successfully transferring their training back to their work environment towards overcoming the pressures of the social world around them and doing so.

Experiment 4C: Value Change, Conformity, and Goal Levels

Another way to potentially overcome the negative effects of conformity on transfer outcomes would be to improve the performative value of the newly trained KSAO represented by Policy B. Doing so should provide extra incentives initially for individuals to break from their work groups and begin using their newly trained KSAO. Then, the pressure to conform should benefit high-valued KSAOs once transfer has begun to spread that tendency quickly through the group and improve overall outcomes. Similarly, especially low-valued KSAOs should quickly be discarded by the group in favor of keeping their old KSAO in place. Therefore, it is expected that outcomes will be made more extreme, positively and negatively, by different levels of value change. It is also expected that the positive effects seen when values are high will be further enhanced when goals are moderately high due to the increased exploration undertaken by agents, but not when goals are so high individuals are unwilling to exploit the better policy once it is found.

Methods

The final model from Study 3C was again used as the base model to conduct this exploration. Goals were swept from 0 to 1.0 in .05 increments, conformity from 0 to 1.0 in .05 increments, and three levels of value change at -.10, .05, and .20. These conditions for value change provide equidistant conditions of one negative behavior we should want the agents to discard, one representing the typical change we have discussed throughout this paper, and one especially beneficial training event. Other variables were held constant, with the true value of Policy A being .70, type 2 likelihood of .80, pre-training time points of 100, with 500 transfer time points, and initial policy value estimates of .50. 500 replications were completed for each condition. However, given the results from the exploration in 4B, and previous simulations of the number of trainees in the model in Study 2, it was decided to choose a constant number of agents for the simulated work group. Based on those results, it was decided to simulate groups of 3 as it appears that results pretty well stabilize once this number is reached. Limiting the simulation to 3 agents also has the benefit of being large enough to be traditionally considered teams (Tannenbaum, Mathieu, Salas, & Cohen, 2012), but go beyond the study of dyadic relationships. In addition, limiting the teams to 3 instead of a larger number would reduce the burden on participant recruitment for any future attempts to apply the results of the present simulations to empirical investigations.

Results

Findings suggest the effects of all three variables explored here are generally in the expected direction across replications on our outcomes of interest. Value change was positively behavioral transfer ($r(661500) = .533$, $p < .001$) and post training performance ($r(661500) = .711$, $p < .001$). Conformity had negative relationships with both behavioral transfer ($r(661500) = -.670$, $p < .001$) and post training performance ($r(661500) = -.369$, $p < .001$). However, goal level showed a positive relationship with behavioral transfer ($r(661500) = .044$, $p < .001$), but a negative one with post training performance ($r(661500) = -.016$, $p < .001$). Given the small nature of this negative relationship and the possibility of negative interactions with the other variables here, this finding should not outweigh the other effects of goals observed in this paper. Moderated multiple regression analyses were completed predicting behavioral transfer, post training performance, and pre-post performance change from the three-way interaction of conformity, goal level, and value change. The resulting parameters for these models can be found in Table 15, and graphs of the interactions in Figures 70-72. Heat maps were then generated at the condition level to explore these effects further and can be found in Figures 73-75. As before, high levels of conformity have substantial negative effects on transfer outcomes. It also does not appear in the regression analysis that high policy values can overcome those negative effects of conformity, but we do potentially gain some nuance on the effects of goals and see they have a slight effect only when both conformity and value changes are low. In examining the heat maps, we gain a greater understanding of the effects, especially an effect such that when value change is especially low, behavioral transfer is best when goals are high. But when values are high the best transfer occurs when goals are lower. We see the opposite essential pattern for performance, in that performance is worst at high goal levels when value change is low, and best when values are high with low conformity and moderate goal levels. In the heat maps, it does appear that high values change the discontinuity for conformity slightly, provided goals are not extremely high, such that positive outcomes occur at slightly higher levels of conformity.

Discussion

These results do not have the strength to overcome the negative consequences of social pressure in the model that was expected. There are slight positive effects of having highly valued new KSAOs in overcoming the detrimental effects of conformity, but these are similarly weak as those seen for goals overall in the previous experiments. Further, the effects of goals in the model becomes clearer as agents again explore sub-optimally under many conditions, but a positive effect of conformity, if there is one, is that agents do not improperly explore undesirable policies if their social group does not allow them to do so. Along the same lines, the transfer that does occur when values are low tends to be maladaptive as agents make the mistake of continuing to apply their training when they should not, largely as a function of high goals and the freedom to do such exploration. An interesting implication here is for training which an organization knows will reduce performance, but may have other necessities, such as legal compliance. In such cases it is apparent that the organization will need to work to overcome substantial individual and group processes to make the new training successfully transfer back to the work environments. It is in such cases where physical tools, such as checklists or software, to assist with compliance seem likely to be of extra value.

Experiment 4D: Type 2 Likelihood, Conformity, and Goal Levels

A final exploratory simulation examined the effect of the ability for individuals to engage in type 2 cognitive processes on observed transfer outcomes across conformity and goal levels.

In this experiment, no direct predictions were made *a priori* as it is unclear what the effect of changing levels of type 2 likelihood might be in this complex simulation. One might think that allowing individuals to engage in deeper cognitive processing would better allow them to think about the benefits of their newly trained KSAOs, but it would also allow them to think more about the potential consequences of not conforming to their social group. This counteractive effect could wash out any gains from improving cognitive processing. At the same time, lower type 2 processing would lead to initial difficulties in transfer as trainees habitually apply their old KSAOs to the presented task, but would provide potential benefits in countering the effects of their social groups if they are able to establish their newly trained KSAO as their habitual response. These contradictory possibilities were explored in this experiment.

Methods

For a final time, the model coming from Study 3C was used to explore the joint effects of type 2 likelihood, conformity, and goal levels. For this experiment, conformity, goals, and type 2 likelihood were again swept from 0 to 1.0 in .05 increments. Other variables were held constant, with the true value of Policy A being .70, type 2 likelihood of .80, pre-training time points of 100, with 500 transfer time points, and initial policy value estimates of .50, 3 trainees per simulation. 500 replications were completed for each condition.

Results

Initial analyses suggest the effects of all three variables explored here have effects in the expected direction across replications on our outcomes of interest. Goal levels again had small but positive relationships with behavioral transfer ($r(4630500) = .068$, $p < .001$) and post training performance ($r(4630500) = .038$, $p < .001$), while type 2 likelihood had positive relationships with both behavioral transfer ($r(4630500) = .542$, $p < .001$) and post training performance

($r(4630500) = .302$, $p < .001$). Conformity again showed negative relationships with both behavioral transfer ($r(4630500) = -.593$, $p < .001$) and post training performance ($r(4630500) = -.330$, $p < .001$). Moderated multiple regression analyses were completed predicting behavioral transfer, post training performance, and pre-post training performance change from the three-way interaction of type 2 likelihood, conformity, and goals. The resulting parameters for these models can be found in Table 16, and graphs of the interactions in Figures 76-78. Heat maps were then generated at the condition level to explore these effects further and can be found in Figures 79-81. The moderation results initially suggest a typical moderation effect where we see the best transfer outcomes when conformity is low and type 2 likelihood is high, largely regardless of goal level, and all other combinations result in poor outcomes. Our heat maps generally confirm this effect with little else to add, with the exception that very high levels of type 2 likelihood are the only levels which substantially overcome the effects of conformity.

Discussion

It was unclear what to expect *a priori* for the present simulation, and it was found that potential beneficial effects of goals and type 2 likelihood were essentially wiped out at all levels of conformity with the only exception being the ability of high type 2 likelihood to lead to positive outcomes. Importantly, the effect of goals in overcoming the effects of conformity were almost non-existent once controlling for the effect of type 2 likelihood. Interestingly, type 2 likelihood appears to do a better job than any other intervention tested here in overcoming the negative effects of conformity, but type 2 likelihood must be high. This effect suggests that in designing training interventions attending to environmental characteristics will be of great concern in facilitating transfer. One must ensure that both the learner's social and physical environment allow them to engage in the kind of cognitive processes and exploration required to lead them to discover their training is beneficial to completing the relevant task.

Overall Discussion

The four experiments described here were meant to be demonstrations of the potential of the modeling platform developed throughout this paper to provide novel insights and guidance for future research and practice in organizational training and transfer. One of the primary strengths of computationally modeling theories such as the LTM lies in the ability to conduct such explorations in a low-cost and risk-free environment prior to committing the resources necessary to do similar explorations in empirical data collections. In these experiments, results suggested that the power of social learning as seen in the mechanism of conformity is a powerful depressing effect on transfer outcomes. Unfortunately, overcoming this effect is not necessarily easy, though goals and the ability to engage in type 2 cognitive processes show some promise. These results can be used to guide future data collections to continue testing the present model, and potentially for guidance in designing and supporting effective organizational training events.

Overall Discussion

Training represents one of the classic areas of inquiry and practice in Organizational Psychology, with over 100 years of research to show for it (Bell et al., 2017). In that time, we have developed a substantial body of knowledge which has allowed us to continuously improve the way we deliver training interventions in organizations and thereby improve training outcomes (Bell et al., 2017; Salas et al., 2012). Unfortunately, this base of knowledge focuses largely on the training event itself and generally treats the transfer of that training as a cross-sectional outcome (Foxon, 1997). This typical approach necessarily limits our knowledge because we are not generally studying transfer as a process that itself unfolds over time. The failure to study transfer as a process is unfortunate as we have acknowledged that it is indeed a longitudinal phenomenon for at least 30 years (Baldwin & Ford, 1988). However, in practice, few studies measure transfer longitudinally, and even fewer unpack the dynamic processes driving that transfer, with few notable exceptions (e.g., Dierdorff & Surface, 2008; Huang et al., 2015; Huang et al., 2017).

Recently, a group of researchers, including the present author, has begun more substantially to attempt to unpack the processes underlying training transfer. Most prominently, Blume et al. (2019) described training transfer as a self-regulatory-driven process, labeled the Dynamic Transfer Model (DTM), where trainees iteratively attempt to transfer their learning to their work environment and subsequently keep or discard their newly acquired KSAOs based upon the feedback they received. The primary drawbacks to their model lie in its narrative nature, and its failure to unpack the cognitive and learning mechanisms underlying the proposed feedback process. Surface and Olenick (forthcoming) are attempting to push the DTM to a lower level of abstraction and begin theorizing about how the transfer process may be driven by the interpretation of environmental cues and subsequent execution of available behavioral scripts, based largely in the same Dual Processing framework used in the present paper. However, their advancement still relies on narrative theorizing. Then, Olenick et al. (in press) began to push transfer research towards using more mathematical bases by applying non-linear dynamics to discuss training and transfer as a process of discontinuous shifts where old patterns of behaviors, represented by attractors in a mathematical sense, must be broken free from and new patterns formed. Their lens demonstrates how transfer trajectories can be modeled as dynamic processes that unfold over time as governed by mathematical attractors, which provides a more formal framework from which to build future research.

The Learning Transfer Model presented in this paper represents a culmination, of sorts, of these efforts. The LTM takes the step of fully formalizing the learning and decision mechanisms I propose underly the process of learning/training transfer in organizations. In doing so, the LTM integrates theories from across psychology, using Dual Process Cognition (e.g., Kahnemann, 2011) as a broad framework, self-regulation (e.g., Carver & Sheier, 1998), and Social Learning Theory (Bandura, 1977), with theories from outside of psychology, such as computational reinforcement learning (Sutton & Barto, 2018). Further, computational approaches to social learning were borrowed from studies of gene-culture coevolution (Richerson & Boyd, 2005) to discuss the effects of social learning on transfer from a lens of the simultaneous emergence of a transfer climate and that climate's effects on transfer outcomes. The final model, demonstrated via experiments in Study 4, broadly suggests that learners return to their work environment and must apply some new KSAO to their work instead of some existing KSAO they already were using. When encountering the applicable task, the learner initially decides quickly and automatically, via type 1 cognitive processes, which KSAO to apply based on previous experience. In some cases, the individual will have the opportunity to engage in deeper levels of cognitive processing and make a more conscious and informed decision regarding which available KSAO they should apply, these are governed by type 2 cognitive processes. Once an approach is chosen, the learner applies that choice to their task and receives feedback regarding the successfulness of their attempt. That feedback allows them to learn over time which of their available KSAOs can best allow them to perform the task to their desired level. If the new KSAO is perceived to be better, regardless of if it is actually better or not, than their previous KSAOs the learner will transfer that new KSAO over the long term. Complicating matters, individuals do not always actually attempt tasks because they lack the confidence in their ability to succeed so may decide not to even attempt to transfer their learning. Further, these learning and decision processes do not take place in a vacuum as learners are often embedded in work groups. The environment for transfer is then a simultaneous emergent phenomenon governed by the individual experiences of all the learners in their transfer attempts, and a causal factor where those learner's transfer decisions are constrained or enhanced by the emergent climate around them through either conforming or imitating mechanisms. As these decisions and learning events play out over time, an individual may follow any one of a nearly infinite set of transfer trajectories that, in the end, result in what we traditionally observe as successful transfer or not.

This overall theory was formalized and instantiated into a computational model in NetLogo, building from existing mathematical frameworks such as computational reinforcement learning. A series of simulations then explored the models and developed them in an iterative fashion. The goal for this iterative process was to explore each model and according to established modeling steps check the models for verification, generative sufficiency, robustness, and sensitivity (Railsback & Grimm, 2012). In addition, this process importantly opened the theory to an initial round of falsification (Popper, 1959). Overall, this process suggested the LTM, as originally proposed, was in many respects successful in its initial attempts to account for broad patterns of findings within the transfer literature, but not completely so. For example, the LTM was able to reproduce a range of behavioral transfer rates typically discussed in the literature (e.g., Ford et al., 2011), and general effect sizes for performance improvement we may expect in real world situations. However, it was also found that these findings were only true for some areas of the potential parameter space covered by the model, which were used in later simulations for further exploration. Such findings do not any more invalidate the present theory than do traditional tests of narrative theories in organizational psychology to establish boundary conditions (e.g. Grant, 2008; Hollenbeck, Colquitt, Ilgen, LePine, & Hedlund, 1998; Yammarino & Dubinsky, 1994). Instead, it appears that the LTM is a plausible process explanation for general transfer findings *provided the model is within certain parameters*. Outside of those parameters the model may not apply to the phenomena of interest for at least two reasons. First, it may be that the theory itself breaks down outside of the established parameter ranges which produce the kinds of relationships and results we are used to seeing in the research literature. If this is the case, the model would be falsified for those conditions and need to be further refined to operate under those conditions if it is deemed necessary, much as we would iterate a narrative theory. Second, as argued previously, it could be that it is not the theory that breaks down, but rather the limited range of conditions in which we tend to do our research. The model may be able to simulate conditions outside the bounds of reality, and therefore would not need to be applicable to them, and the breakdown in these ranges is therefore not a shortcoming.

However, one of the strengths of formal theorizing and computational modeling is the greater ability to falsify and iterate theories than achieved through traditional narrative theory building. This strength is clearly shown in Study 2 where the initially proposed pooling mechanism was incapable of replicating the expected social effects observed in the transfer literature. This model, being overly parsimonious and subsequently falsified via virtual experimentation, was able to be iterated by testing two alternate models of social learning borrowed from modeling of cultural effects on populations (Richerson & Boyd, 2005) which utilized mechanisms of imitation and conformity. Unlike the originally proposed mechanism in the LTM, both mechanisms appeared to provide some plausible results and novel insights into the nature of social effects in the transfer environment. Following some exploration, it was argued that with some reconsideration of how we operationalize culture and climate for transfer, the conformity model may fit current findings in the research literature better and was retained for further exploration.

Over the course of the iterative theorizing and model-building approach outlined throughout this paper, a final version of the LTM was accepted, for now, and more fully explored in Study 4. Through this process, it is argued that the present paper has accomplished its primary goals of 1) providing a formal, process-oriented theory of training transfer, 2) integrating multiple disparate theories to explain that process, 3) bringing outside theories, such as computational reinforcement learning and dual process cognition, more into the organizational psychology literature, and 4) building a modeling platform that would allow for the thorough exploration of the proposed theory for both theoretical and practical implications. It is to these implications we now turn.

Theoretical Implications and Future Research Directions

To quote Lewin's famous maxim, "there is nothing so practical as a good theory" (1943). In that spirit, the present paper sought to further our understanding of one of the most practically impactful research areas in all organizational psychology, training and transfer, by introducing a mechanistic process theory of transfer. To support the veracity of this theory, the Learning Transfer Model, a computational model was generated and explored to account for existing general findings in the research literature, a process referred to as establishing generative sufficiency. As discussed throughout this paper, these simulations suggest that the LTM can reproduce the general patterns of many research findings in this space. Therefore, it is argued that *the LTM, as currently specified, generally provides a plausible process-explanation for training transfer*. The ability of the present model to broadly account for many findings in the transfer literature is a critical first step in building a unifying theory for this area of our science and continue to improve our scientific rigor (Muthukrishna & Henrich, 2019).

The general success of the LTM displayed in this paper has a couple of interesting implications for how we think about training and transfer in our literature. First, Blume et al. (in press) recently suggested the need for more work on transfer as an individualized process where trajectories between individuals are likely to be highly idiosyncratic. Modeling the LTM reaffirms this case as it was evident that individual trajectories of agents can vary substantially. One further implication of the LTM in terms of that individualization process is the importance of viewing transfer from a perspective of need fulfillment. Throughout this paper we have seen agents are only likely to transfer their training when that training represents an improvement over their old behaviors, the training allows them to meet their personal goals, and they are allowed the ability to ascertain that benefit. Thus, if an individual is unable to discern how or whether their training meets their own needs then transfer is unlikely. Future work should continue to unpack this individualized nature of training transfer.

Further, the development of the LTM in this paper should encourage other researchers to look more closely at other fields as they begin to develop formal models of their own processes of interest. As a field, organizational psychology has not been on the forefront of the development of formal models and many other fields, from computer science, to biology, to economics, have been using mathematical tools to model their processes for decades. We could likely draw on their already existing models and associated mathematical approaches to inform much of our own work on the organizational processes in which we are interested. Being willing to use their work will keep us from reinventing the wheel when it comes to discovering many of the same essential processes. Similarly, through integrating models from across the sciences we can likely help place a break on continued construct and theoretical proliferation where many researchers from many different fields all study the same essential phenomenon but develop their own theories and constructs to explain and describe those phenomena. The historically siloed approach to science has likely slowed our knowledge accumulation and led to sprawling and confused literatures passing each other like ships in the night as each independently seek to solve similar problems. The ability of the LTM to provide a process capable of largely reproducing typical transfer findings in a relatively parsimonious model by integrating knowledge from across several disparate fields should provide further impetus for interdisciplinary work in the future.

However, it is not contended that the present paper has established the LTM as the *correct* model of training transfer, only that it is a plausible explanation, or at least a plausible step in establishing such a theory. Perfection was never the goal of the present theorizing, and the

LTM cannot be evaluated against such a standard. As Box (1976) contends, all theories are wrong, the goal is to remain parsimonious while providing an explanation for the phenomena at hand. The LTM, although integrating multiple disparate theories, only has a few actual mechanisms when expressed formally, making the overall model fairly parsimonious while still appearing to be broadly applicable to transfer research. The question then becomes not necessarily whether the theory is incorrect, but in which ways it is *meaningfully wrong* (Box, 1976).

As has been discussed through the results of the simulations above, there are at least a couple of ways in which the current version of the LTM is, or was, meaningfully wrong. For example, the effects of practice in the simulations was in the correct direction, but obviously not capable of reproducing the desired effects. This is problematic as practice effects are some of our best-established tools for improving learning outcomes (e.g., Dunloski et al., 2013). Additionally, even though in some cases the effects of policy value estimates in the LTM worked nearly perfectly as with the recreation of the effect of efficacy on transfer, those value estimates only reproduced the effect of utility reactions at the condition level and not the individual level as expected. On the extreme end, it was shown that the initially proposed social learning mechanism for the LTM was inadequate for producing the desired social effects. Already within this paper two alternative mechanisms were proposed and explored, with both showing greater potential for illuminating social effects in the transfer process.

Future iterations of the LTM, combined with targeted data collections, will be required to fine tune these mechanisms. In the case of value estimates in relation to utility reactions the underlying mathematics will need adjusting. The current effect of initial value estimates quickly becomes swamped by the experience of the learning agent, and therefore does not substantially affect the willingness of the agent to continue engaging with a task in the face of initial failure. If this effect can be drawn out over time by changing the updating procedure for value perceptions, the initial estimate may be able to better approximate the effect of utility reactions that it was thought those initial estimates would approximate. Similarly, the effect of practice within the LTM is not strong enough. It is not feasible, in most situations, for practice attempts to approximate the number of attempts an individual had using the behavior they are trying to replace. Therefore, the mathematical effect of practice attempts must be increased in some way. One way to accomplish this would be a multiplier on the practice attempt variable indicating the relative effectiveness of those practice attempts. Low numbers of this moderator variable, such as the de facto 1 it is set to in the present model, would represent poor practice. Higher numbers could represent better practice, such as following recommendations for spaced practice, recall effects, etc. that would improve the strength of those practice attempts. Future iterations of the model should explore these possibilities.

As for the social learning mechanisms, more modeling and data collection will be necessary to decide if the imitation, conformity, or a possible mix of both (e.g., Lopes et al., 2009) is needed to account for the social effects observed in transfer environments. Future empirical work should be partnered with versions of the social learning mechanisms tested in the LTM to ascertain which models better fit observed data regarding social interactions, learning, and how those lead to transfer or the lack there of over time. Targeted data collections and further modeling should then trade off in an iterative way to refine the models and determine which has the stronger support in the real world. Doing this would be a prime example of strong theoretical development (Sutton & Staw, 1995), which is one of the primary draws of engaging in computational modeling.

More generally, studies will be required to begin directly parameterizing the model against real data and go beyond the replication of general results. Several good examples of such approaches exist in the organizational sciences, ranging from the study of motivational phenomena (e.g., Vancouver, Weinhardt, & Vino, 2014), to the study of response processes to situational judgement tests (Grand, in press). However, it is unlikely that many opportunities exist to collect data within real organizations at the level of granularity required to fit the LTM to the kind of moment to moment decisions that are being proposed to drive transfer patterns. Such a collection would, almost of necessity, be highly intrusive and distracting to the point of overly interfering with normal organizational operations. For this reason, I reiterate the calls of other papers (e.g., Blume et al., 2019; Olenick et al., in press) to look for opportunities to use new technologies which can collect data on decisions and behaviors *in situ* in near real time. These include the ability to collect data on momentary use of electronic systems, or sociometric badges to study interaction patterns (e.g., Zhang, Olenick, Chang, Kozlowski, & Hung, 2018) which could provide windows into both individual and group behavioral norms.

Alternatively, experimental paradigms will need to be adapted to study the mechanisms outlined in this paper. Existing options include: a) scheduling tasks which track decisions made over many time points to study motivational processes (e.g., DeShon & Rench, 2009; Schmidt & DeShon, 2007), and b) a radar simulation task called TANDEM which can track participant decisions down to individual clicks of a mouse and time spent on various tasks, in a difficult environment where much learning is possible (e.g., Bell & Kozlowski, 2008). A major drawback of such platforms, however, is that the odds of success on the task attached to any specific behavior are unknown and might not be possible to be known without extensive simulation or prior data collection. This poses a problem in testing the LTM as it relies on the underlying probability of success associated with each behavioral option. Possibilities to overcome this limitation lie in using games which are well studied by mathematicians regarding the various probabilities associated with different tactics. It would also be ideal if such a game allowed for many repetitions of the same essential task with an obvious success or failure in short amounts of time. Natural fits here include poker and blackjack, which have well established guidelines for play, happen quickly, and participants can typically be taught different strategies with little difficulty. Any of these could also potentially be adapted to study the social pressures surrounding transfer of any training interventions programed into the study environment using electronic confederates (Leavitt, Qiu, & Shapiro, 2019), or other real participants.

Regardless of the research platform utilized, the present model has apparent implications for how we analyze training data. Many of the simulations discussed in this paper suggest the effects of various parameters do not always demonstrate the types of smooth, linear effects we typically study in the organizational literature, or at least that we typically capture using our ordinary least squares regression analytics. Instead, variables often display fairly sudden and rapid changes in their effects as levels in the variable of interest change. One example of such an effect was the change in behavioral transfer across levels of the threshold variable in Study 3C where behavioral transfer rates rapidly decreased from a threshold level of .60 to about .70. Such a pattern is not a complete discontinuity, but it suggests a pattern that may be better analyzed through nonlinear methods. For example, a cusp catastrophe model could assess the likelihood of a target falling on either level of the observed rate of behavioral transfer while treating threshold level as a control variable for the location of that discontinuity. Such models have long been used in studies of animal and human learning (e.g., Baker & Frey, 1980; Guastello, 1987), and have recently been suggested for greater use in the study of organizational training and transfer (Olenick et al., in press). The simulated results of the LTM in this paper reaffirm this suggestion.

Future iterations of the LTM should also seek to include other emerging research on human learning and decision making and its potential effects on transfer outcomes. For example, Spicer, Mitchell, Wills, and Jones (2020) suggest that humans protect their established causal beliefs instead of updating them when their predictions do not match observed outcomes, violating existing prediction error models. Their findings could be matched with the LTM to discuss why in transfer space learners/agents do not necessarily accurately update their beliefs regarding the value of their behavioral policies in the face of experience. For example, one of the biases operating in type 2 processing systems could be a discounting of the effects of failures for learning about the utility of Policy A. When the learner enters the transfer environment then, not only does their new policy have to outperform Policy A outright to convince the learner it is better for the task, but also overcome any bias of the learner ignoring failures of Policy A in a protection of their prior beliefs. This is an intriguing idea that at least anecdotally fits with experience in real organizational environments and seems to be worth further exploration.

Another interesting possibility would be to combine with other computational models that explore pertinent aspects of the transfer process that are not yet included in the present model. For example, the LTM currently assumes that trainees can accurately perceive their environment in order to activate the relevant decision processes discussed here. This assumption can be relaxed by incorporating mechanisms in other models, such as Weichart, Turner, and Segerberg's (2020) recent model which examines decision making to understand how decisions emerge within a task trial as a function of a subject's attention to aspects of their environment. The incorporation of similar mechanisms into the LTM would allow us to model how learners might interact with their environmental cues to activate the relevant behavioral scripts represented by the policies used in the terminology of reinforcement learning. One interesting interaction would likely occur with the ability to identify the relevant environmental cues to fully realize the benefits of implementation intentions. As discussed previously, implementation intentions are described as if-then type rules where the learner applies the relevant response in the presence of the correct cue (Gollwitzer, 1999). For this mechanism to operate, the individual must be able to recognize the cue and doing so requires paying sufficient attention to the relevant environmental factors. Therefore, there is likely a moderating effect of attention on the effects of implementation intentions within transfer environments.

Another frontier for the LTM will be to account for more and evolving behavioral options. Many tasks have specific ways they are supposed to be carried out, to which the current version of the LTM is most applicable. However, many tasks are more open, allowing trainees greater discretion over how exactly they approach the task (e.g., Yelon & Ford, 1999). To incorporate many different behavioral options, the LTM should be expanded to utilize reinforcement principals for multiple behaviors. The k-armed bandit approach used here is technically capable of assessing multiple policies at a time, but more sophisticated models exist (Sutton & Barto, 2018). Other reinforcement algorithms are likely better fits for different types of transfer questions, and they should be systematically explored for that fit. Similarly, it may be possible that different types of learning, reinforcement or otherwise, are better fits for the learning mechanisms occurring within either type 1 processes or type 2 processes during transfer events. The present approach was chosen as a starting point as historical research on animal learning and applied reinforcement learning models largely focuses on naïve learners (see Sutton & Barto, 2018 for a discussion), while the specific question being addressed in the present paper is that of adults who's learning processes are affected by prior experience (Knowles, 1984). However, as suggested in the CLARION model (Sun et al., 2005), the type of experiential learning that lies at the heart of the reinforcement algorithms used in this paper (Sutton & Barto, 2018) are proposed to fit with type 1 processes but not necessarily with type 2 learning processes, although we are interested in more than the explicit informing of an individual regarding the usefulness of new KSAOs in the present case, thus tackling a different question than CLARION. Further research and modeling to refine these mechanisms to best fit the transfer environment will be required.

In addition, the present paper has only focused on a single learning and transfer event, where a single old behavior must be overcome for transfer to occur. However, the development of individuals within organizations, and more broadly expertise, can be viewed as the constant breaking of these old habits and establishment of new ones (Ericsson, 2006; Olenick et al., in press). In traditional reinforcement learning problems, such as an agent discovering the most efficient way to navigate a maze, the agent generates solutions to its environment and learns their values over time (Sutton & Barto, 2018). In the same way, general employee development could be viewed as a series of pseudo-randomly generated solutions to organizational problems where the learner then chooses which to apply to their particular work situation or not, over time developing preferences for some behavioral policies over others and requiring new policies to overcome that preference in order for transfer to occur. Through such an approach we could go beyond the study of the transfer of a single learning event to better understand sequential learning events. Simultaneously, such models can account for changing environments (Sutton & Barto, 2018) which would open the LTM to application further to questions of far transfer (Beier & Kanfer, 2010), and problems of adaptability (e.g., Baard et al., 2014).

A final key area for exploration, both within the present version of the LTM and across future versions will be to explore the many other potential combinations of interventions and effects that were not in this paper. For example, once practice effects are refined, how might they interact with implementation intentions? Much as we expected, initially, that improving engagement in type 2 processes would augment effects of implementation intentions but the model suggests that is incorrect, it would seem logical that both practice and implementations would be beneficial and augment each other. However, maybe once one effect is accounted for the other provides no gain in transfer outcomes, and therefore it would not be worth the effort and cost to utilize both in a training intervention. The LTM could provide such guidance for future investigations into these interactive effects and therefore guidance for the efficient practical application of research findings. It is to those more practical implications we now turn.

Practical Implications

Many practical implications for the individual models explored in this paper have been discussed throughout. However, there are a few overarching implications which warrant discussion. First, not only does the LTM and computational results have implications for how we measure transfer for research, it has implications for how we measure transfer for training evaluation. In this paper, the outcomes tracked were at the behavioral and performance levels of the classic Kirkpatrick (1994) typology. To merely encourage organizations to evaluate training outcomes at these levels would be banal, although they should do so more frequently than is currently the standard. What the modeling in this paper suggests further is that the timing of the measurement for these outcomes is of great importance. It is commonly stated in the research literature that the timing of measurements should be chosen based on the timing of the phenomenon of interest (e.g., Hanges & Wang, 2012), and this clearly pertains to the estimation of transfer outcomes in the LTM. Specifically, if transfer measurements are taken too early the outcomes of interest may not have had a chance to emerge and stabilize which could lead to a drastic over or underestimate of the final effect of a training event. To make matters worse, the models here suggest that transfer may more likely emerge later than one might expect, causing an underestimate of the effect of training, and therefore potentially leading an organization to incorrectly conclude their training was ineffective. Therefore, patience is urged in the timing of the collection of transfer data when possible to improve the final estimates of the effect of training.

In fact, the timing of every aspect of training appears to be of incredible performance. In Olenick et al. (in press) it was, in part, suggested that "the longer one waits to intervene, the harder it likely is to create lasting change" (pagination not yet assigned) due to the formation over time of an attractor due to the recurrent success of the targeted behavior. Their piece only applied a mathematical lens to training to make such a suggestion, and the present paper further demonstrates their point via modeling. In the initial exploration of the LTM in Study 1, we saw a drastic effect on training outcomes according to how long the pre and post training time frames ran. What is occurring in the simulation is the essential formation of the kinds of attractors Olenick et al. (in press) were discussing as the agent gained experience with their task. The present simulations showed that even 100 attempts, on this particular "task" at least, were sufficient to create a strong enough attractor that agents struggled to form new patterns unless given five times as many attempts to change that behavior. Such difficulties only become greater the longer the pre-training period is allowed to extend, as we see in the difficulty of overcoming implicit biases through training when those biases are the result of years or decades of experience (Lai, Hoffman, & Nosek, 2013; Lai et al., 2016). Although the exact number of trials likely does not map cleanly onto any given real-world task, the overall message is clear for the timing of training interventions: the sooner, the better. The advice for any practitioner choosing when to hold a key training event, at least regarding a task the trainees are already completing in some way, is to implement the intervention as soon as feasible as any delay is likely to make the task of causing permanent on-the-job change even more difficult.

Olenick et al. (in press) also suggest that the strength of the intervention will be critical in overcoming established KSAOs, especially when they are long-held patterns. One way to increase the strength of a single training event should theoretically lie in stacking multiple kinds of best practices or training enhancers into a learning event when possible. For example, maybe as a training designer you incorporate both spaced practice and implementation intentions, and from the present modeling you want to target the transfer environment to improve their use of type 2 cognitive processes. Independently, all these additions should improve learning and transfer outcomes, so it seems logical that doing all of them would be even more beneficial. However, the modeling in this paper suggests that may not always be the case. Instead, some types of interventions may not be able to effectively stack with each other to further improve outcomes and might even interfere with one another. In such a case, adding extra apparent enhancements to a training event could result in decreased return on investment for the event as energy is wasted in implementing unhelpful tools. Therefore, training designers should think carefully about which such tools will best fit with their planned training event to enhance desired outcomes.

Finally, the LTM suggests there may be other individual differences and environmental factors to consider when choosing who might be a good candidate for a given training event. During the typical person analysis phase of a training needs assessment, an employee’s readiness for training is assessed, which includes personal characteristics such as ability, attitudes, personality, and motivation, and if their work environment will facilitate the desired outcomes (Langdon, 1997; Noe, 2017; Rummler, 1996). Some of these characteristics are directly informed by the LTM. For example, we saw an interesting interplay between goals and the outcomes of training which suggests that individuals with extremely high goals might not be good fits for trainings which do not allow them to reach said goals. Rather, the focus should be on individuals whose current goals match well with what the training is offering. Further, we know individuals who are learning, or mastery in other nomenclature, oriented are focused on increasing their ability on their targeted tasks and this leads to improved performance outcomes over time (e.g., Dweck, 1986; Elliott, 1999; Payne, Youncourt, & Beaubein, 200), and part of doing so tends to be a greater willingness to explore the task for better solutions, leading to poorer performance early in those tasks but greater success over time (e.g., Bell & Kozlowski, 2008). In a similar vein, the present model shows that moderate levels of exploration in response to falling short of one's goals are beneficial for enhancing transfer outcomes. Thus, the model reinforces the potential importance of targeting individuals who are learning oriented for training interventions, or even adding a new measure directed specifically at their willingness to search for better task approaches in the face of adversity. Finally, on the environmental side, we want to ensure that not only do trainees have the theoretical opportunities to apply their training in the sense that the correct situations present themselves, we want to ensure those trainees have the time and ability to think more deeply about the situation and engage their type 2 cognitive processes to improve the chances that they will make the correct decision regarding whether or not to use their training.

Conclusion

The Learning Transfer Model introduced in this paper has four central aims. First and foremost, it provides a formal process-oriented theory which has the potential to unify many current effects in the transfer literature under a single umbrella. Second, it further integrates multiple important theories across disciplines both from within and outside of psychology. Additionally, the LTM brings important formal models of reinforcement learning, and dual process models of cognition further into organizational psychology. Finally, the LTM was instantiated into a computational model to provide a powerful tool for future theoretical development and practical application. The present work is not meant to be the final word on any of the theories incorporated into the LTM, or even be the final word on the mechanisms driving transfer in organizational contexts over time. Instead, the LTM as presented here is meant to provide a plausible and parsimonious model of the transfer process to drive future research and practice. To that end, over the course of several virtual experiments, the overall generative sufficiency of the model was largely established, although pieces of the model were falsified and subsequently revised, and novel implications of the model were explored. Substantial work remains to fully validate the present model against real world observations, which will inevitably lead to various tweaks to the underlying mathematics driving the proposed mechanisms in the LTM. However, the model established in this paper represents a substantial step in a formal process model of the transfer process.

Table 1. *Model 1 Variables*

Variable	Definition
a	Policy A
b	Policy B
R_a	True reward for Policy A
R_b	True reward for Policy B
$Q_t(a)$	Value estimate for Policy A at time t
$Q_t(b)$	Value estimate for Policy B at time t
R_{ta}	Reward received at time t given Policy A
R_{tb}	Reward received at time t given Policy B
$Q_1(a)$	Initial value estimate for Policy A
$Q_1(b)$	Initial value estimate for Policy B
P_t	Policy chosen at time t
E	Error rate in choosing most valuable policy, also referred to as exploration
S_2	Probability of activating System 2 decision process
$Z_t(a)$	Probability of choosing to apply Policy A automatically in system 1
L	Number of times an agent has attempted their new policy in practice before entering the transfer environment
I	Effect of forming an implementation intention to activate Policy B in the presented situation

Table 2. *Model 1 Equations.*

Equation	Definition
$Q_{t+1}(a) = Q_t(a) + \frac{1}{t_a}[R_{t\|a} - Q_t(a)]$	Value estimate at time t + 1 for Policy A where t_a is the number of times Policy A has been applied
$Q_{t+1}(b) = Q_t(b) + \frac{1}{t_b}[R_{t\|b} - Q_t(b)]$	Value estimate at time t + 1 for Policy B where t_b is the number of times Policy B has been applied
$P_t\|T_2 = \max[Q_t(a), Q_t(b)]$ $with\ probability\ 1 - E$	Policy chosen at t is the maximum expected value from policies a and b with a probability of 1 – E given the use of System 2
$Z_t(a)\|T_1 = \frac{\sum P_t(a)}{\sum P_t(a) + \sum P_t(b) + L} - I$	Probability of choosing to apply Policy A automatically in system 1 as calculated from the number of times that policy has been chosen out of possible applications and accounting for implementation intentions

Table 3. *Overall results for practice effect on behavioral transfer and performance change in Model 1*.

Number Practice attempts	Behavioral Transfer	Performance Change
0	.47	.43
25	.49	.48
50	.48	.31
75	.50	.33
100	.51	.38
125	.54	.54
150	.57	.54
175	.55	.63
200	.55	.54

Table 4. *Experimental comparisons of practice conditions to control for behavioral transfer and performance change in Model 1.*

Number Practice attempts	Behavioral Transfer	Performance Change
25	.25	.01
50	.08	.04
75	.51	.02
100	.56	.05
125	.97	.12
150	1.50	.14
175	1.15	.14
200	1.25	.14

Table 5. *Initial policy value estimate effects on behavioral transfer and performance change in Model 1.*

Initial Policy B Estimate	Behavioral Transfer	Pre-Post Performance (d)
.00	.43	.17
.05	.42	.21
.10	.44	.23
.15	.43	.19
.20	.44	.28
.25	.45	.43
.30	.44	.38
.35	.43	.33
.40	.42	.30
.45	.45	.27
.50	.46	.35
.55	.44	.40
.60	.45	.28
.65	.42	.27
.70	.43	.31
.75	.45	.36
.80	.42	.30
.85	.45	.36
.90	.45	.31
.95	.45	.39
1.00	.47	.41

Table 6. *Implementation level effects on behavioral transfer and performance change in Model 1.*

Implementation Level	Behavioral Transfer	Pre-Post Performance
0	.43	.31
.05	.45	.17
.10	.47	.34
.15	.50	.41
.20	.51	.39
.25	.54	.52
.30	.56	.33
.35	.55	.40
.40	.57	.32
.45	.59	.50
.50	.60	.47

Table 7. *Model 2 Variables.*

Variable	**Definition**
$G_t(a)$	Mean of other agents' estimates of value of Policy A at time t
$G_t(b)$	Mean of other agents' estimates of value of Policy B at time t
C	Level of connected to group of co-learners
$wQ_t(a)$	Weighted value estimate for Policy A
$wQ_t(b)$	Weighted value estimate for Policy B

Table 8. *Model 2 Equations.*

Equation	Definition
$G_t(a) = \frac{\sum_1^N Q_t(a)_i}{N}$	Calculation of the average value estimates of other transfer agents 1-N as the sum of all the value estimates for each agent, *i*, divided by the number of agents for Policy A.
$G_{tj}(b) = \frac{\sum_1^N Q_t(b)_i}{N}$	Calculation of the average value estimates of other transfer agents 1-N as the sum of all the value estimates for each agent, *i*, divided by the number of agents for Policy B.
$wQ_t(a) = (1 - C)Q_t(a) + CG_t(a)$	Weighted value estimation for Policy A when N > 0
$wQ_t(b) = (1 - C)Q_t(b) + CG_t(b)$	Weighted value estimation for Policy B when N > 0

Table 9. *Effects of number of trainees on behavioral transfer and pre-post performance change in Model 2A.*

Trainees	Behavioral Transfer	Pre-Post Cohen's d
1	.43	.28
2	.44	.52
3	.42	.54
4	.44	.68
5	.45	.78
6	.44	.91
7	.44	.86
8	.43	.84
9	.44	1.03
10	.43	.99
11	.44	1.09
12	.44	1.00
13	.44	1.19
14	.44	1.21
15	.43	1.21
16	.44	1.29
17	.44	1.24
18	.44	1.40
19	.44	1.42
20	.44	1.43

Table 10. *Connectedness effects on behavioral transfer and pre-post performance change in Model 2A.*

Connectedness	Behavioral Transfer	Pre-Post Cohen's d
.00	.44	1.12
.05	.44	1.12
.10	.43	.95
.15	.44	.84
.20	.44	1.06
.25	.44	1.06
.30	.43	.84
.35	.43	1.01
.40	.43	.91
.45	.44	1.05
.50	.43	.92
.55	.44	1.03
.60	.43	.95
.65	.44	1.10
.70	.44	.94
.75	.43	1.01
.80	.44	1.08
.85	.44	1.09
.90	.44	1.02
.95	.44	.96
1.00	.44	.88

Table 11. *Model 3 Variables.*

Variable	Definition
T	Goal of target agent
Y	Performance of target agent
D	Difference between performance and goal
J	Decision mechanism, takes 0 if goal is met, 1 if not
F	How much exploration increases when performance is short of the agent's goal
V	Threshold below which agent will not apply policy

Table 12. *Model 3 Equations.*

Equation	Definition
$Y_{t+1} = \frac{\sum_1^t R_i}{t}$	Performance Y is the average of all previously experienced rewards
$D_t = T - Y_t$	Difference calculated as the difference between the agent's goal and current performance
$E_{t+1} = E_0 + FJ_t$	Error rate in choosing highest valued Policy As changed by comparison of current performance to goal

Table 13. *Three-way interaction models for Experiment 4A.*

	Behavioral Transfer				Post Training Performance				Pre-Post Performance Cohen's *d*			
Predictor	b	β	t	p	b	β	t	p	b	β	t	p
Constant	.280		554.85	< .001	.761		6282.61	< .001	1.157		16.97	< .001
Intentions	.060	.026	20.28	< .001	.012	.020	16.92	< .001	.233	.016	.58	.560
Threshold	-.706	-.283	-221.41	< .001	-.107	-.168	-139.11	< .001	-1.973	-.125	-4.58	< .001
Value Change	1.983	.649	507.87	< .001	.559	.721	595.95	< .001	14.149	.733	26.80	< .001
Intentions*Threshold	-.221	-.015	-11.84	< .001	-.029	-.008	-6.43	< .001	-.409	-.004	-.16	.871
Intentions*Value Change	.361	.020	15.79	< .001	.109	.024	19.87	< .001	2.597	.023	.84	.401
Threshold*Value Change	-2.139	-.111	-86.65	< .001	-.603	-.123	-101.61	< .001	-11.008	-.090	-3.30	.001
Intentions*Threshold*Value Change	-.275	-.002	-1.90	.057	-.128	-.004	-3.69	< .001	-1.576	-.002	-.08	.936

Dfs for all models are 8, 296991

Table 14. *Three-way interaction models for Experiment 4B.*

	Behavioral Transfer				Post Training Performance			
Predictor	b	β	t	p	b	β	t	p
Constant	.712		4602.11	< .001	.247		141606.60	< .001
Trainees	.000	-.145	-699.16	< .001	-.007	-.199	-373.13	< .001
Conformity	-.024	-.556	-2690.95	< .001	-.476	-.776	-1434.90	< .001
Goals	.003	.068	325.94	< .001	.058	.093	174.56	< .001
Trainees*Conformity	.000	-.055	-267.85	< .001	-.008	-.076	-141.43	< .001
Trainees*Goals	.000	-.001	-153.55	< .001	.000	-.001	-81.51	< .001
Conformity*Goals	-.004	-.032	-3.19	.001	-.090	-.044	-2.39	.017
Trainees*Conformity*Goals	.000	-.007	-33.35	< .001	-.003	-.009	-17.26	< .001

Dfs for all models are 8, 4409991

Table 15. *Three-way interaction models for Experiment 4C.*

	Behavioral Transfer				Post Training Performance				Pre-Post Performance Cohen's *d*			
Predictor	b	β	t	p	b	β	t	p	b	β	t	p
Constant	.223		2739.29	< .001	.726		33146.55	< .001	.736		44.91	< .001
Conformity	-.473	-.670	-1754.87	< .001	-.056	-.369	-779.08	< .001	-2.690	-.381	-49.69	< .001
Goals	.031	.044	115.48	< .001	-.002	-.016	-34.19	< .001	-.070	-.010	-1.29	.196
Value Change	.964	.553	1448.33	< .001	.269	.711	1502.67	< .001	12.945	.742	96.75	< .001
Conformity*Goals	-.041	-.018	-46.54	< .001	.007	.014	29.73	< .001	.260	.011	1.46	.146
Conformity*Value Change	-2.181	-.379	-991.81	< .001	-.572	-.459	-969.18	< .001	-27.448	-.476	-62.11	< .001
Goals*Value Change	-.272	-.047	-123.84	< .001	.008	.006	13.10	< .001	.940	.016	2.13	.034
Conformity* Goals*Value Change	.607	.032	83.59	< .001	.010	.002	5.27	< .001	-.673	-.004	-.46	.645

Dfs for all models are 8, 661491

Table 16. *Three-way interaction models for Experiment 4D.*

	Behavioral Transfer				Post Training Performance				Pre-Post Performance Cohen's *d*			
Predictor	b	β	t	p	b	β	t	p	b	β	t	p
Constant	.145		3982.65	< .001	.707		120782.07	< .001	-.137		-176.95	< .001
Type 2	.288	.542	2386.45	< .001	.014	.302	743.90	< .001	.699	.608	272.84	< .001
Conformity	-.314	-.593	-2608.91	< .001	-.016	-.330	-811.81	< .001	-.763	-.663	-297.75	< .001
Goals	.036	.068	297.86	< .001	.002	.038	92.34	< .001	.088	.077	34.48	< .001
Type2*Conformity	-.580	-.331	-1456.80	< .001	-.029	-.185	-454.89	< .001	-1.401	-.369	-165.6	< .001
Type 2*Goals	.053	.030	133.91	< .001	.003	.017	41.42	< .001	.143	.038	16.88	< .001
Conformity*Goals	-.061	-.035	-152.40	< .001	-.003	-.020	-48.21	< .001	-.141	-.037	-16.62	< .001
Type 2*Conformity*Goals	-.070	-.012	-53.06	< .001	-.004	-.007	-17.09	< .001	-.184	-.015	-6.59	< .001

Dfs for all models are 8, 4630491

Figure 1. Conceptual model for initial LTM.

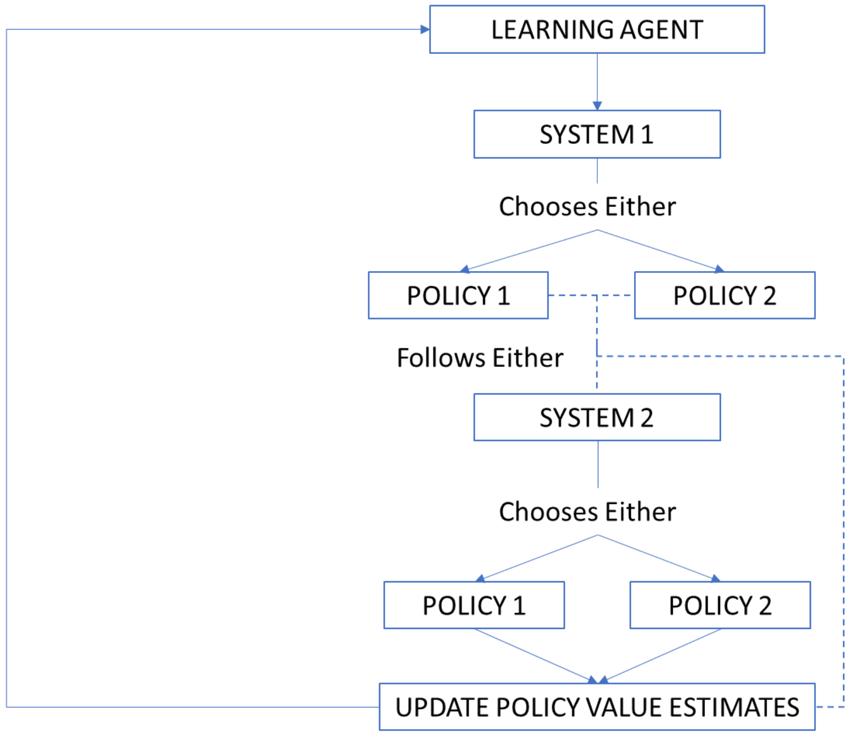


Figure 2. Behavioral Transfer for exploration of policy values in Model 1.

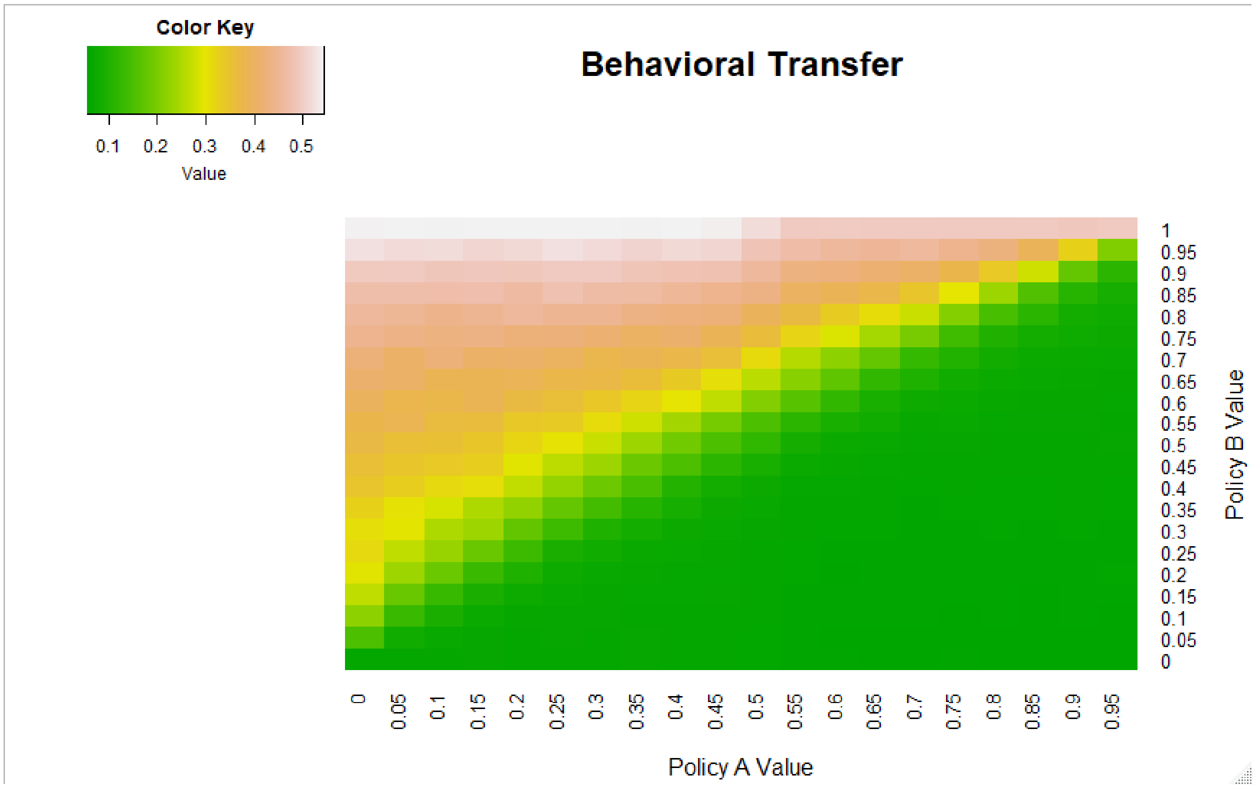


Figure 3. Performance change (in Cohen's d) for exploration of policy values in Model 1.

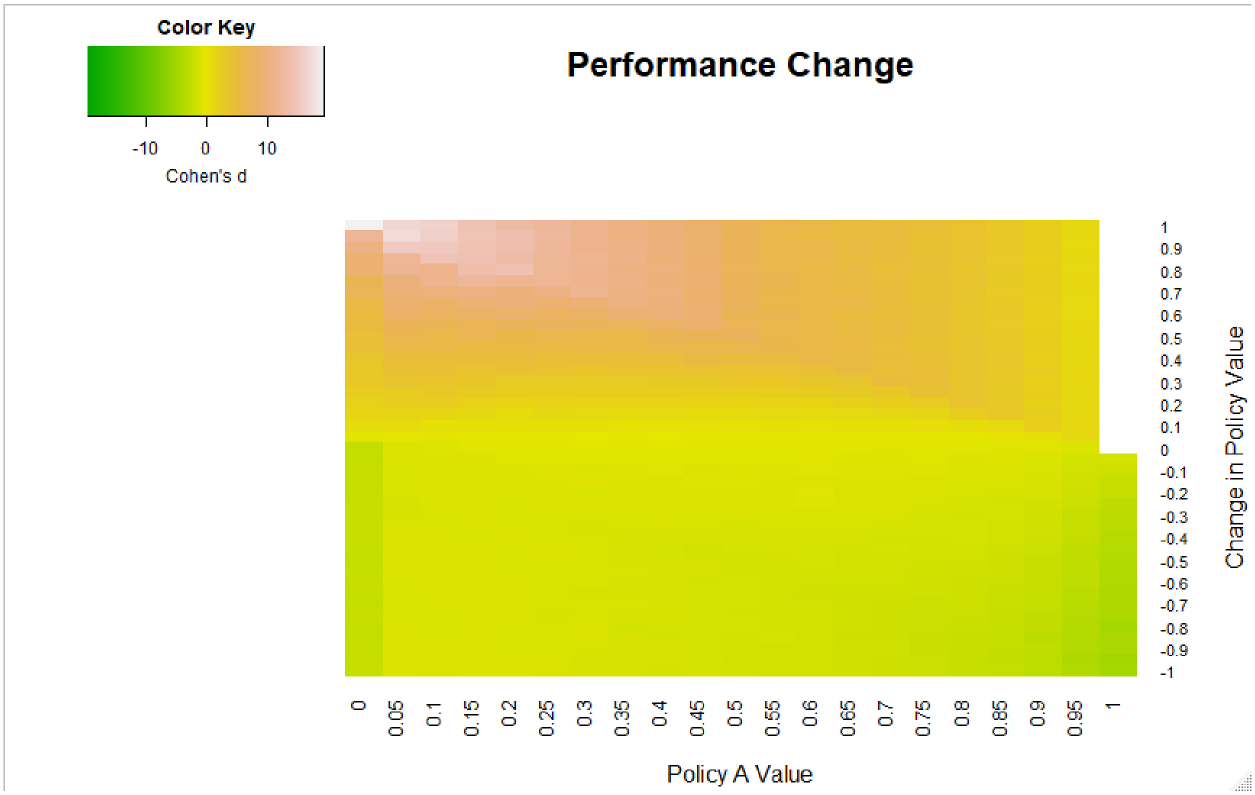


Figure 4. Behavioral Transfer for exploration of policy value changes in Model 1.

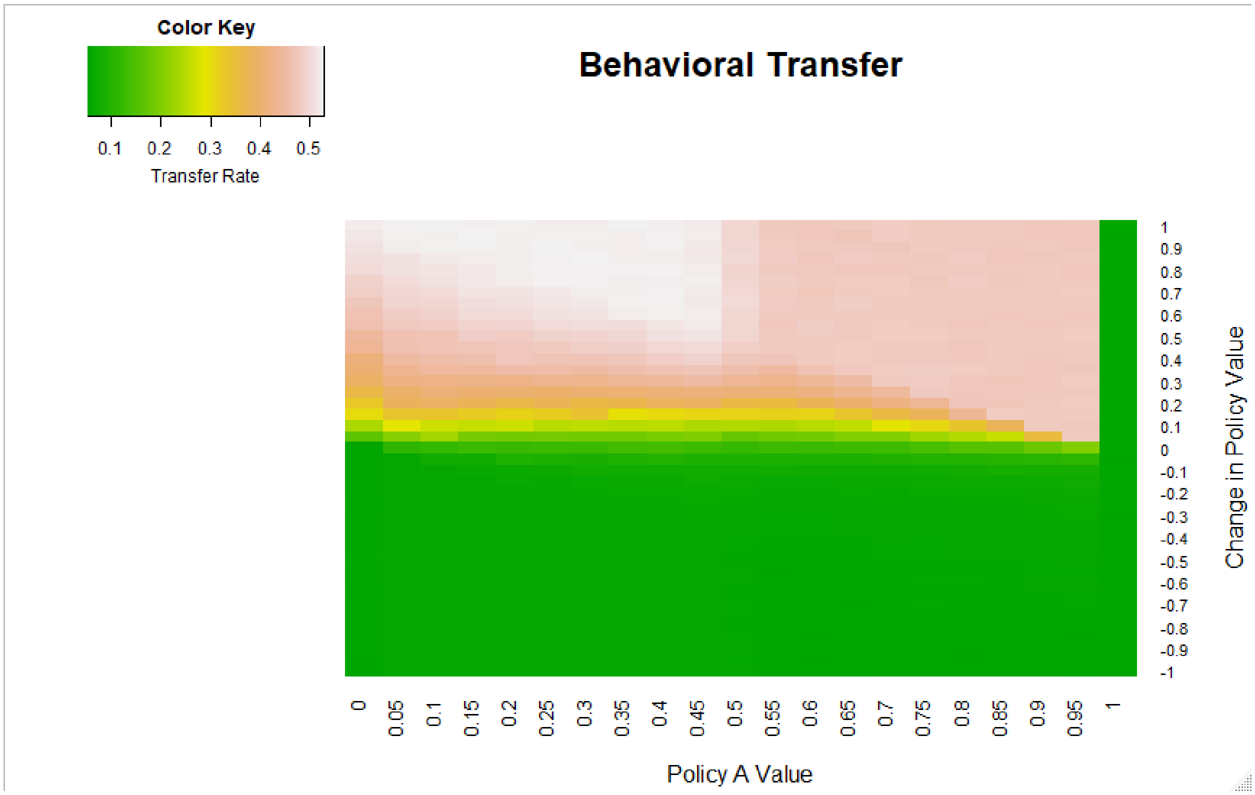
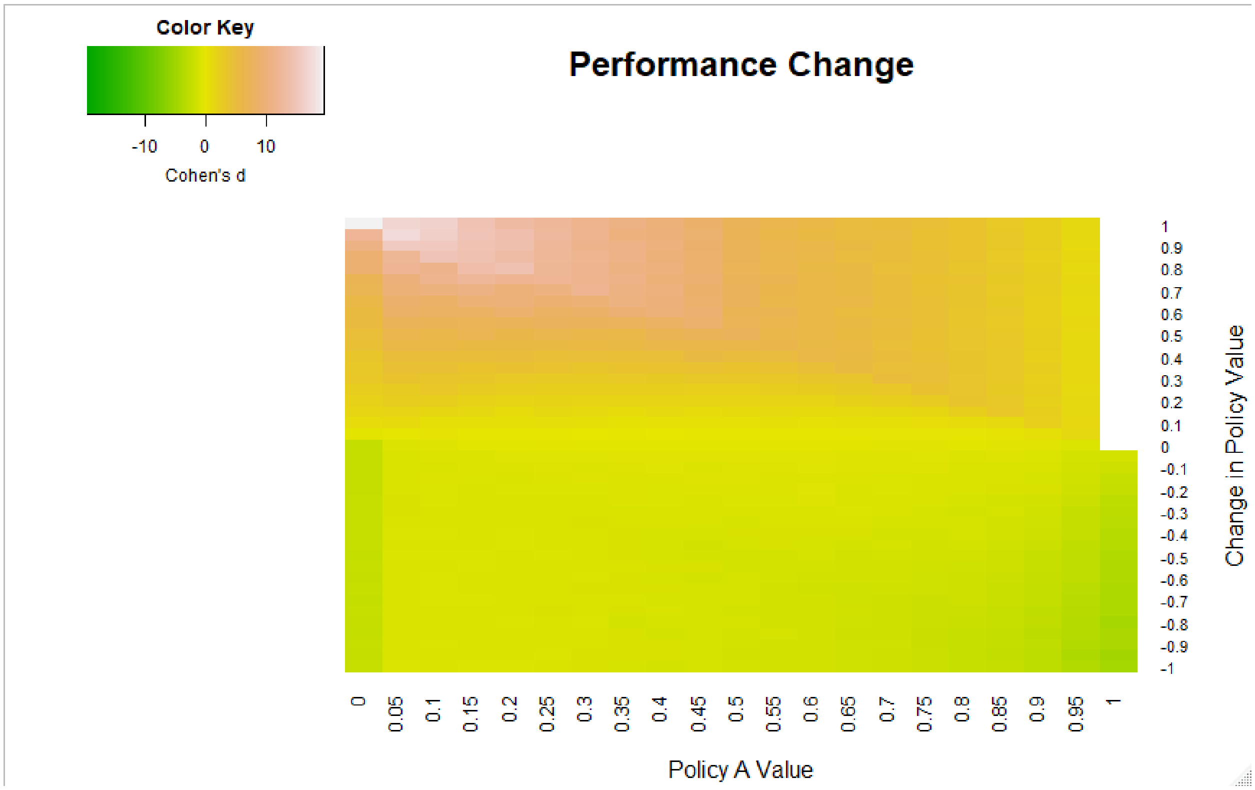


Figure 5. Performance change (in Cohen's d) for exploration of policy value change in Model 1.



Note: the white rectangle on the right is because the value is undefined since their performance pretraining was always perfect there is no variability on which to calculate an effect size.

Figure 6. Behavioral Transfer for exploration of burn-in and transfer times in Model 1.

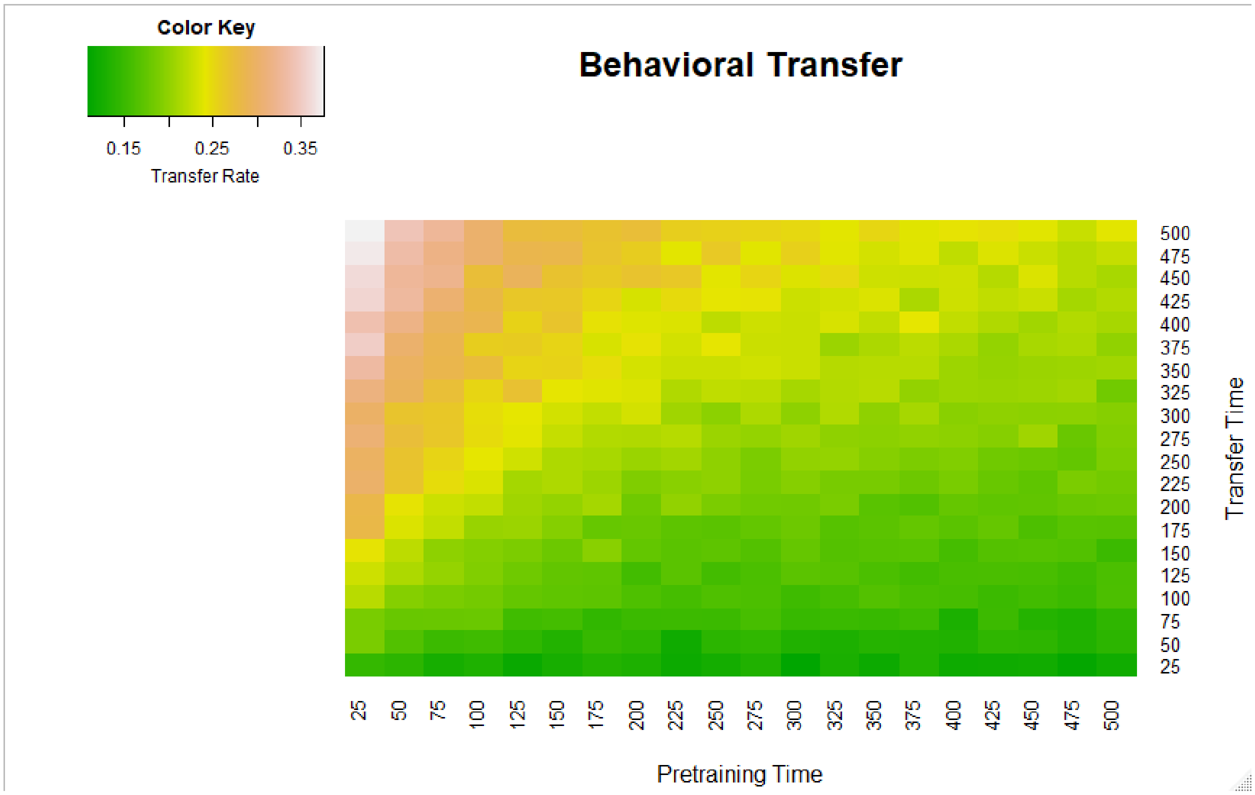


Figure 7. Performance change for exploration of burn-in and transfer times in Model 1.

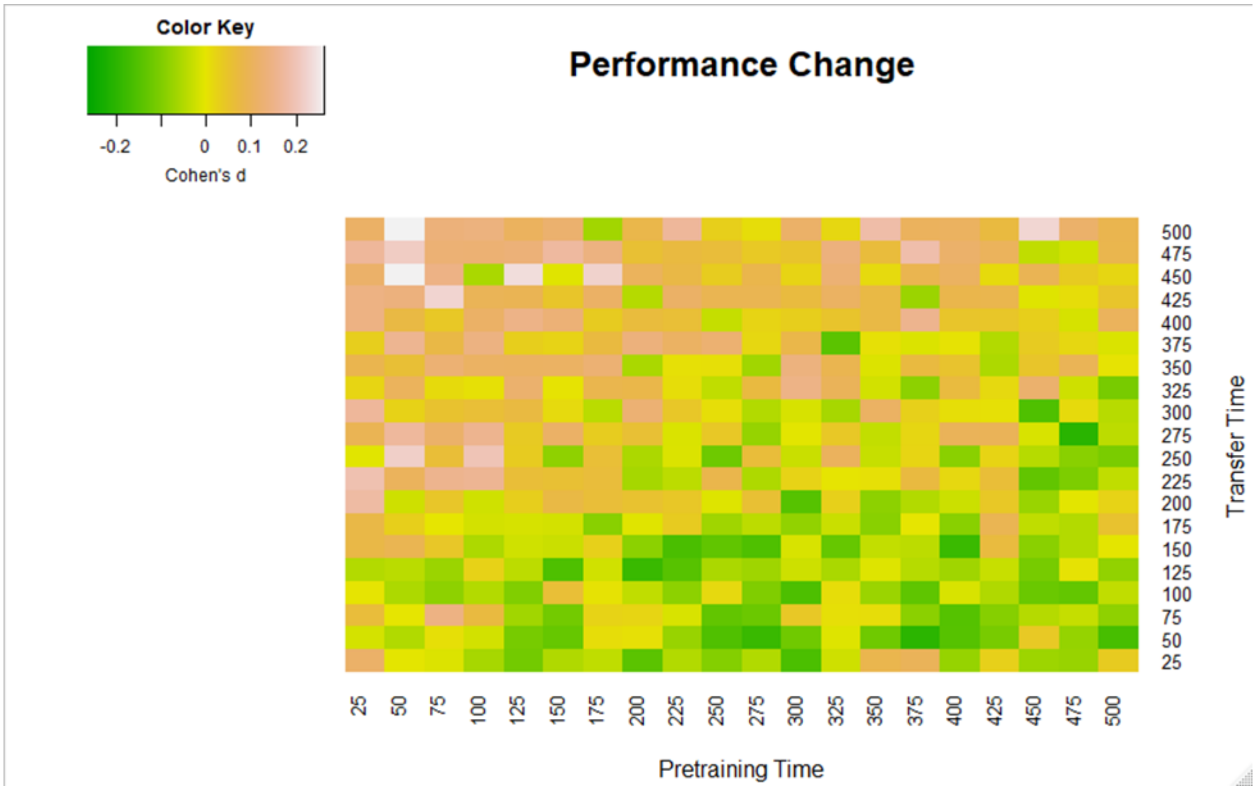


Figure 8. Predicting behavioral transfer from type 2 processing likelihood in Model 1.

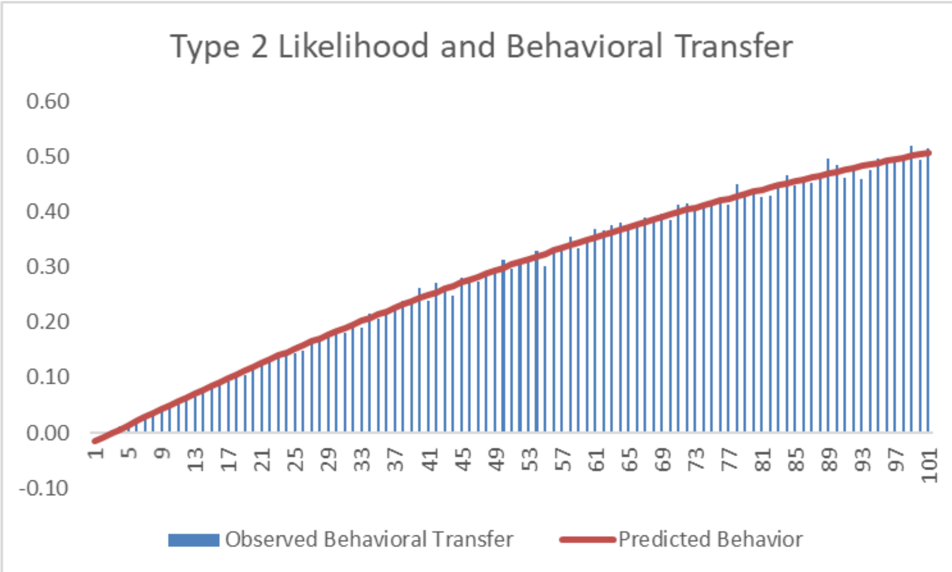


Figure 9. Predicting performance change from type 2 processing likelihood in Model 1.

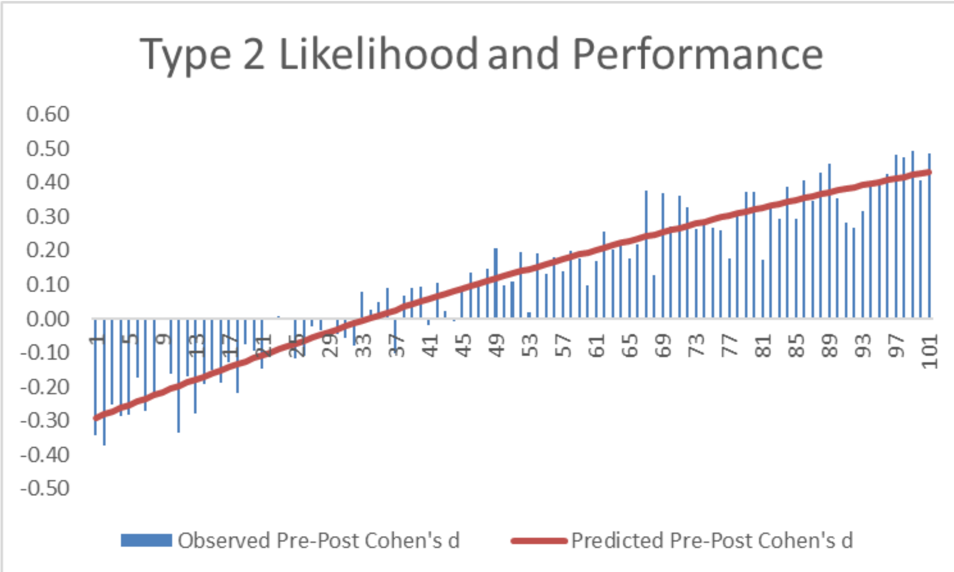


Figure 10A-D. Example transfer trajectories for Model 1.

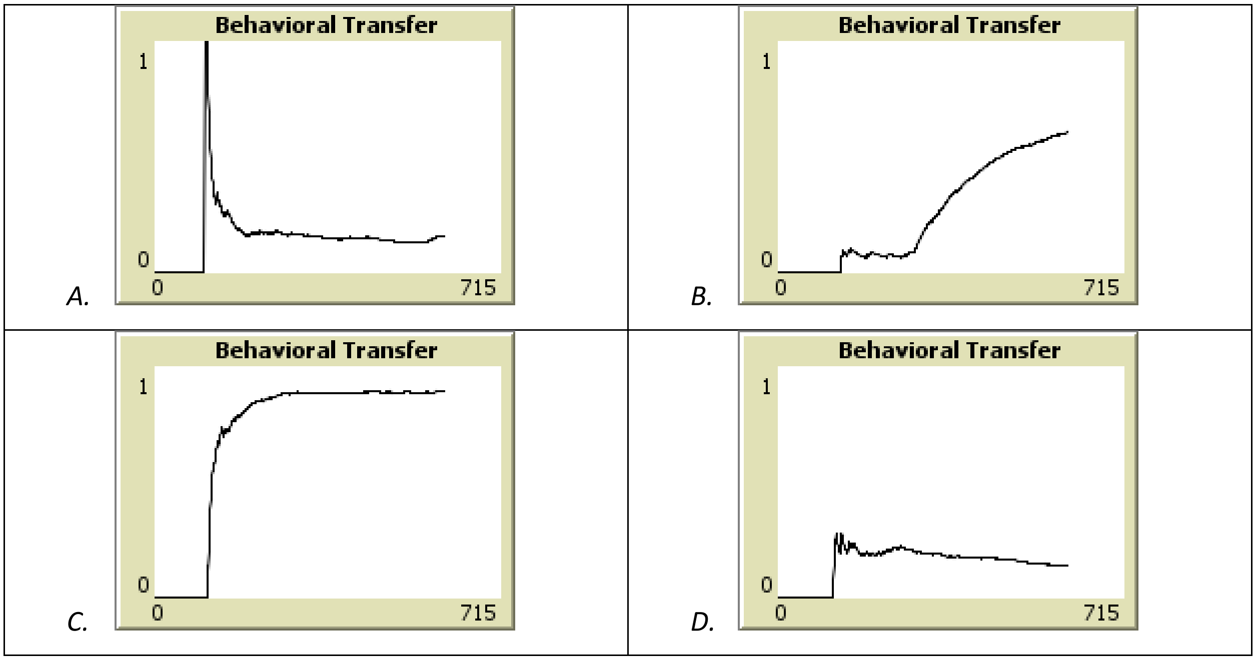


Figure 11. Exploration rate effect on behavioral transfer in Model 1.

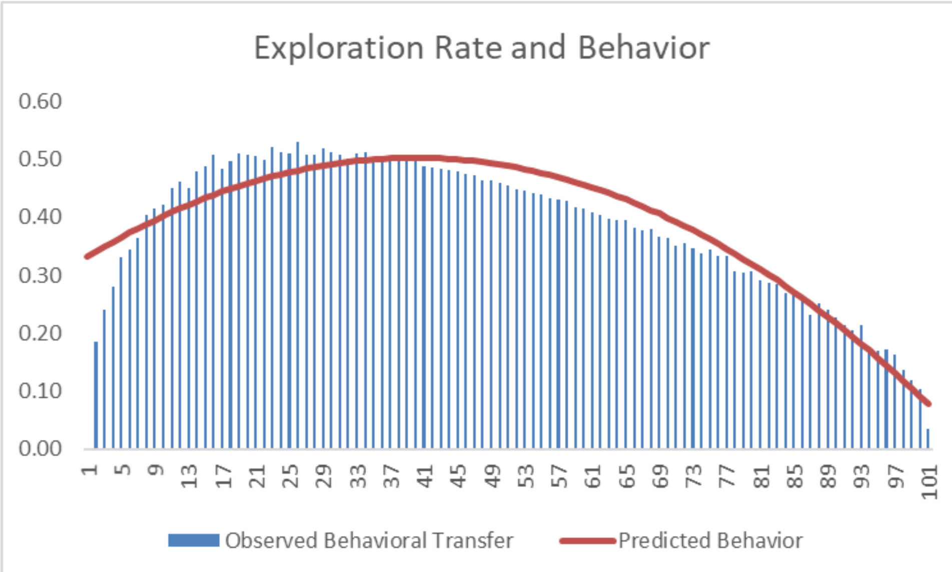


Figure 12. *Exploration rate effect on performance change in Model 1.*

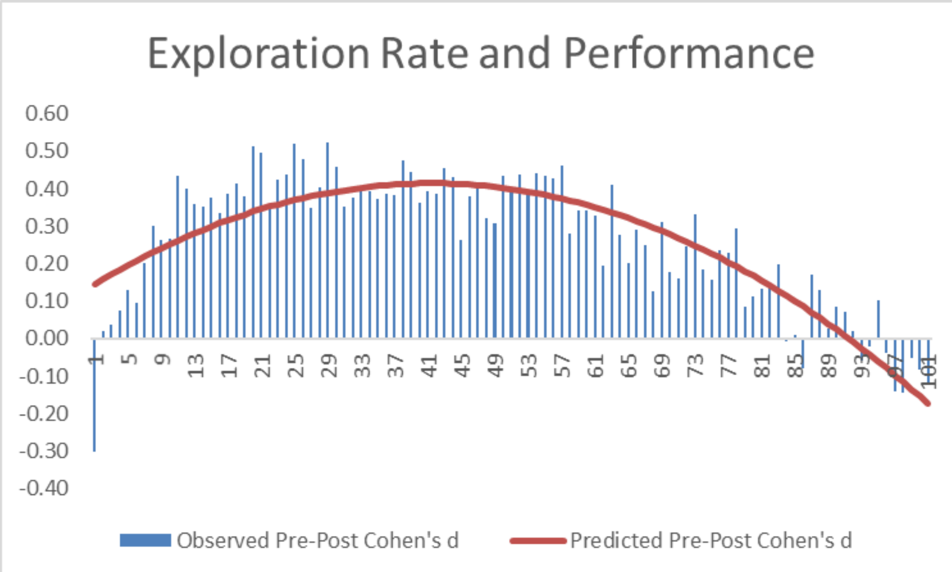


Figure 13. Type 2 likelihood vs implementation intention experimental effect on behavioral transfer in Model 1.

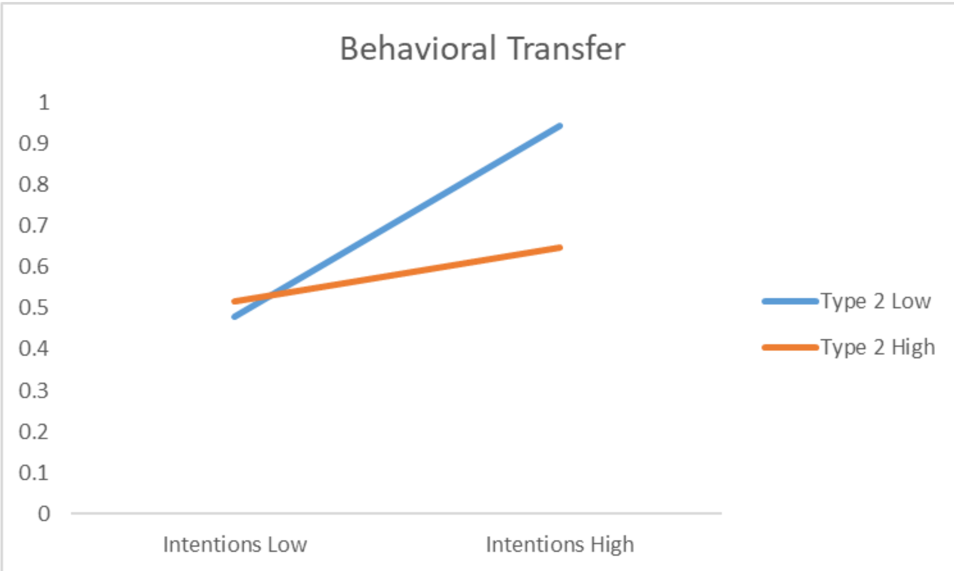


Figure 14. Type 2 likelihood vs implementation intention experimental effect on performance change in Model 1.

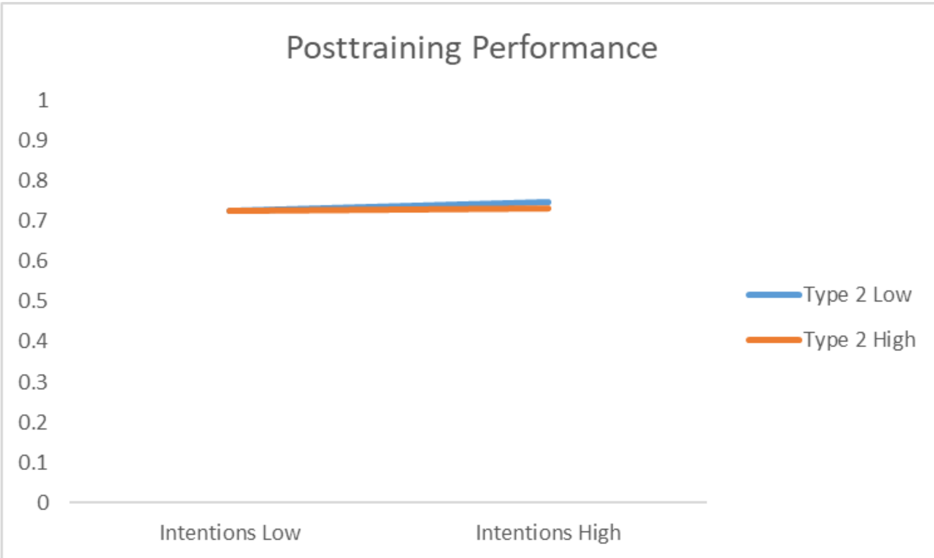


Figure 15. Type 2 likelihood vs implementation intention experimental effect on behavioral transfer in Model 1 heat map.

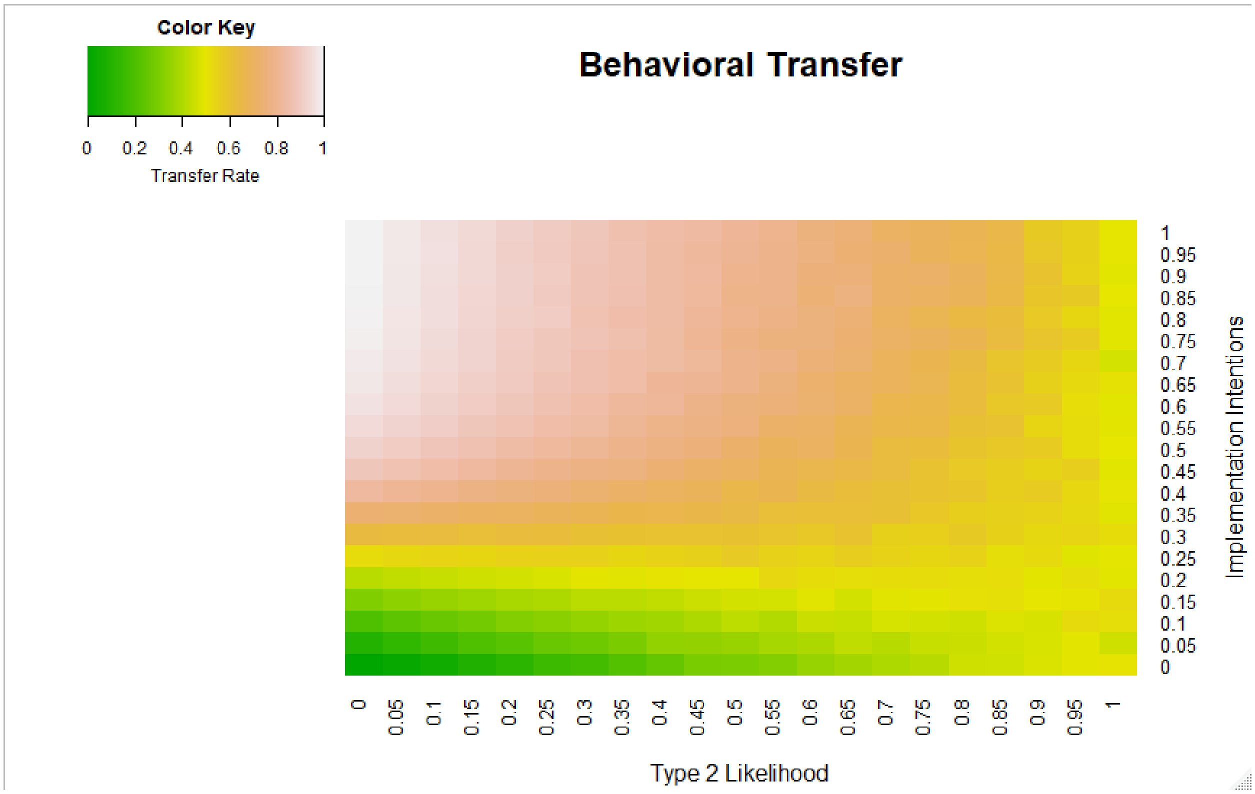


Figure 16. Type 2 likelihood vs implementation intention experimental effect on post training performance in Model 1 heat map.

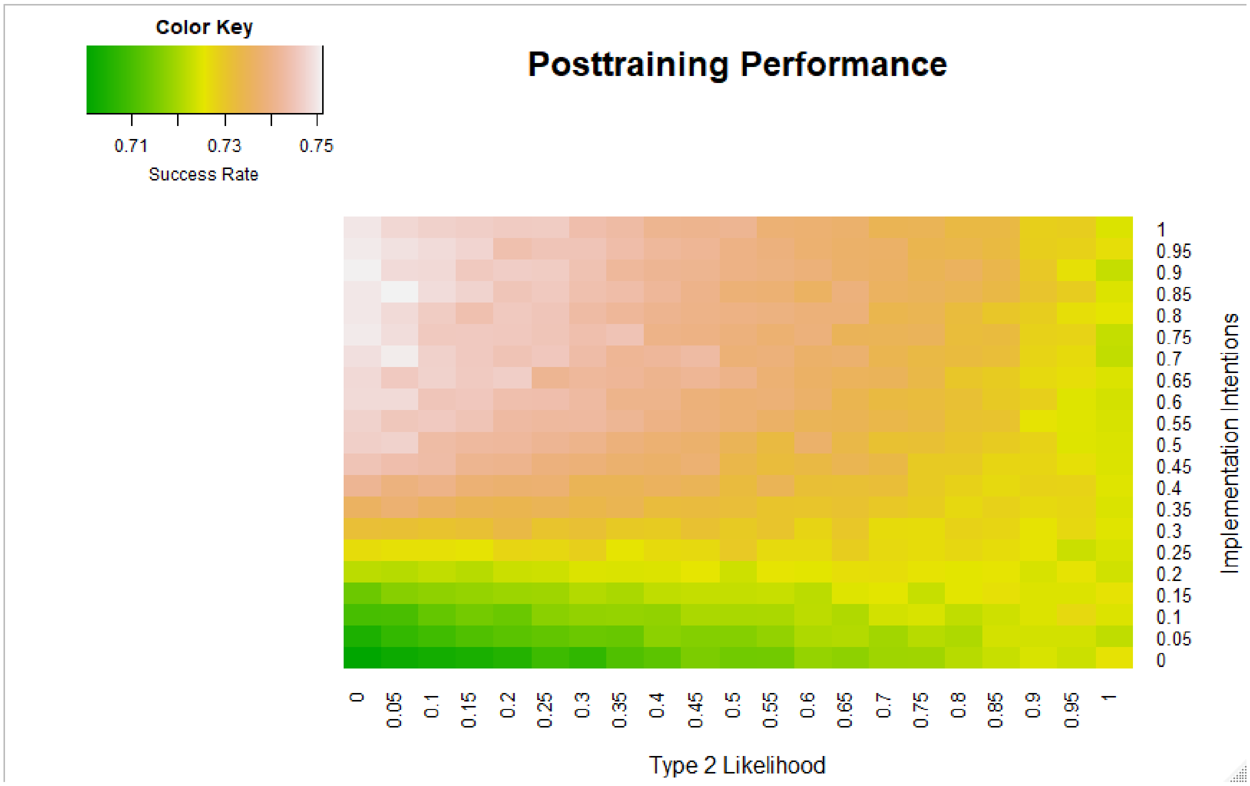


Figure 17. Type 2 likelihood vs implementation intention experimental effect on performance change in Model 1 heat map.

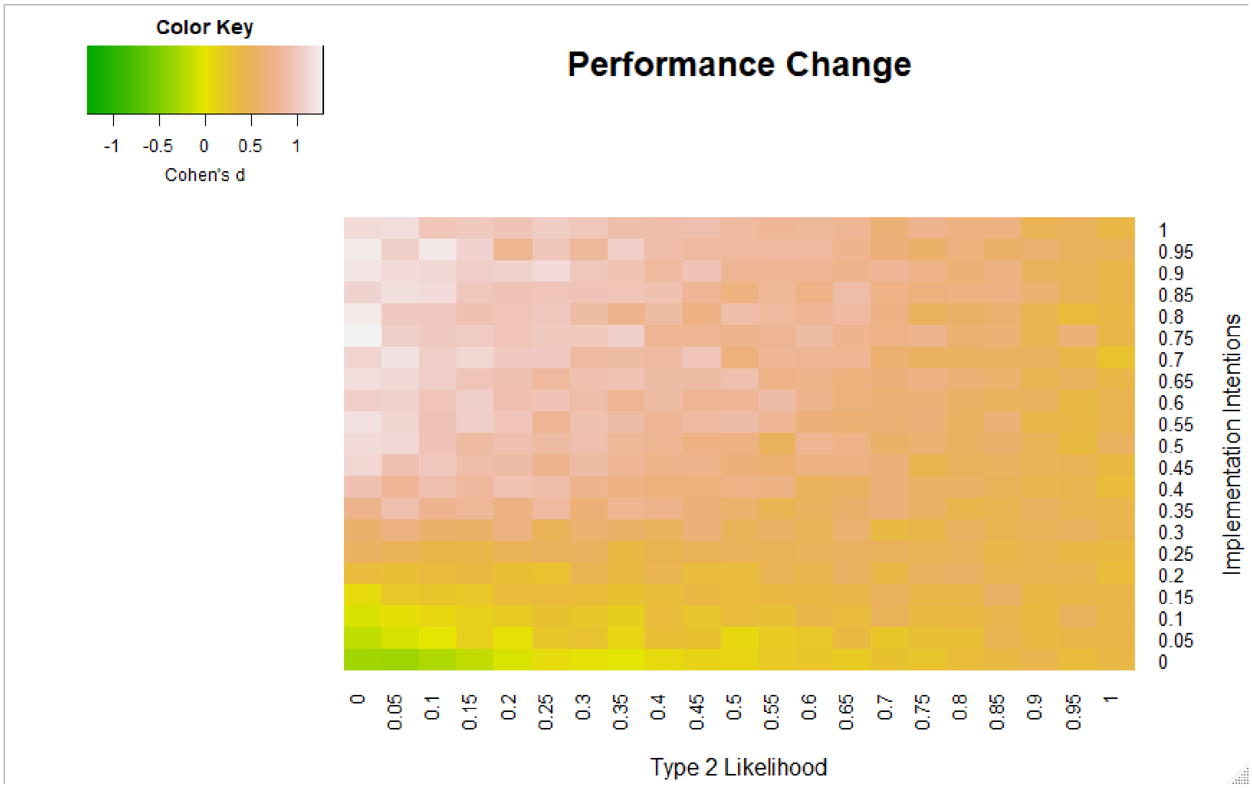


Figure 18. Proposed conceptual model for LTM with Social Learning.

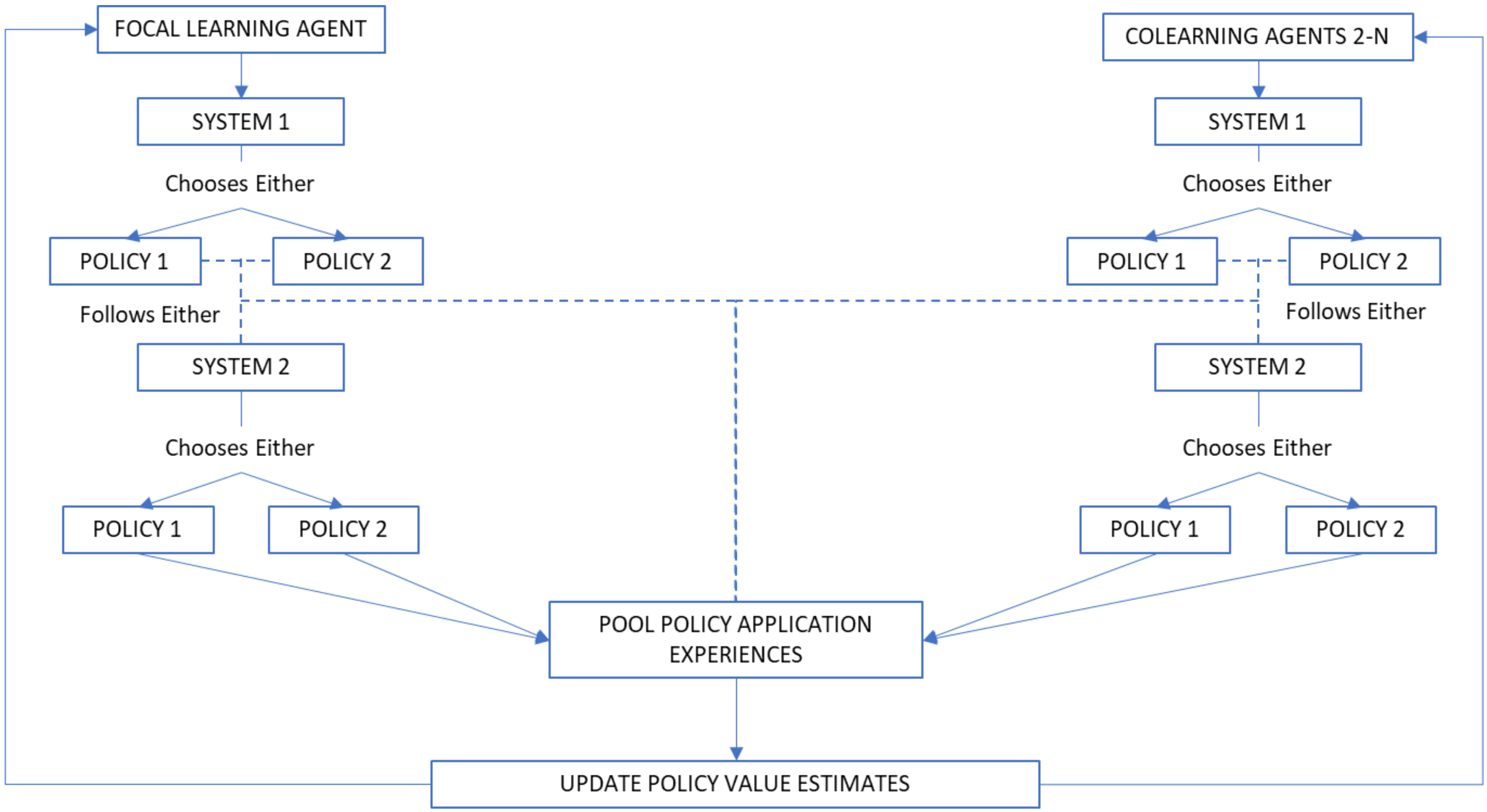


Figure 19. Heatmap of interaction effect of number of trainees and connectedness on behavioral transfer in Model 2A.

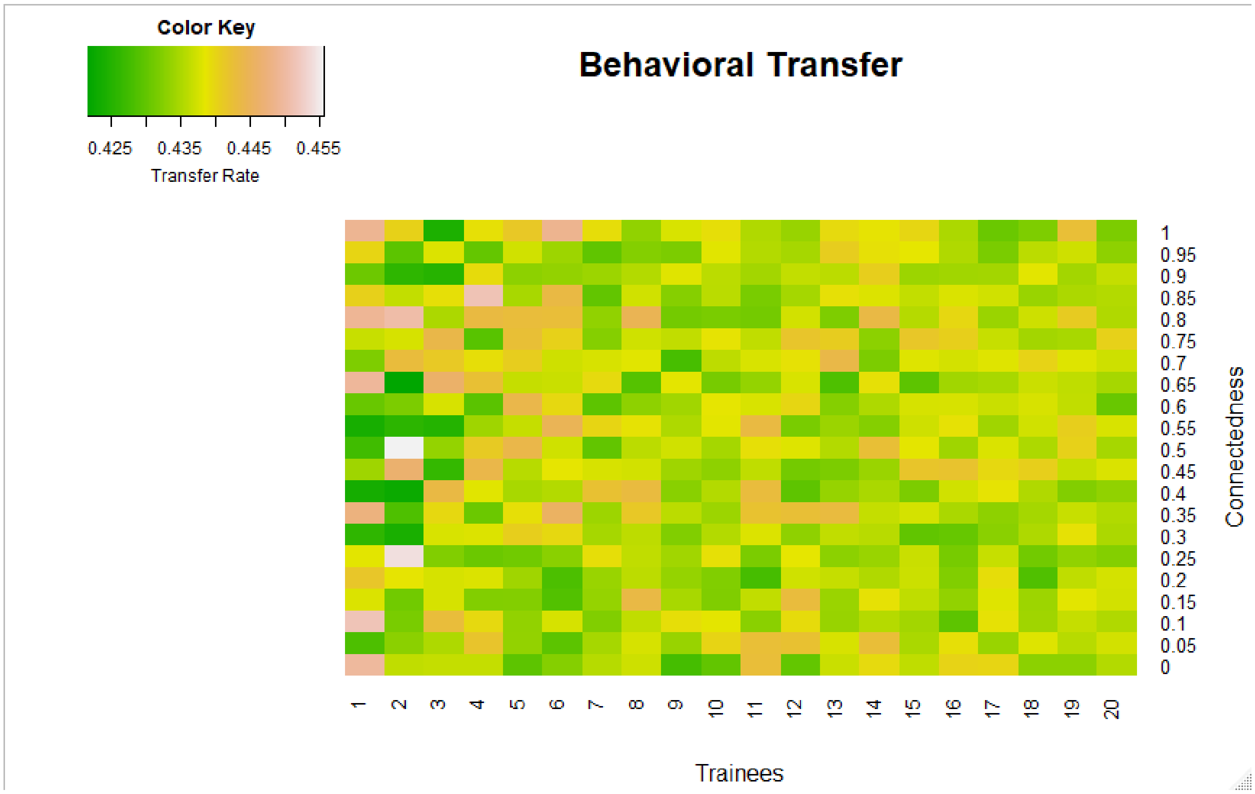


Figure 20. Number of trainees and level of imitation predicting behavioral transfer in Model 2B (replication level).

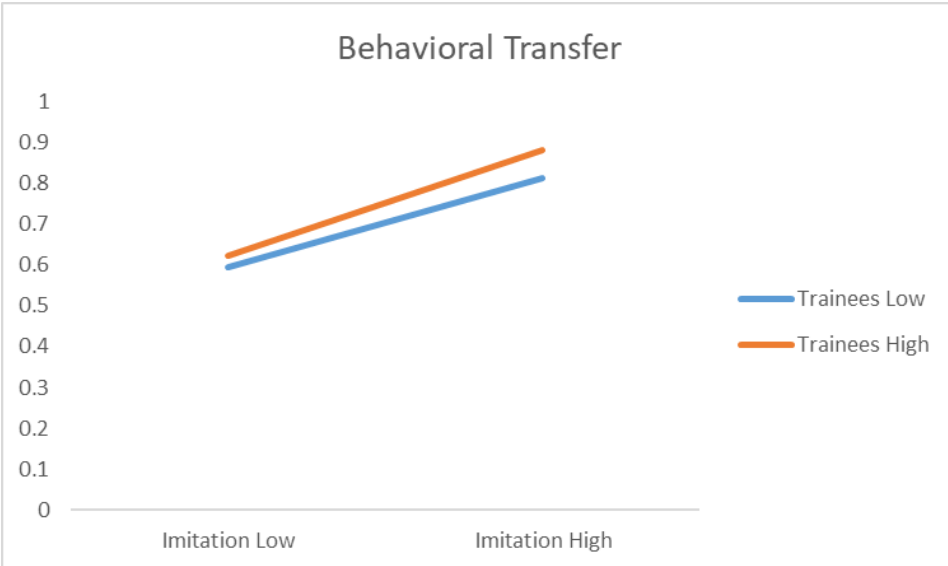


Figure 21. Number of trainees and level of imitation predicting post training performance in Model 2B (replication level).

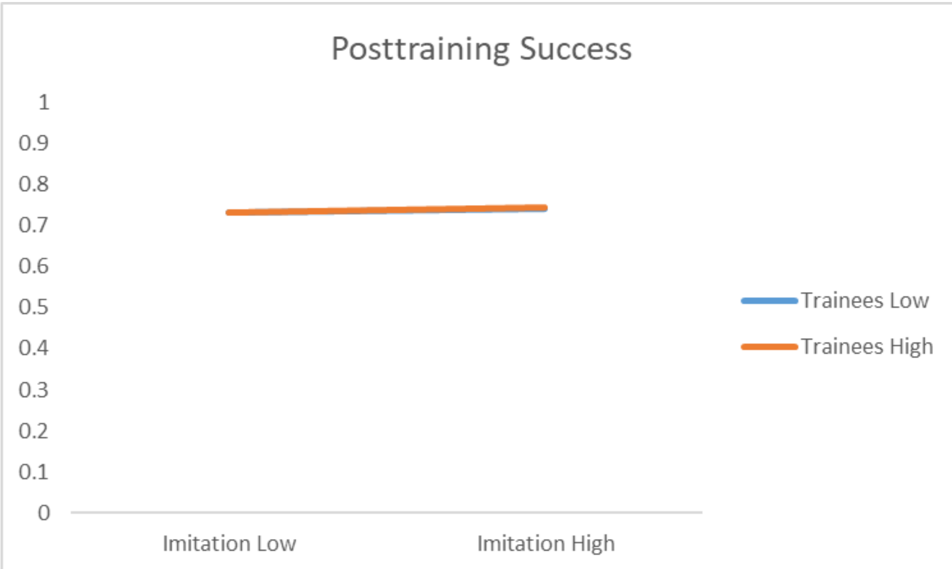


Figure 22. Number of trainees and level of imitation predicting pre-post training performance in Model 2B (condition level).

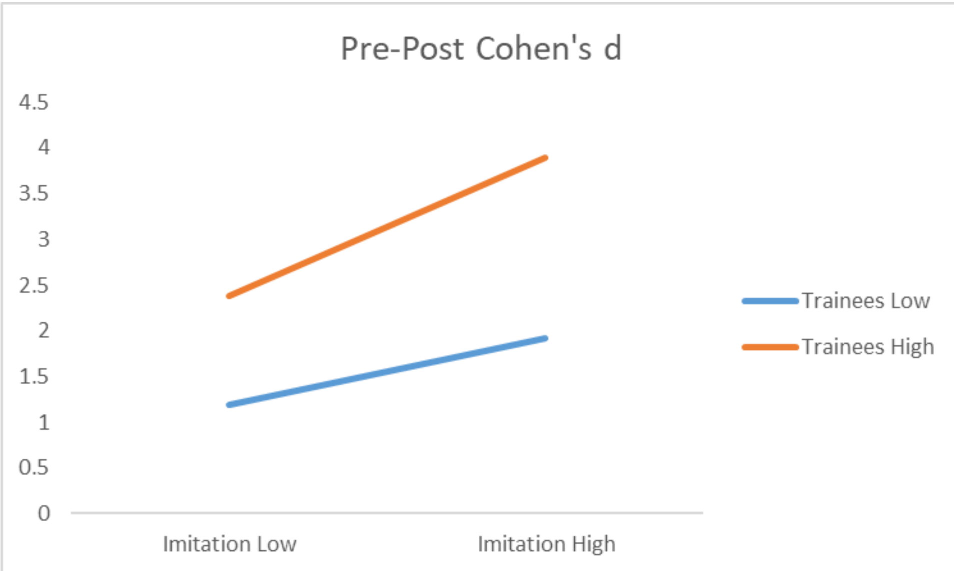


Figure 23. Heatmap of trainees and imitation predicting behavioral transfer in Model 2B.

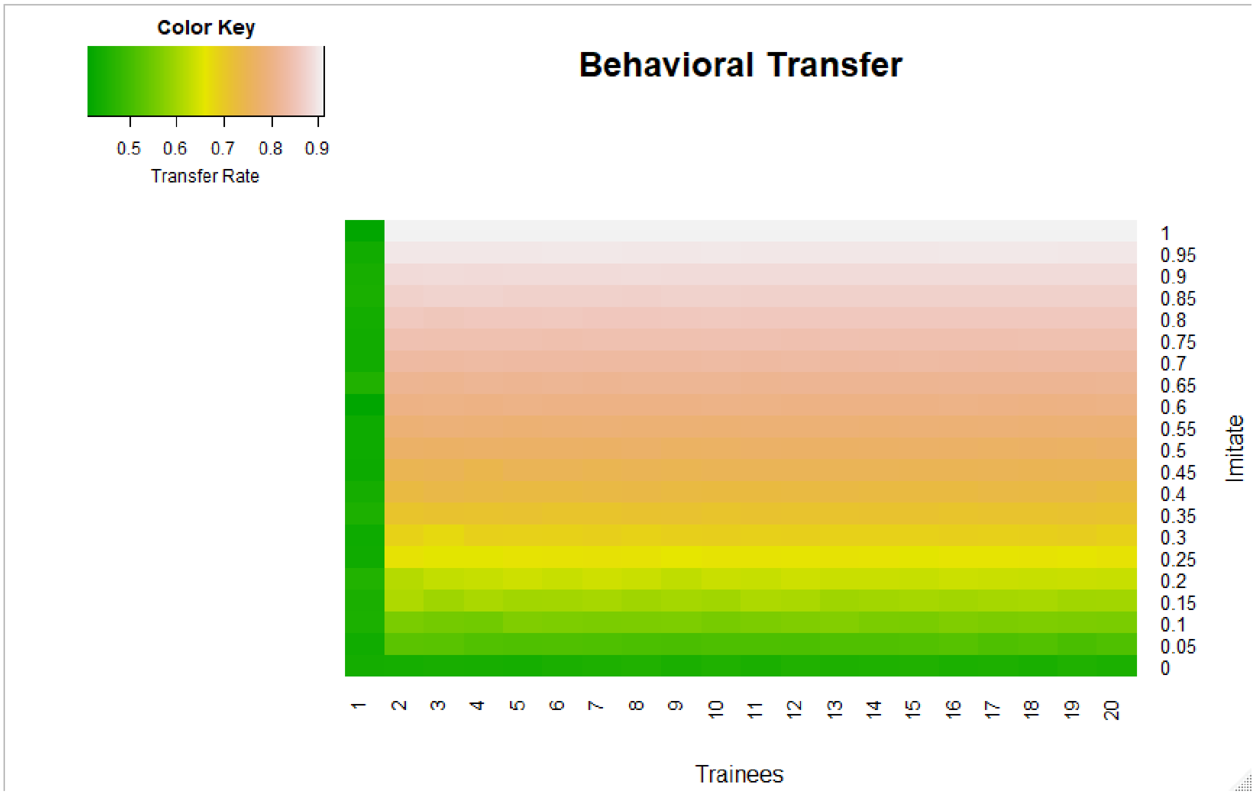


Figure 24. Heatmap of trainees and imitation predicting post training performance in Model 2B.

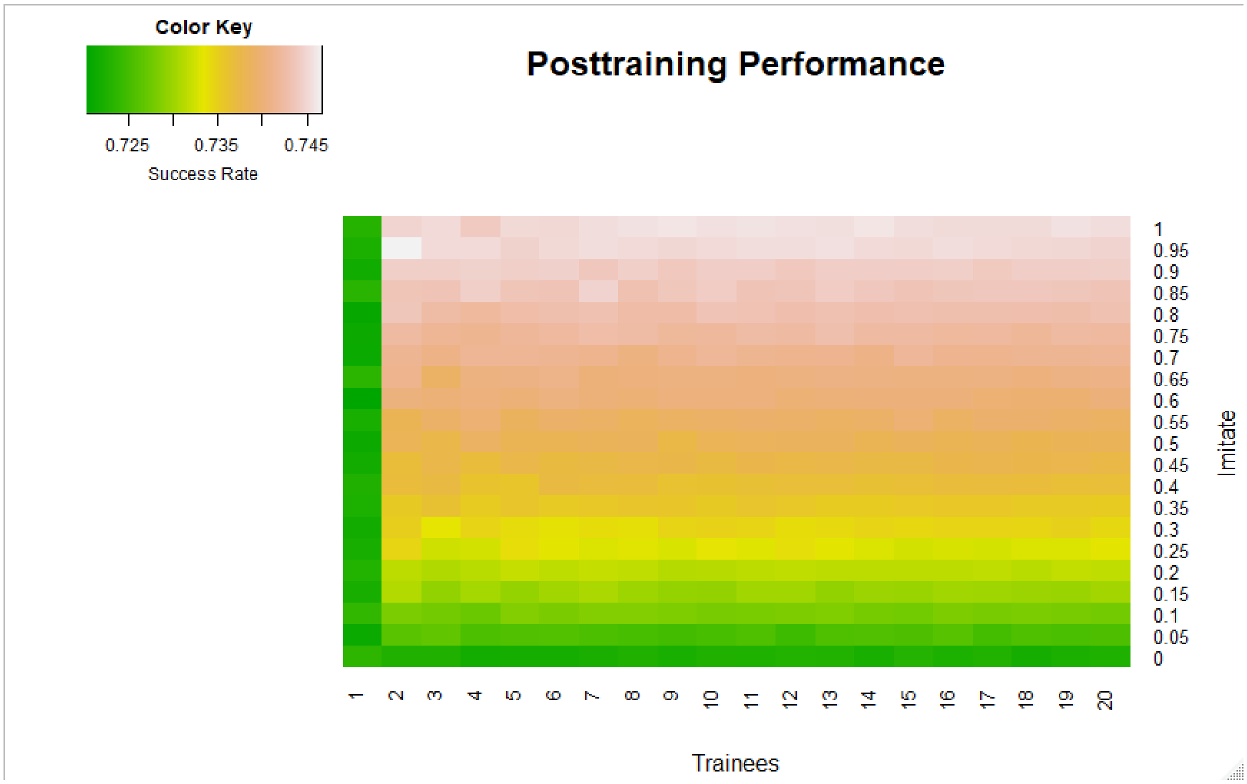


Figure 25. Heatmap of trainees and imitation predicting pre-post performance change in Model 2B.

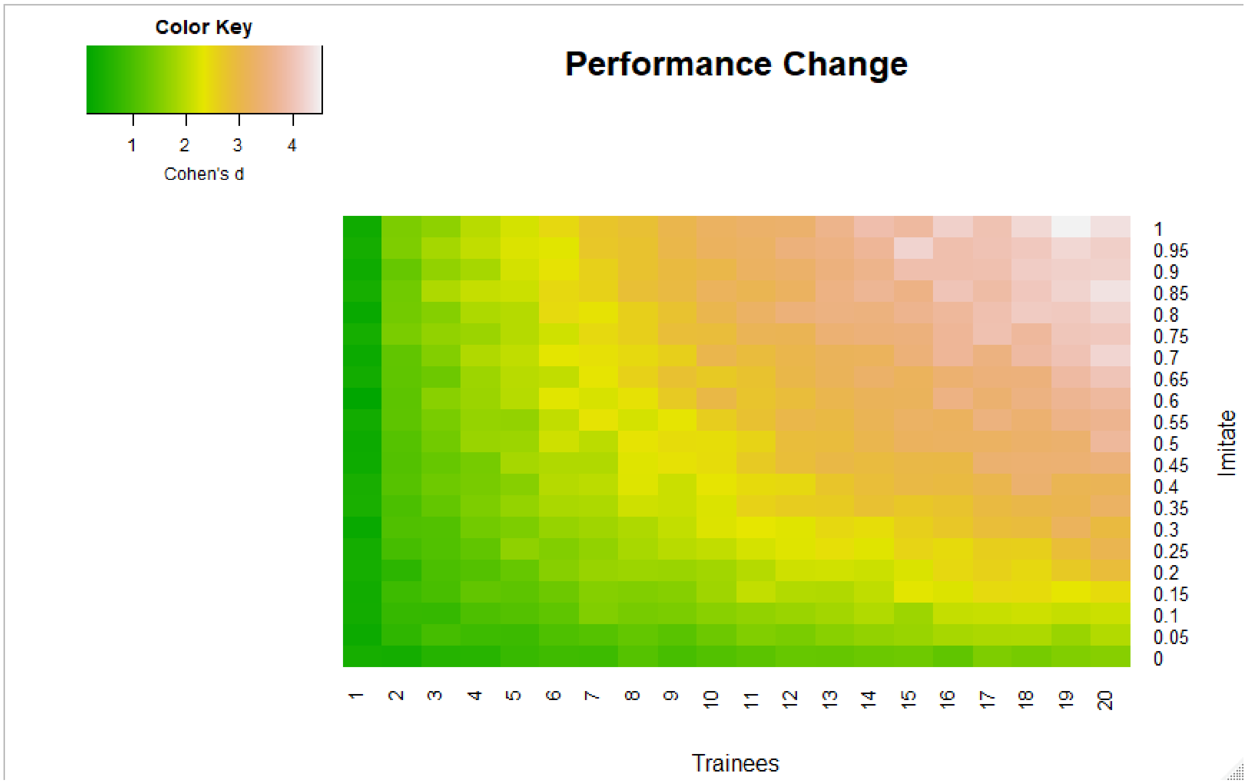


Figure 26. Number of trainees and level of conformity predicting behavioral transfer in Model 2C (replication level).

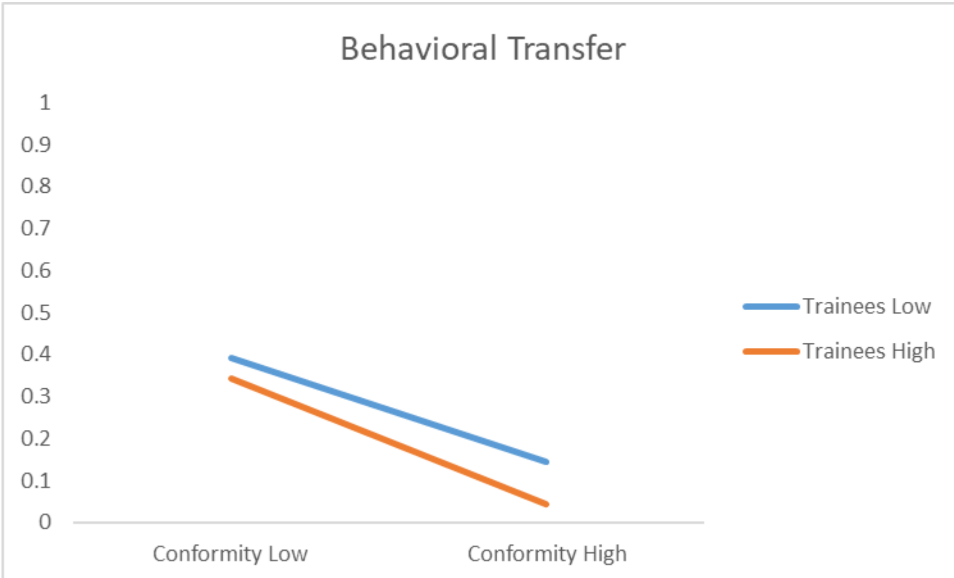


Figure 27. Number of trainees and level of conformity predicting post training performance in Model 2C (replication level).

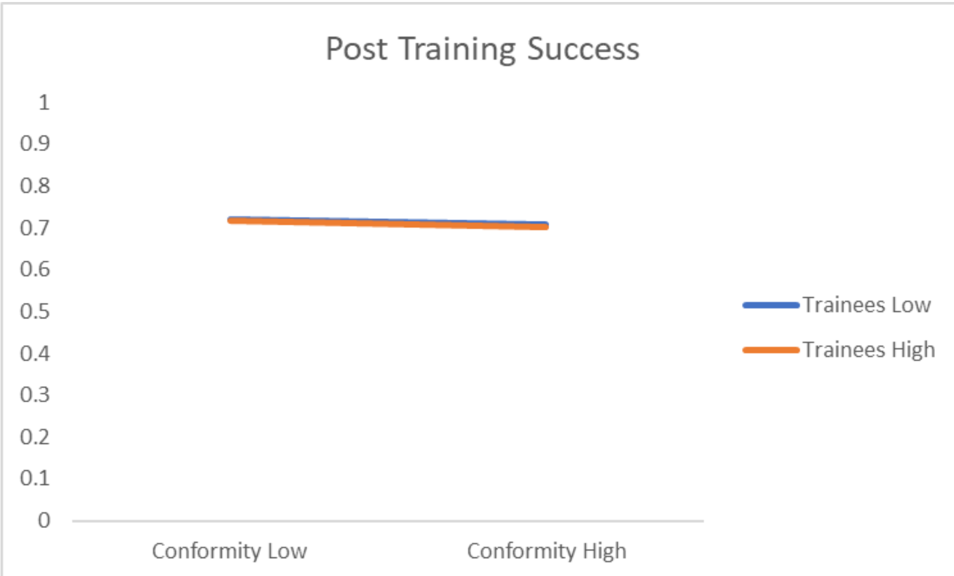


Figure 28. Number of trainees and level of conformity predicting pre-post performance change in Model 2C (condition level).

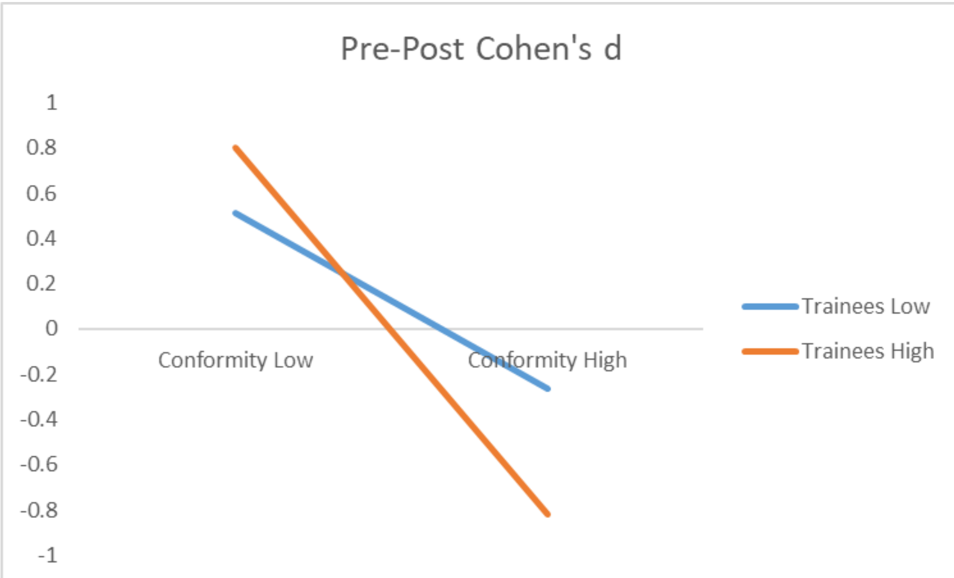


Figure 29. Heat map of number of trainees and level of conformity predicting behavioral transfer in Model 2C.

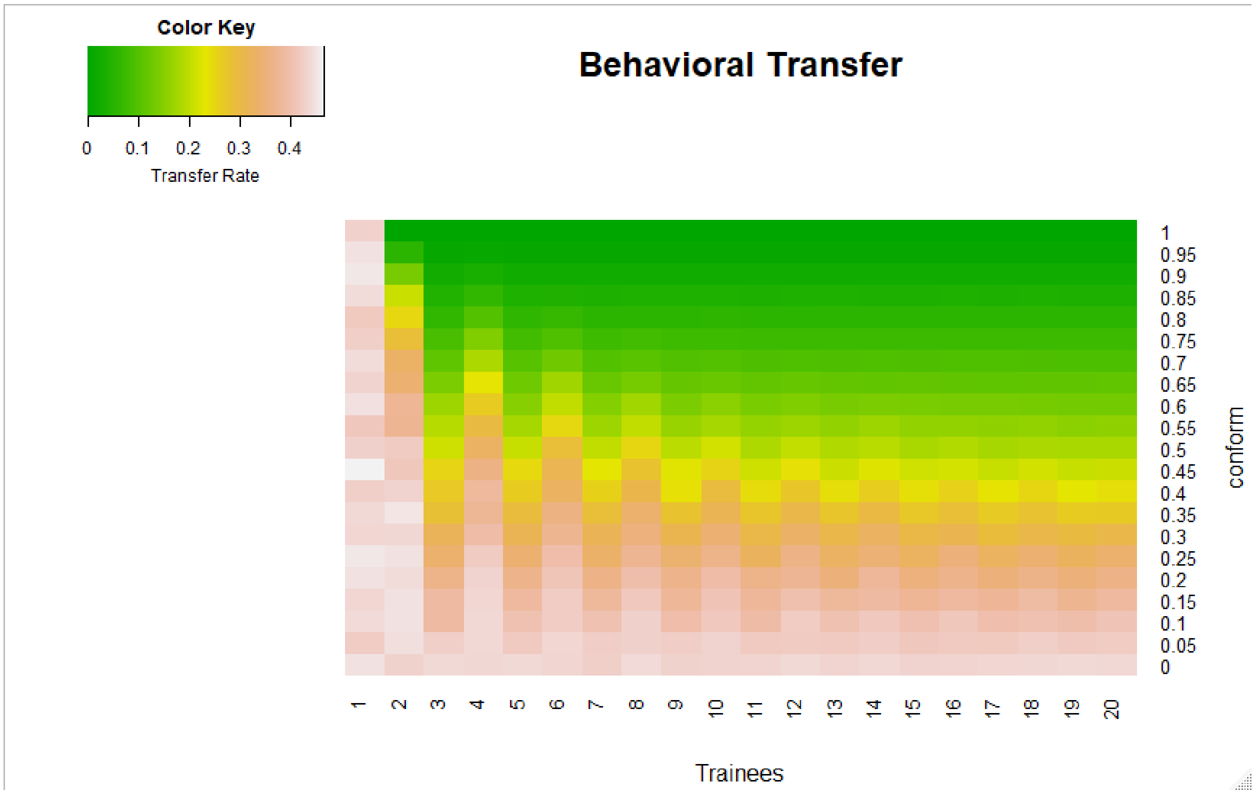


Figure 30. Heat map of number of trainees and level of conformity predicting post training performance in Model 2C.

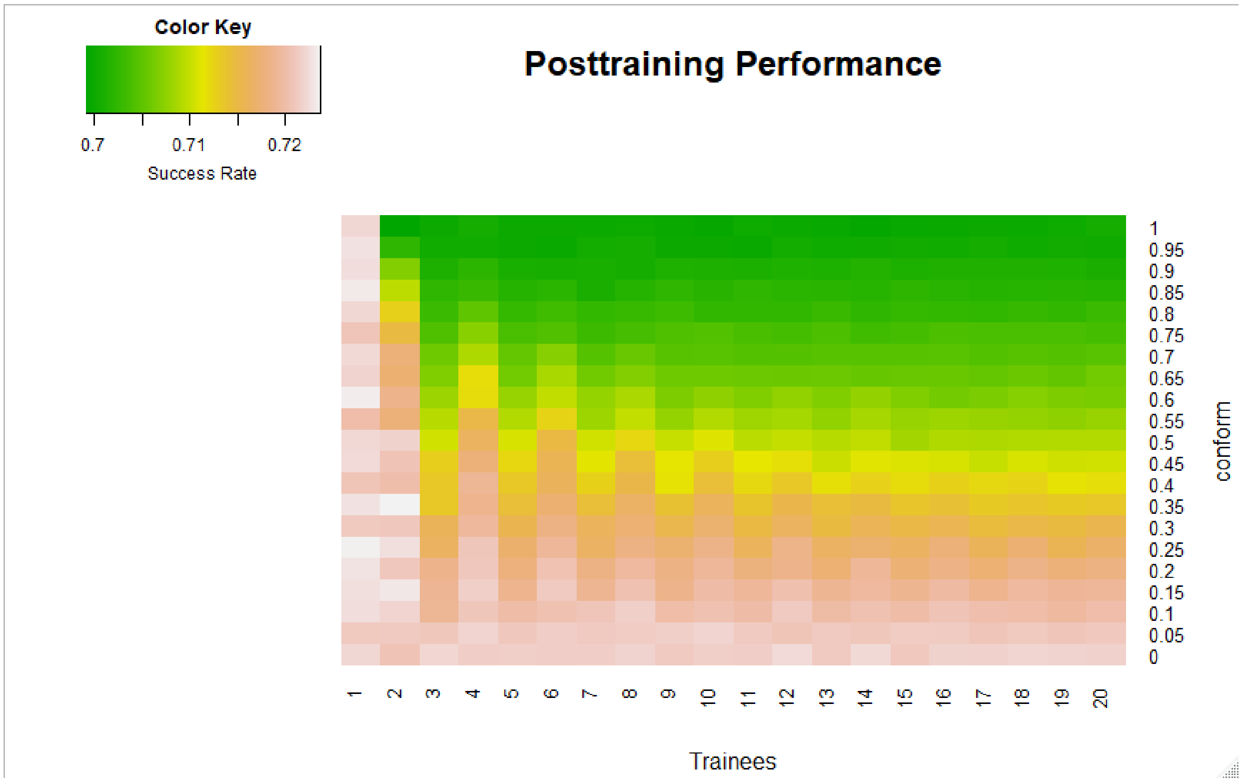


Figure 31. Heat map of number of trainees and level of conformity predicting pre-post performance change in Model 2C.

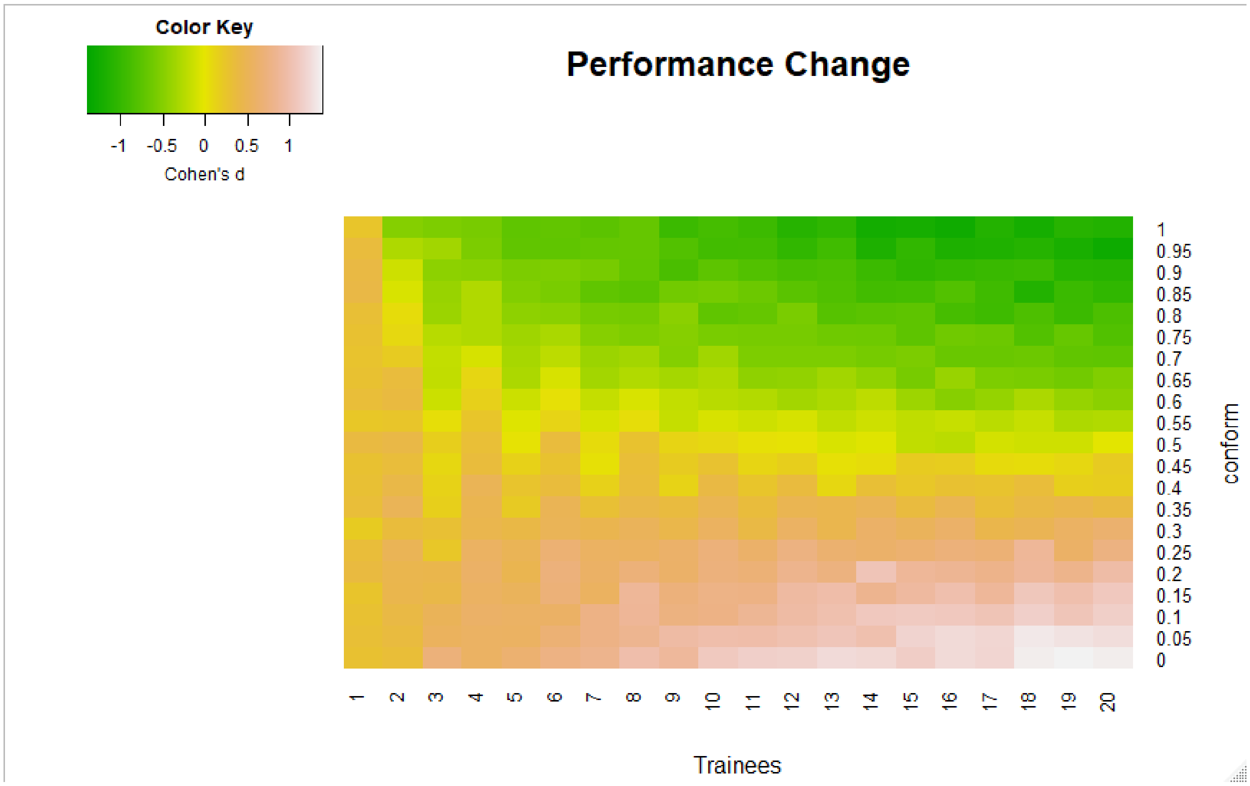


Figure 32. Conceptual model for LTM including self-regulation.

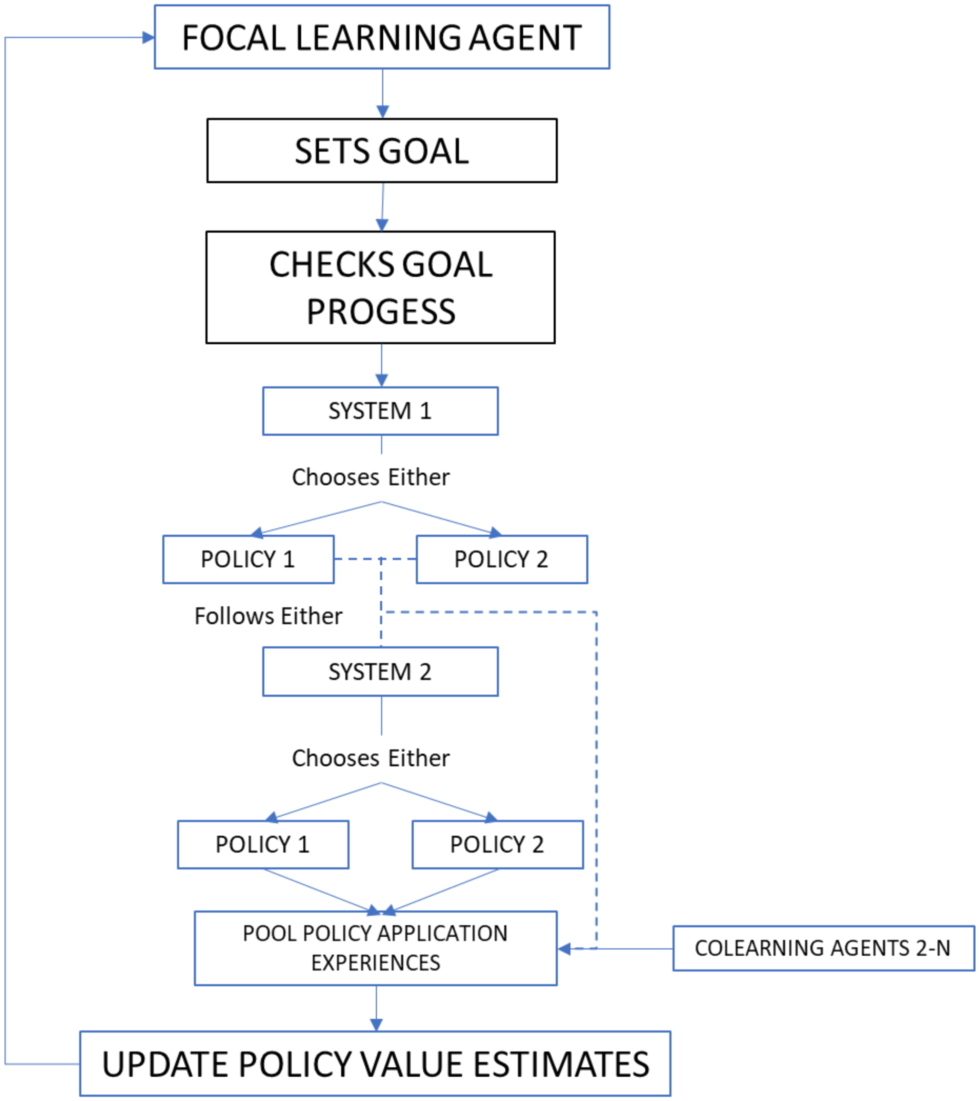


Figure 33. Goal level and exploration rate change predicting post training performance in Model 3A (replication level).

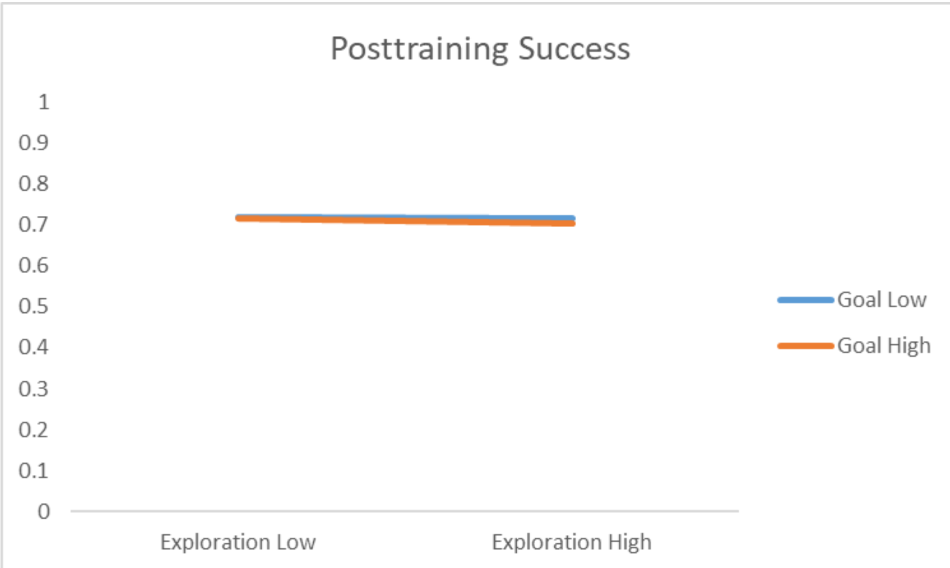


Figure 34. Goal level and exploration rate change predicting behavioral transfer in Model 3A (replication level).

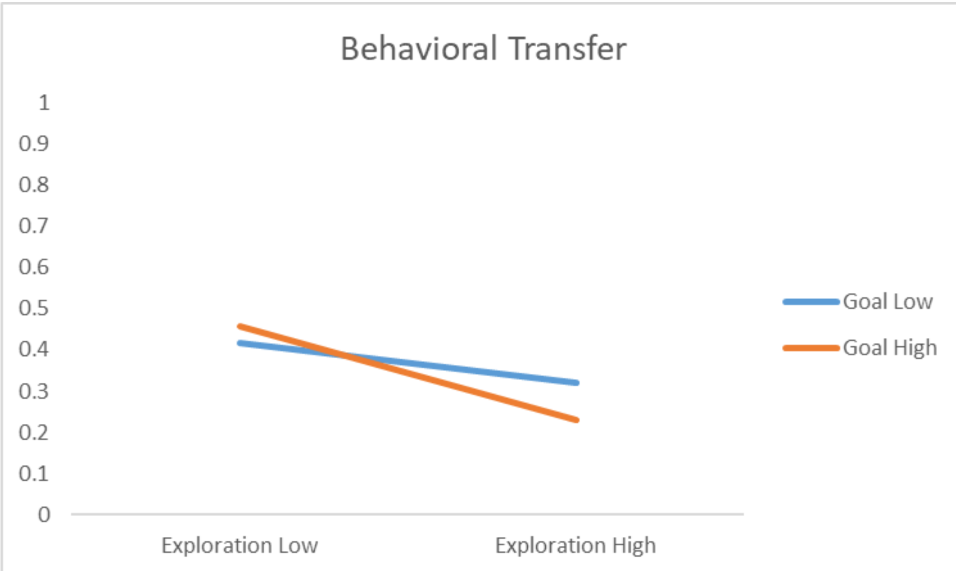


Figure 35. Goal level and exploration rate change predicting pre-post performance change in Model 3A (condition level).

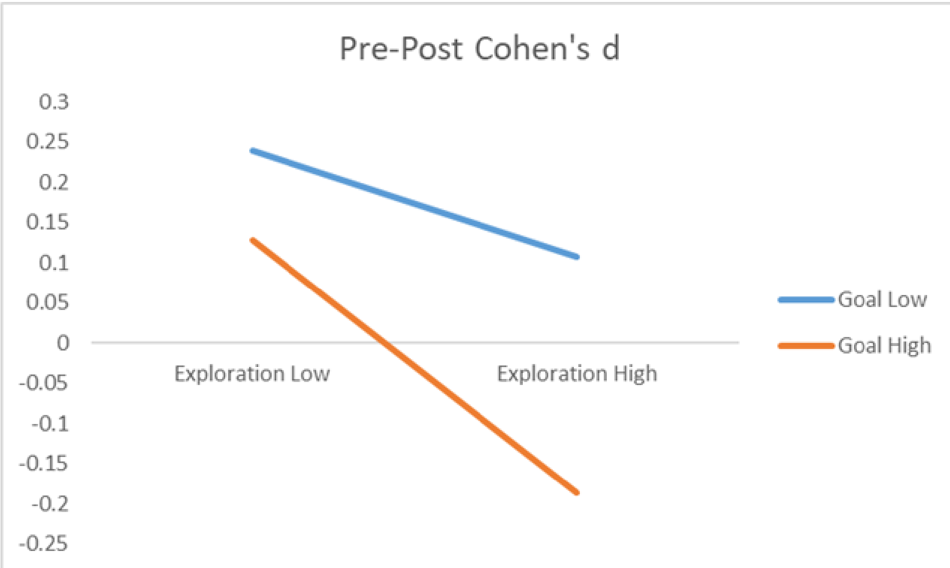


Figure 36. Heat map of goal level and exploration rate change predicting behavioral transfer in Model 3A.

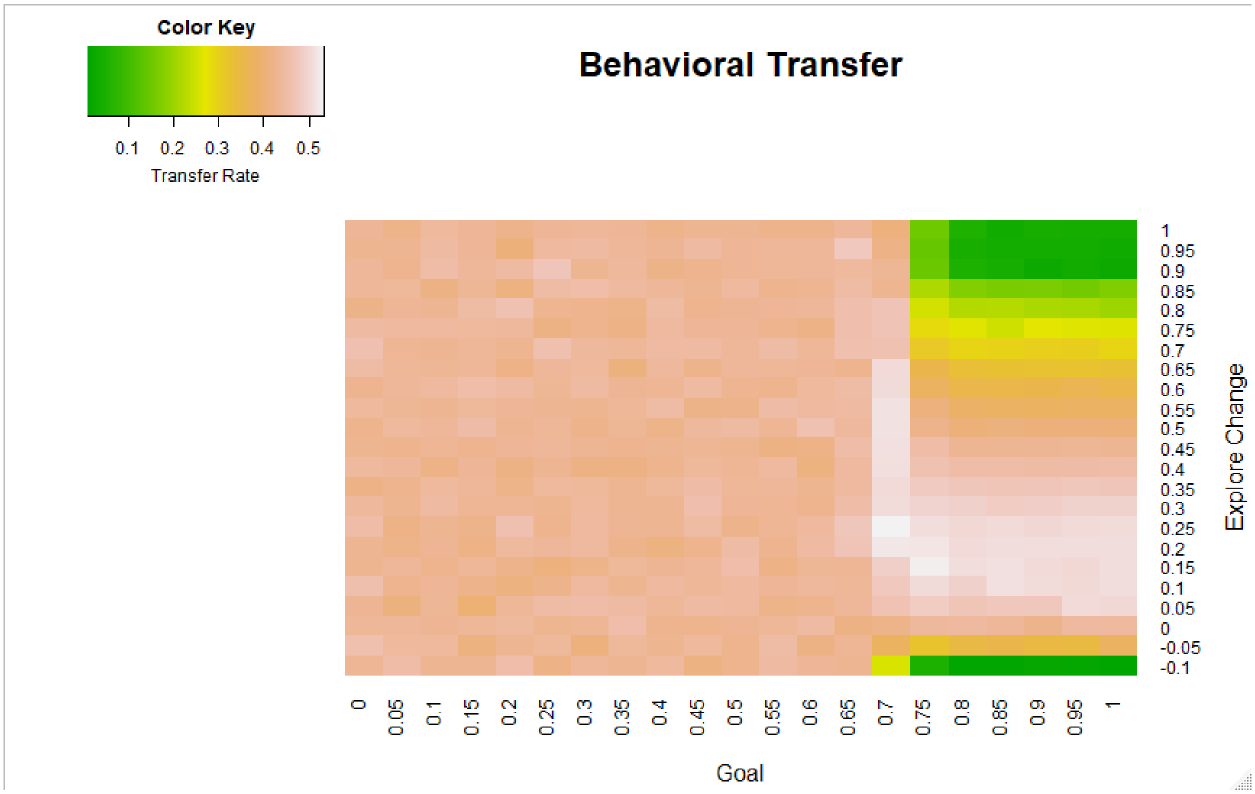


Figure 37. Heat map of goal level and exploration rate change predicting post training performance in Model 3A.

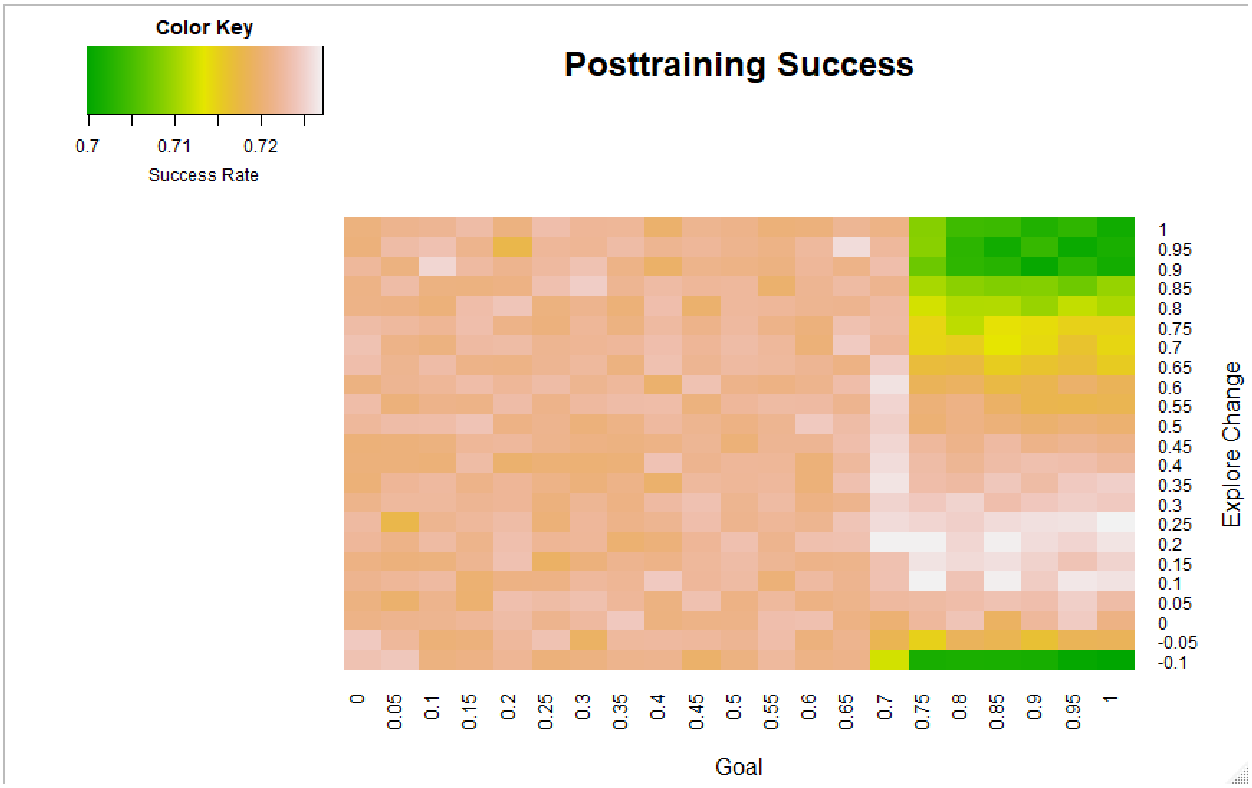


Figure 38. Heat map of goal level and exploration rate change predicting pre-post performance change in Model 3A.

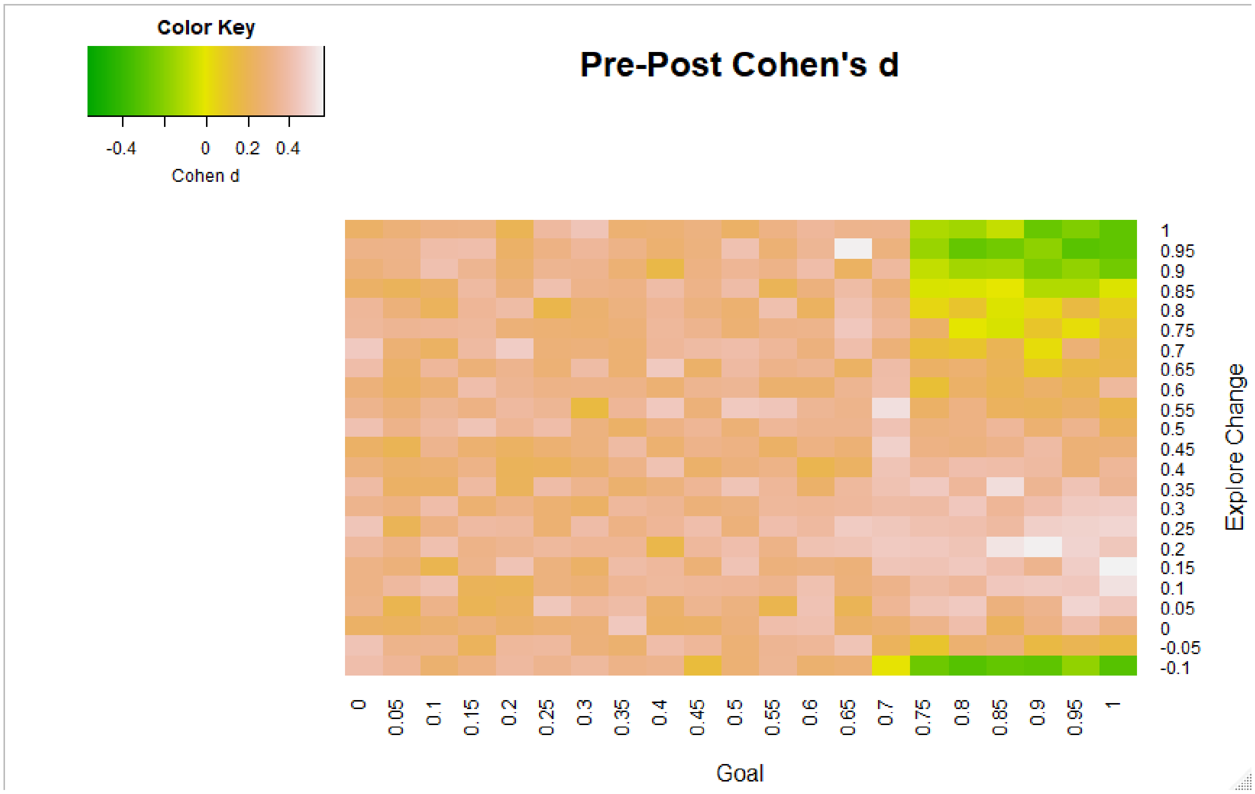
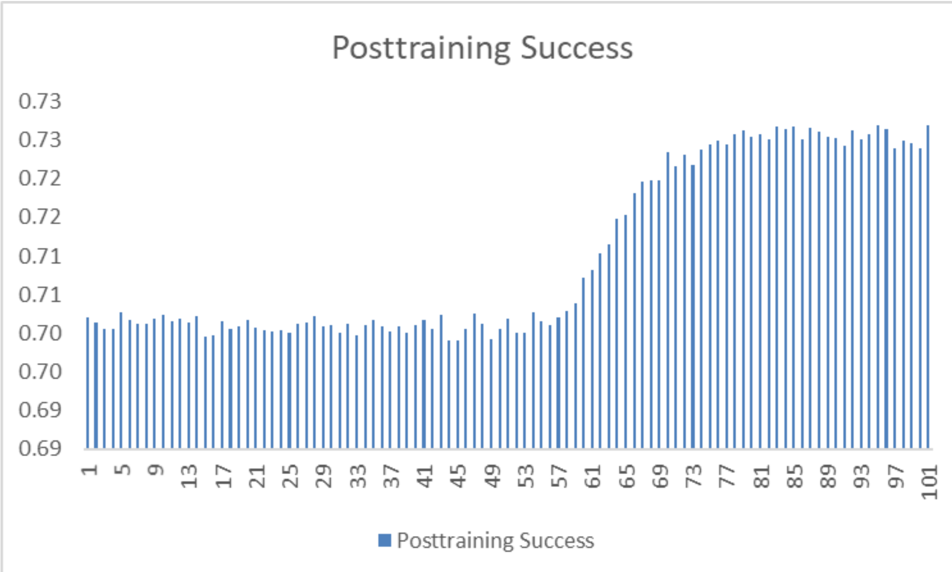


Figure 39. Observed post training performance by goal level in Model 3B-1.



Note scale intentionally not starting at 0 to show sudden shift in percentages more clearly.

Figure 40. Observed behavioral transfer by goal level in Model 3B-1.

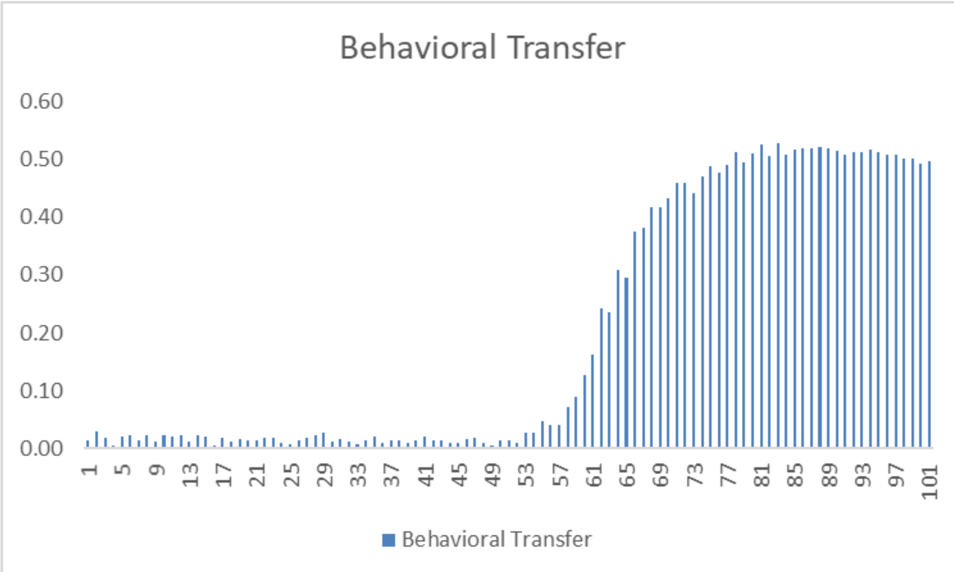


Figure 41. Observed pre-post performance change by goal level in Model 3B-1.

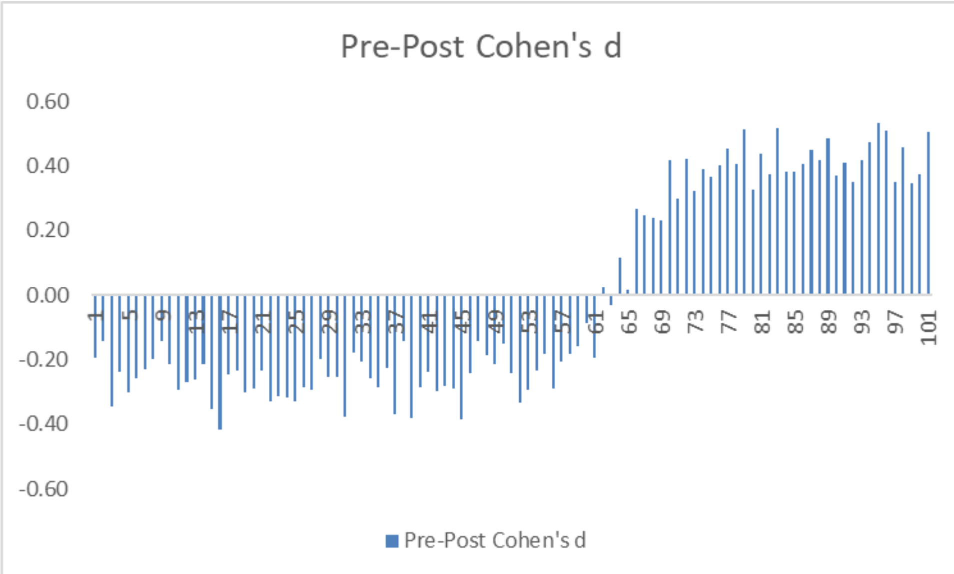


Figure 42. Goal level and policy value change predicting behavioral transfer in Model 3B-1 (replication level).

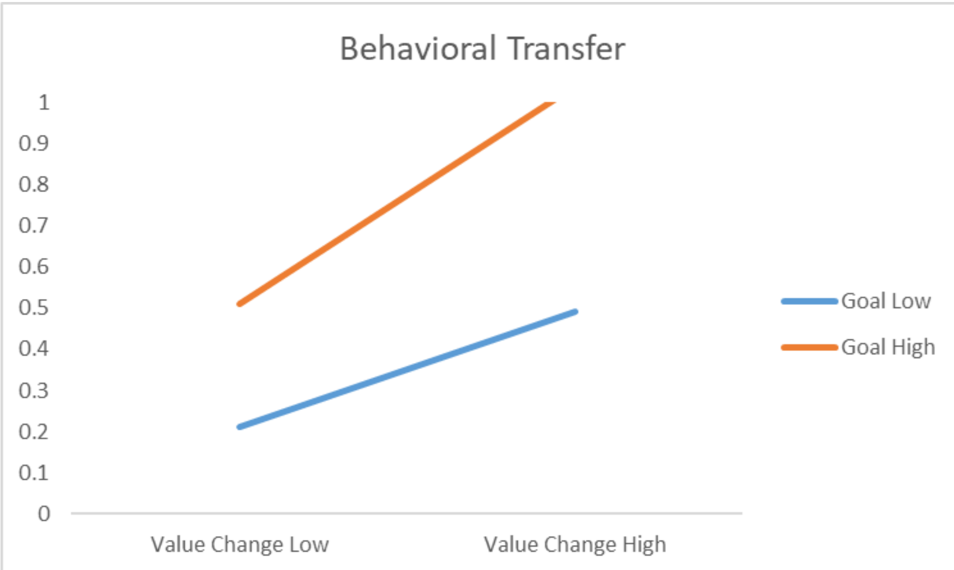


Figure 43. Goal level and policy value change predicting post training performance in Model 3B-1 (replication level).

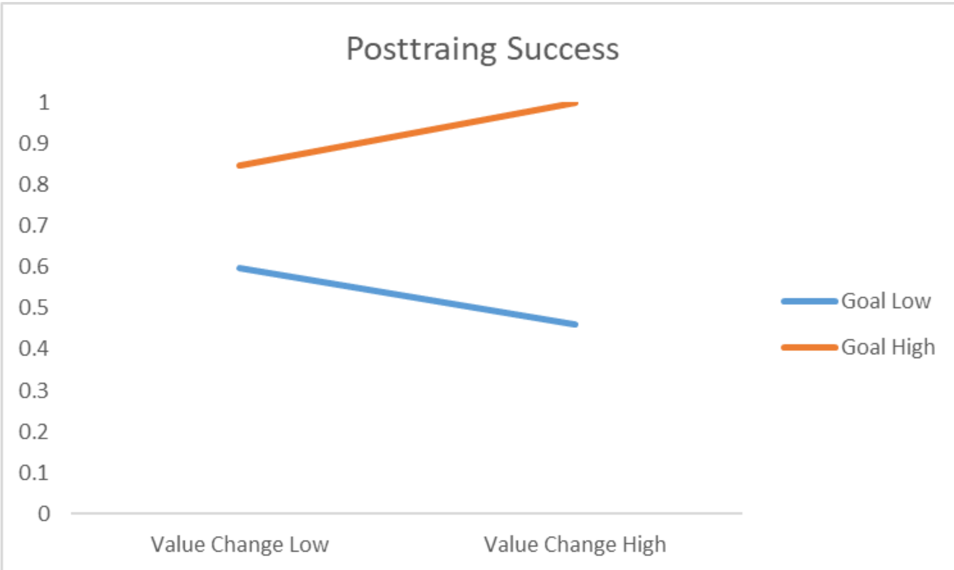


Figure 44. Goal level and policy value change predicting pre-post performance change in Model 3B-1 (condition level).

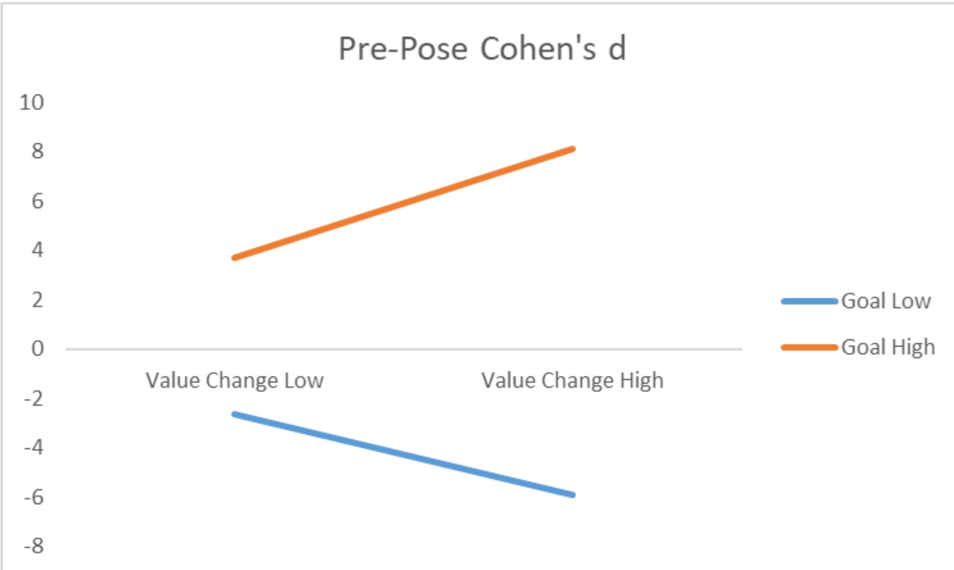


Figure 45. Heat map of goal level and policy value change predicting behavioral transfer in Model 3B-1.

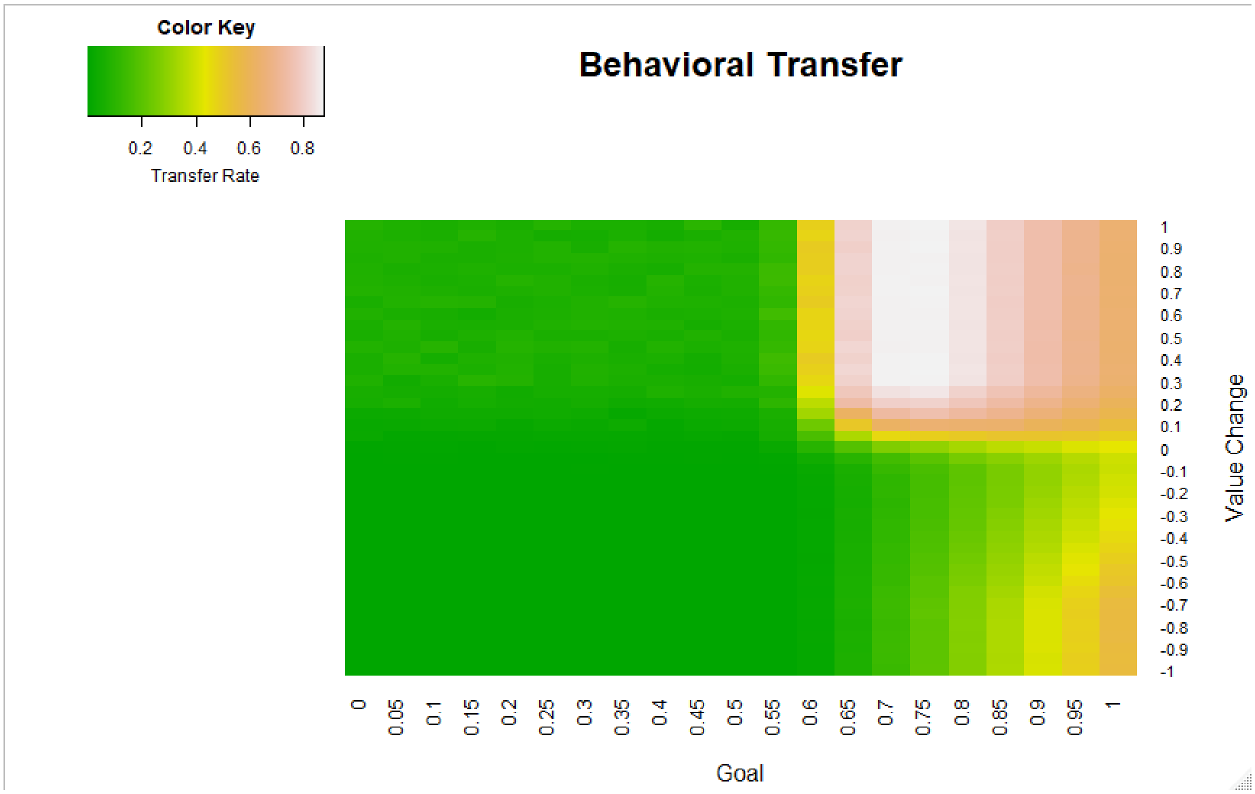


Figure 46. Heat map of goal level and policy value change predicting post training performance in Model 3B-1.

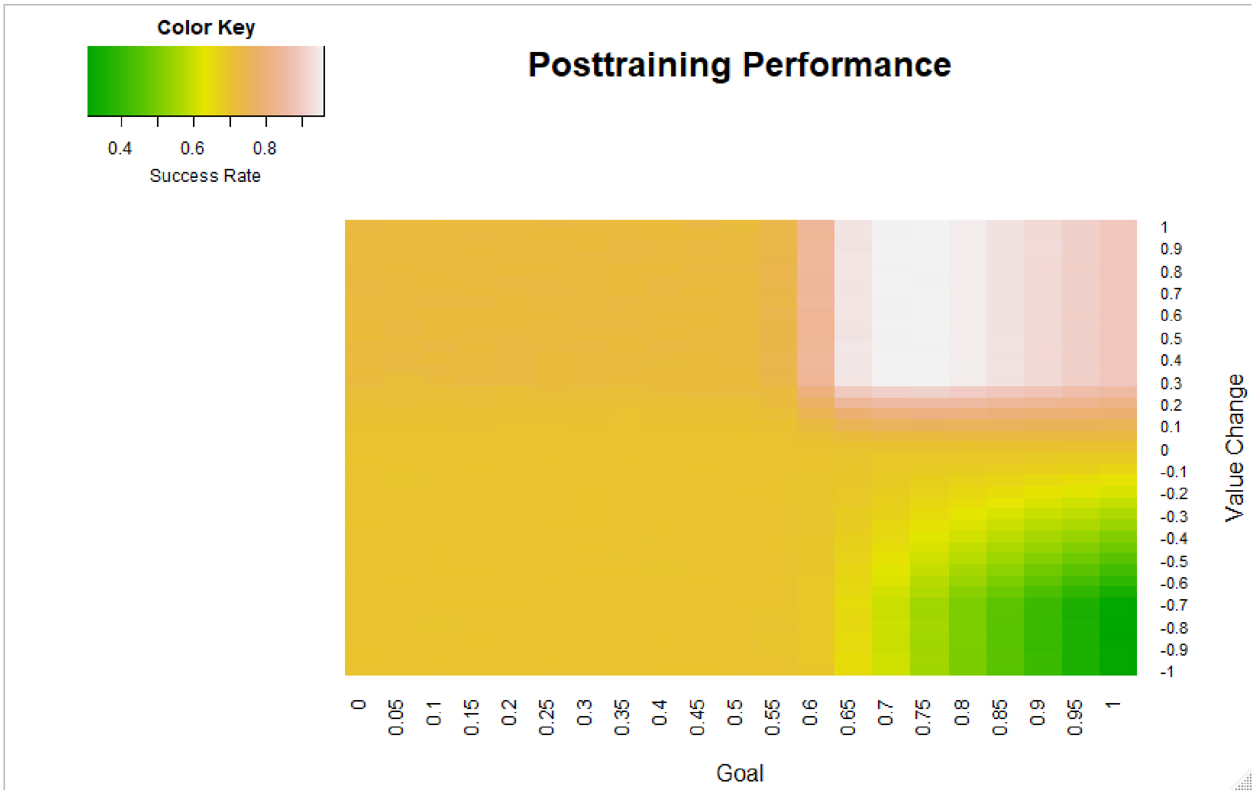


Figure 47. Heat map of goal level and policy value change predicting pre-post performance change in Model 3B-1.

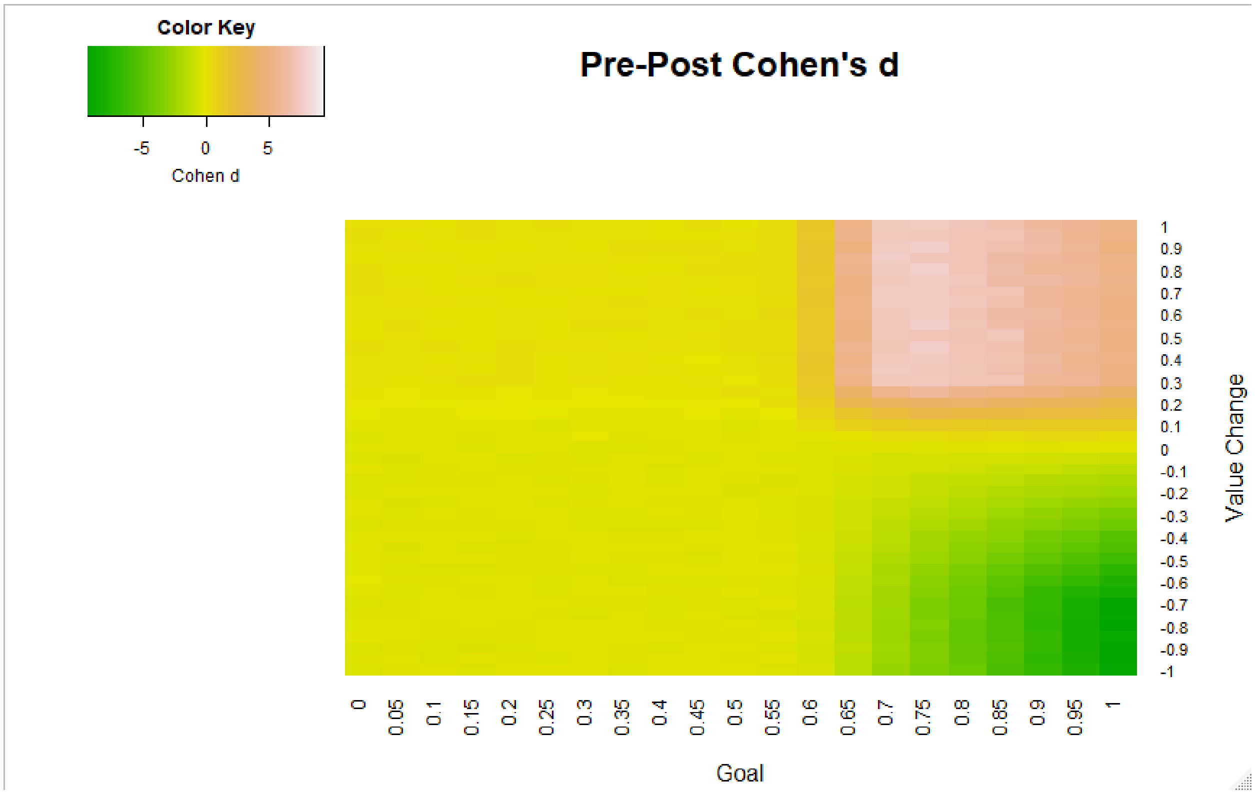
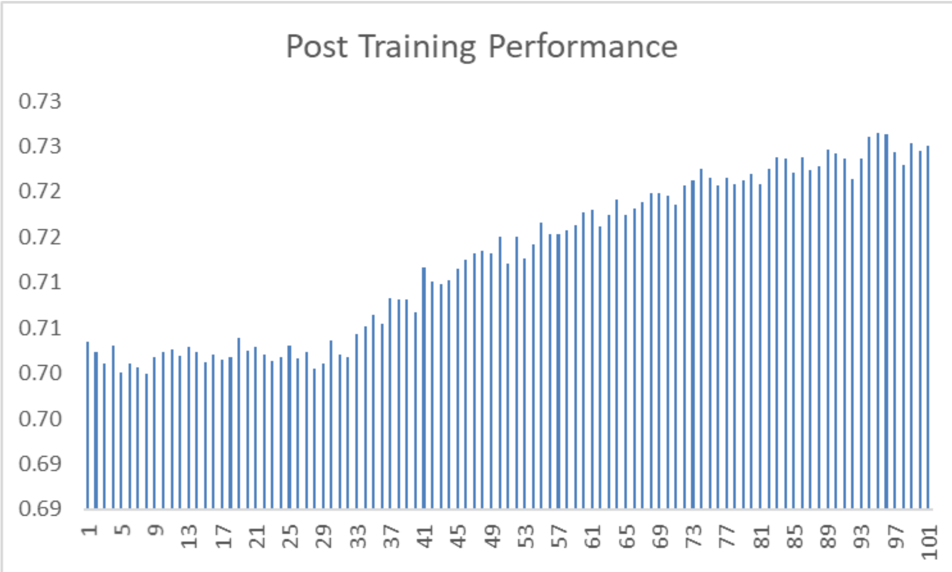


Figure 48. Observed post training performance by goal level in Model 3B-2.



Note scale intentionally not starting at 0 to show sudden shift in percentages more clearly.

Figure 49. Observed behavioral transfer by goal level in Model 3B-2.

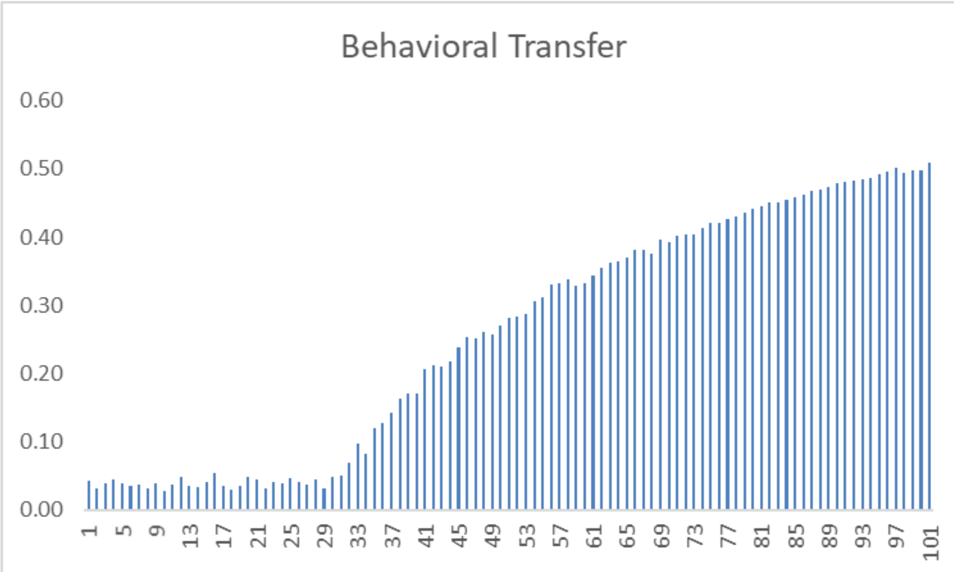


Figure 50. Observed pre-post performance change by goal level in Model 3B-2.

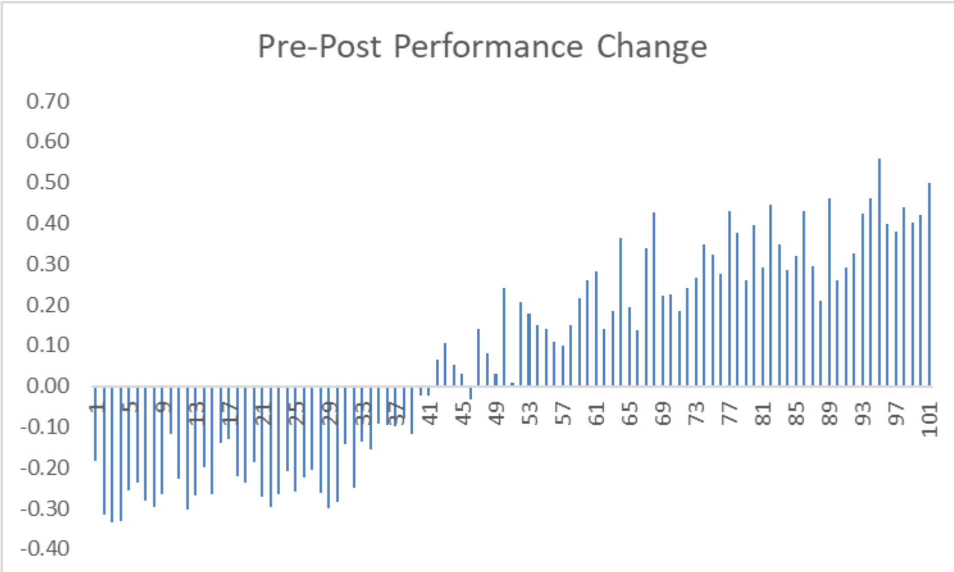


Figure 51. Goal level and policy value change predicting behavioral transfer in Model 3B-2 (replication level).

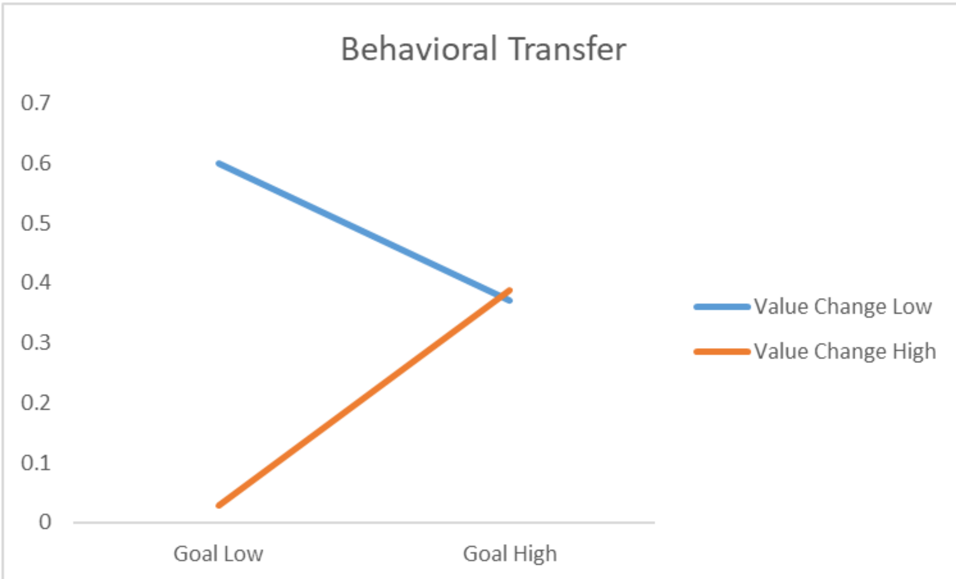


Figure 52. Goal level and policy value change predicting post training performance in Model 3B-2 (replication level).

Figure 53. Goal level and policy value change predicting pre-post performance change in Model 3B-2 (condition level).

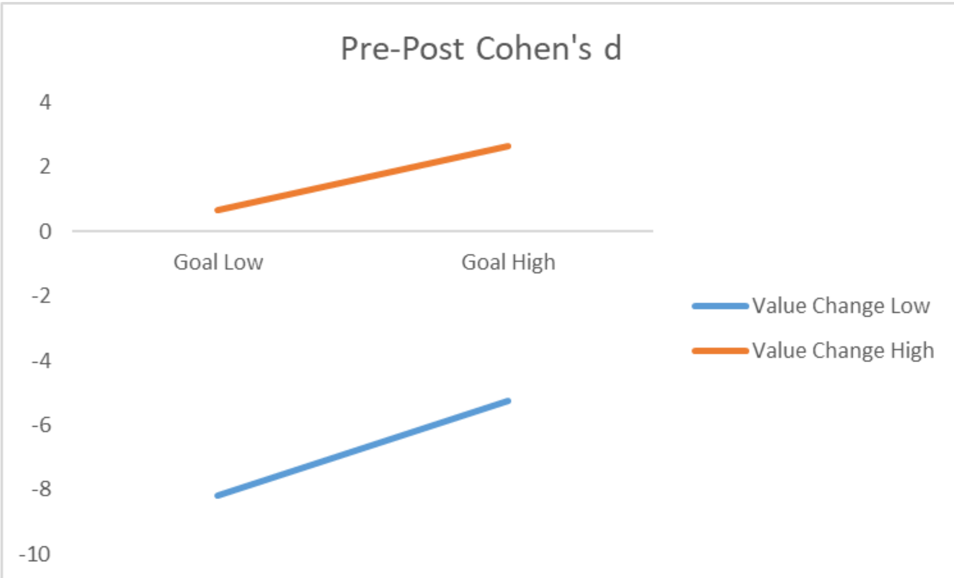


Figure 54. Heat map of goal level and policy value change predicting behavioral transfer in Model 3B-2.

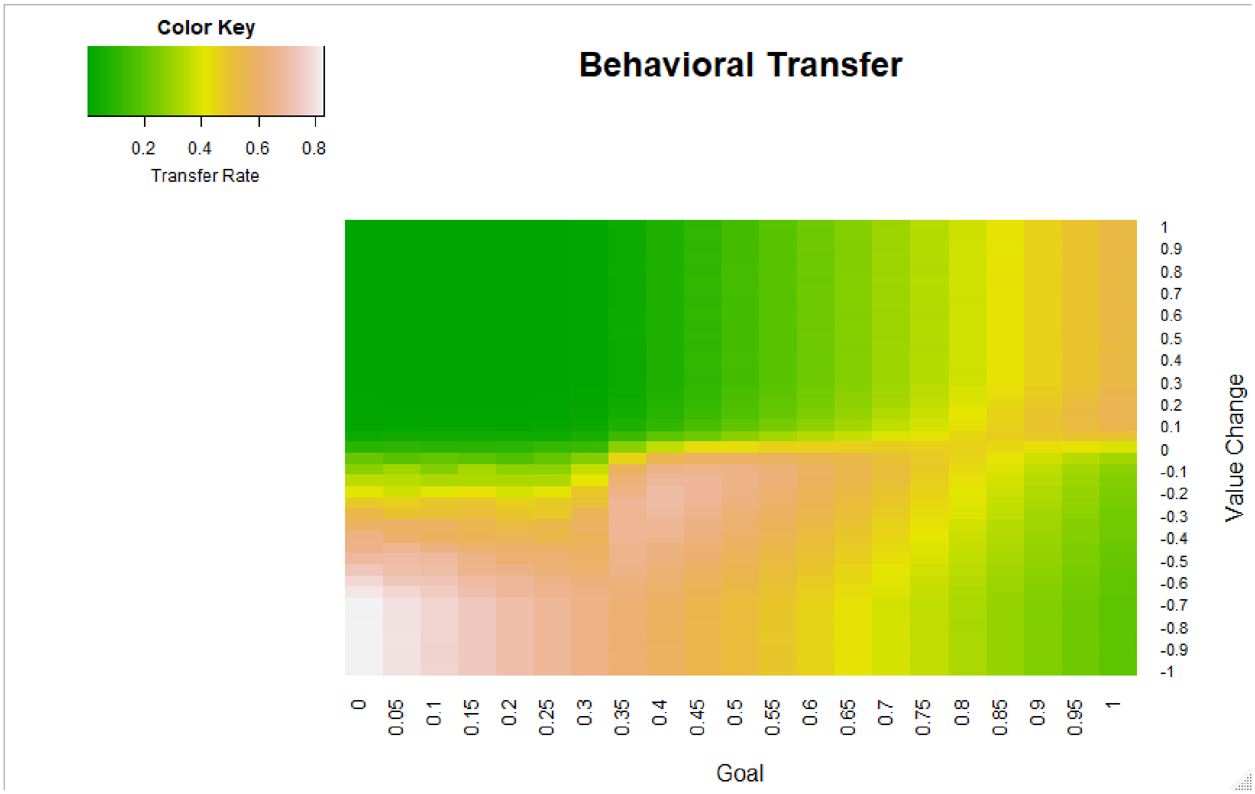


Figure 55. Heat map of goal level and policy value change predicting post training performance in Model 3B-2.

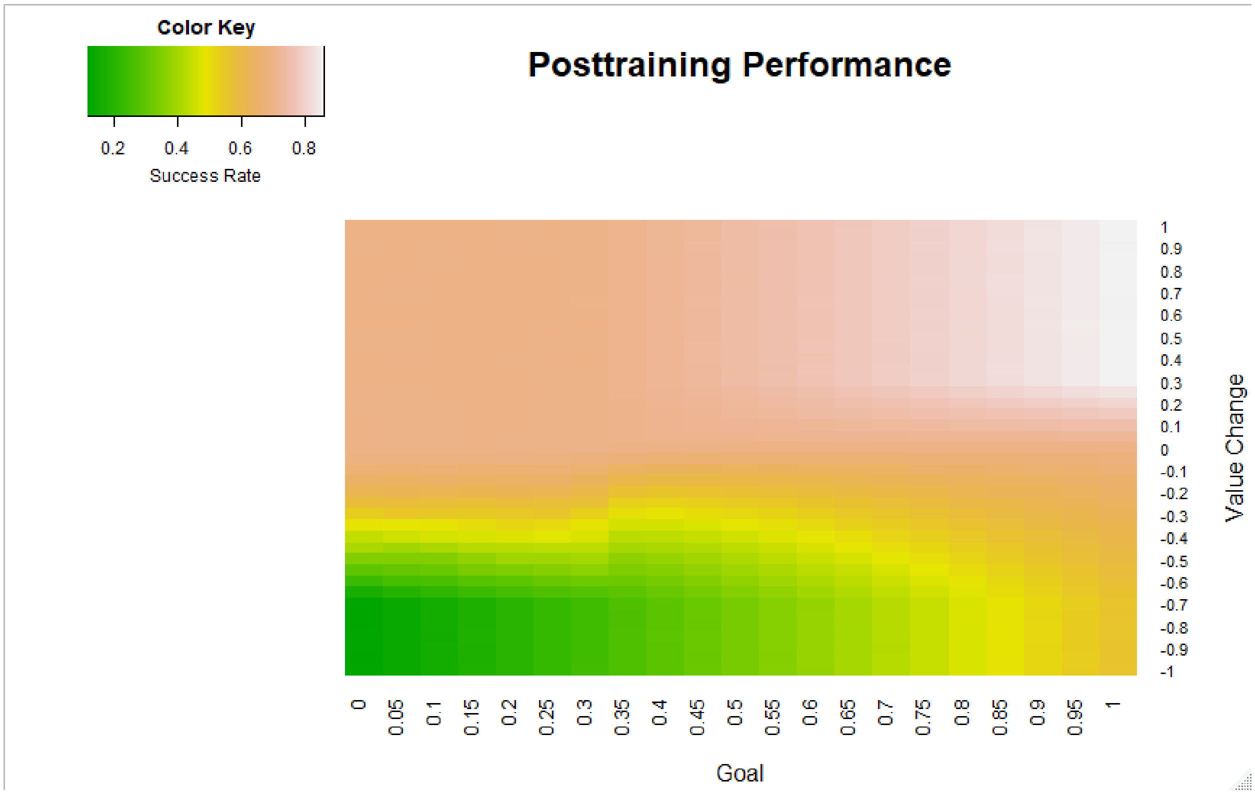


Figure 56. Heat map of goal level and policy value change predicting pre-post performance change in Model 3B-2.

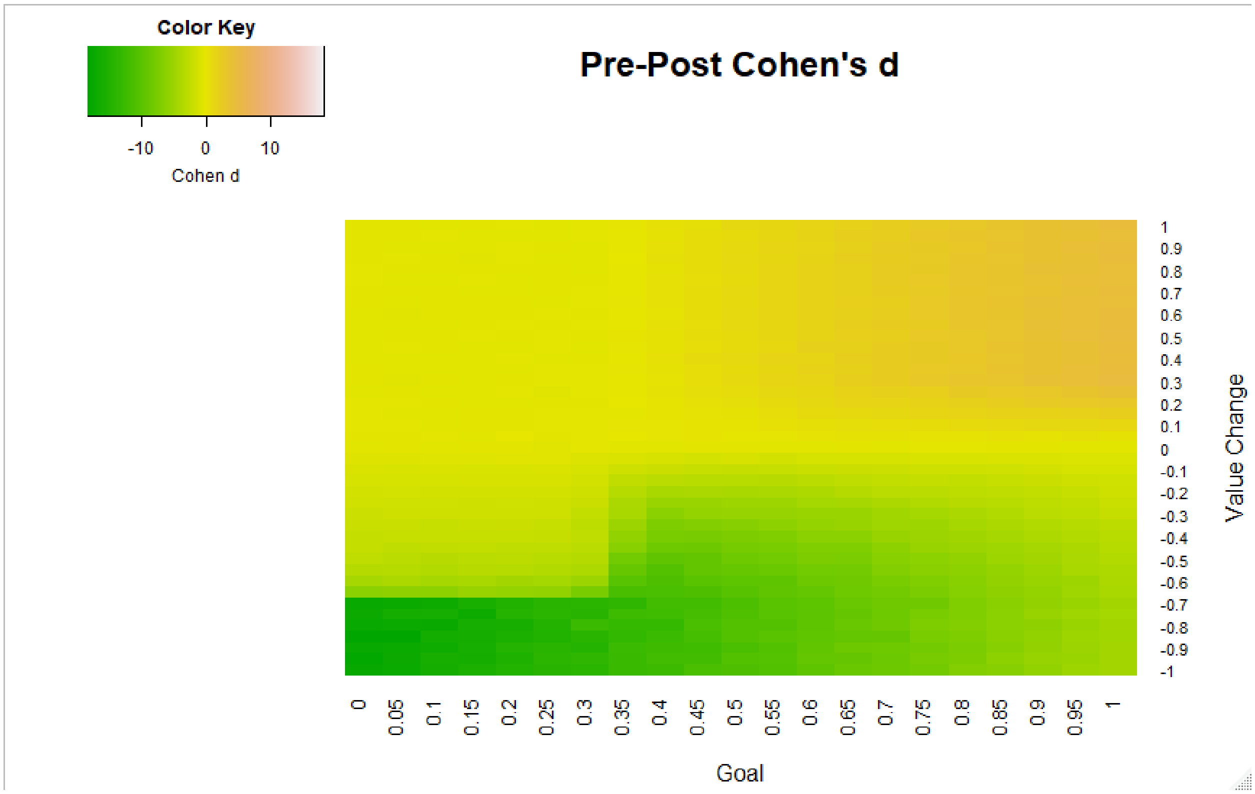


Figure 57. Observed and predicted behavioral transfer from threshold level in Model 3C.

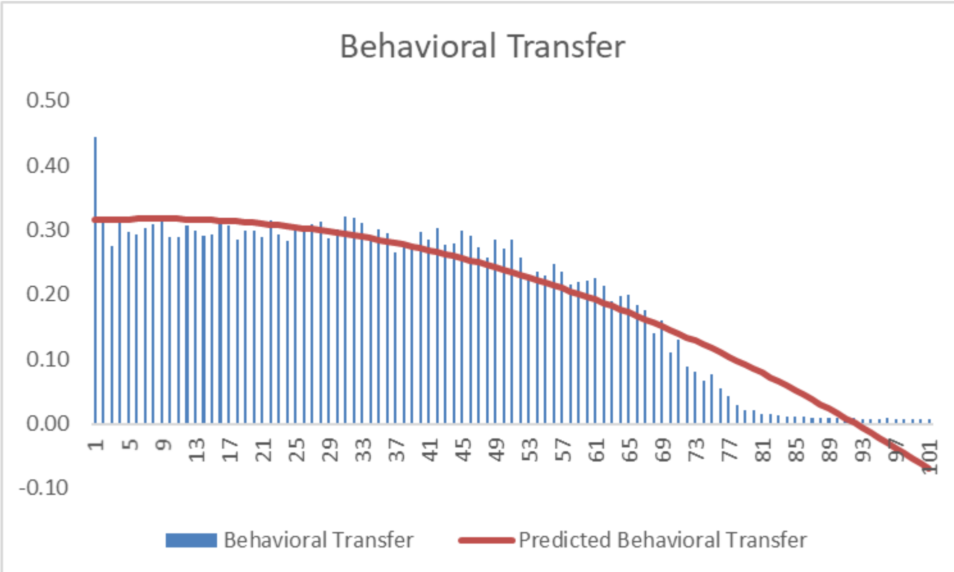
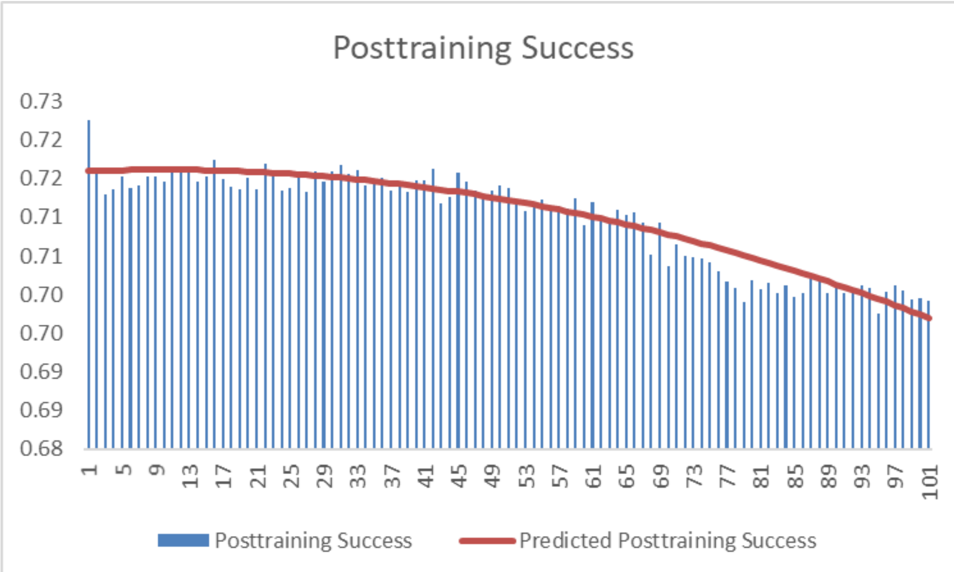


Figure 58. Observed and predicted post training performance from threshold level in Model 3C.



Note scale intentionally not starting at 0 to show shift in percentages more clearly.

Figure 59. Observed and predicted pre-post performance change from threshold level in Model 3C.

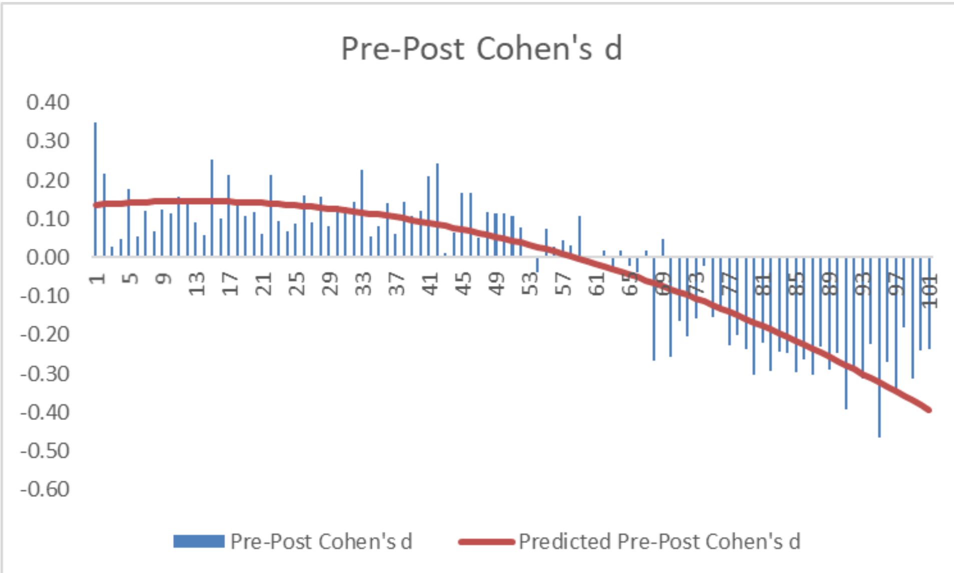


Figure 60. Three-way interaction of engagement thresholds, implementation intentions, and value change predicting behavioral transfer in Experiment 4A (replication level).

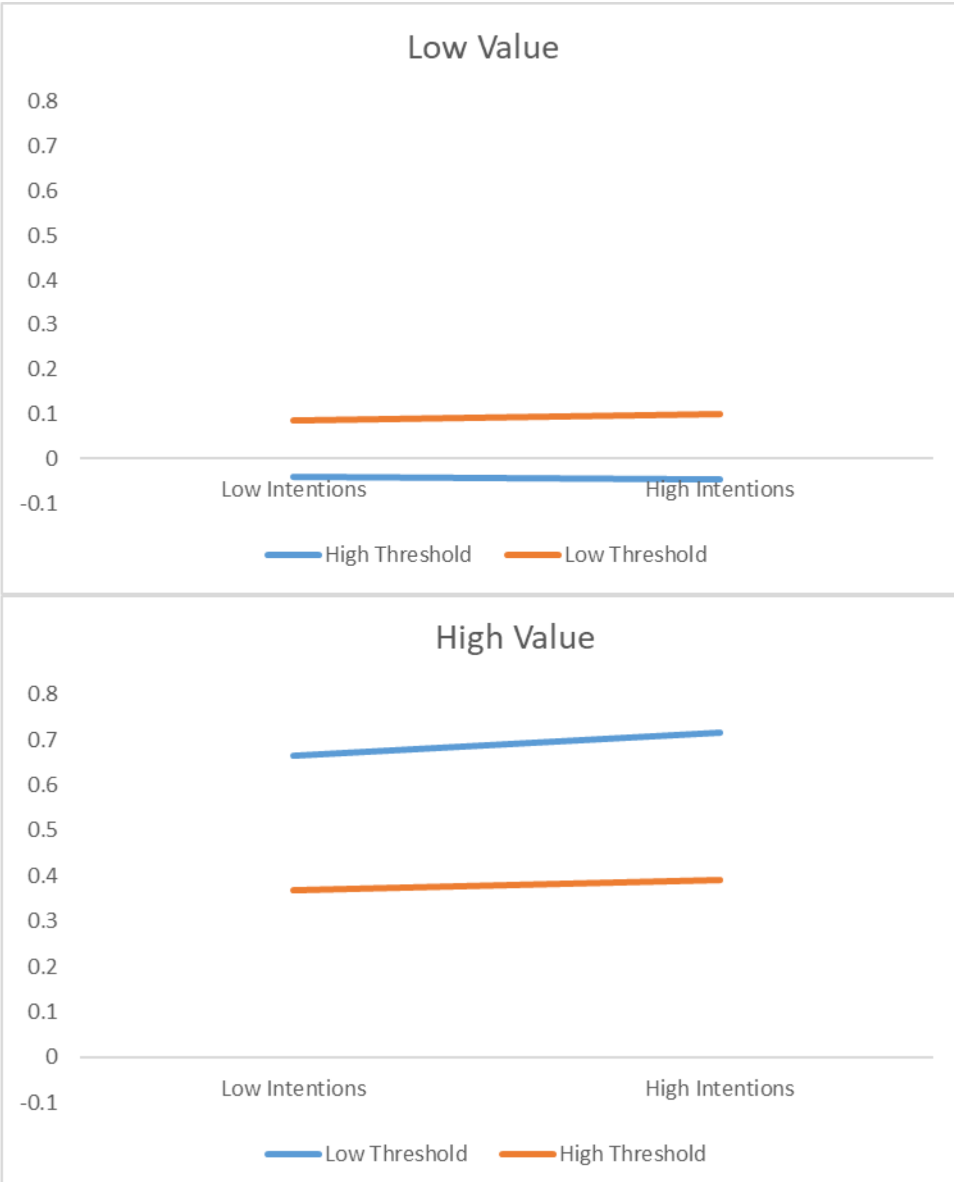
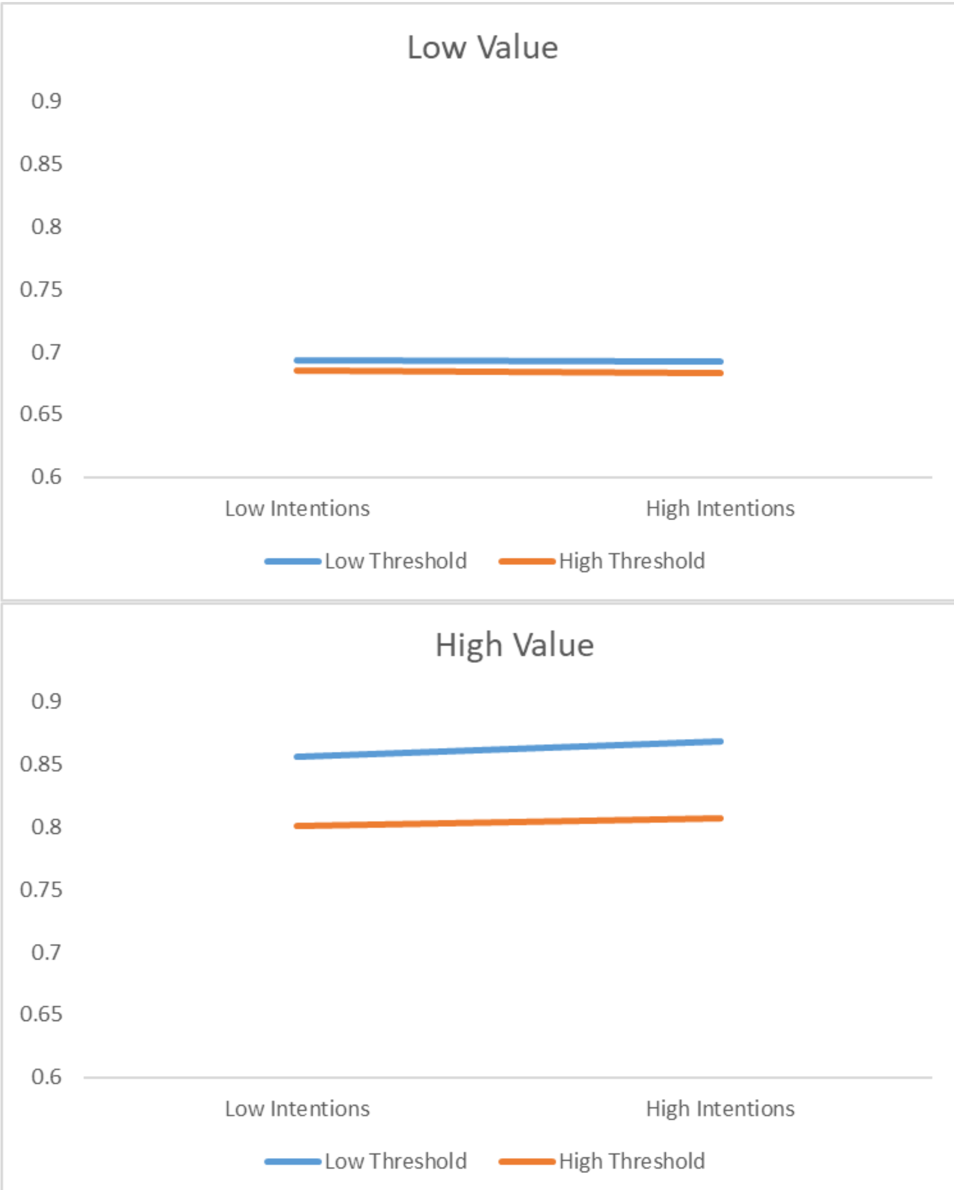


Figure 61. Three-way interaction of engagement thresholds, implementation intentions, and value change predicting post training performance in Experiment 4A (replication level).



Note: Y axis does not start at 0 to better highlight effect

Figure 62. Three-way interaction of engagement thresholds, implementation intentions, and value change predicting pre-post training performance change in Experiment 4A (condition level).

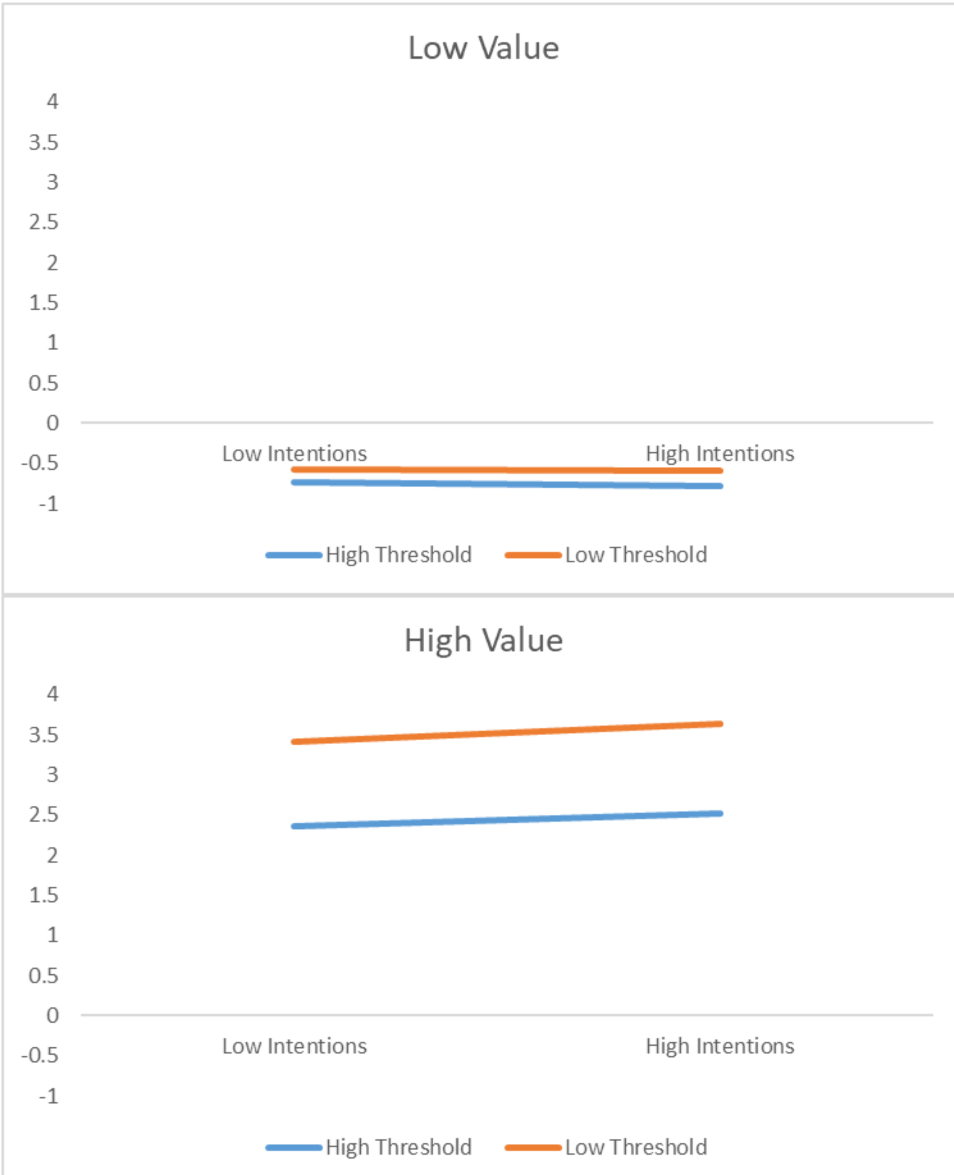


Figure 63. Heat map of three-way interaction of engagement thresholds, implementation intentions, and value change predicting behavioral transfer in Experiment 4A (replication level).

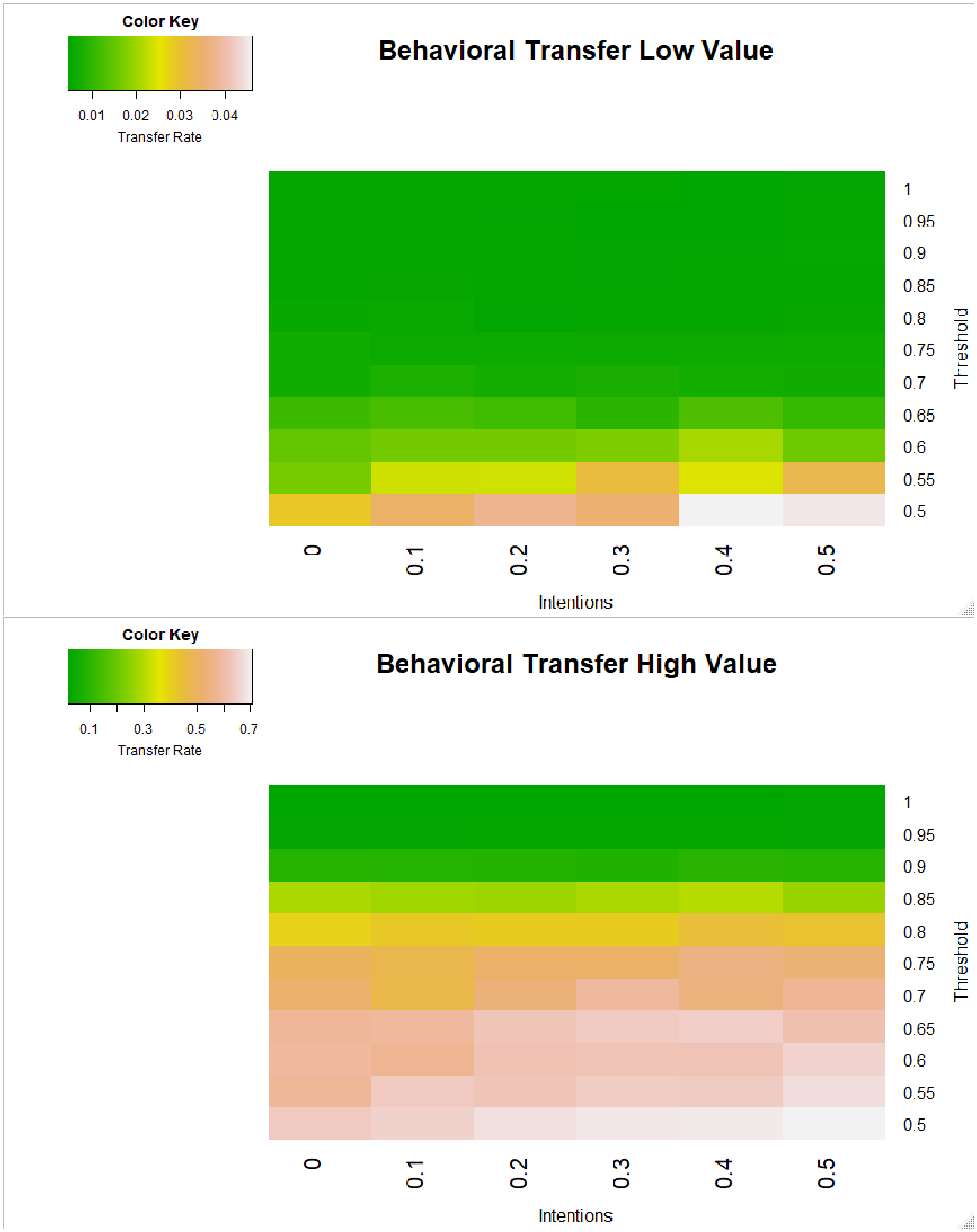


Figure 64. Heat map of three-way interaction of engagement thresholds, implementation intentions, and value change predicting post training performance in Experiment 4A (replication level).

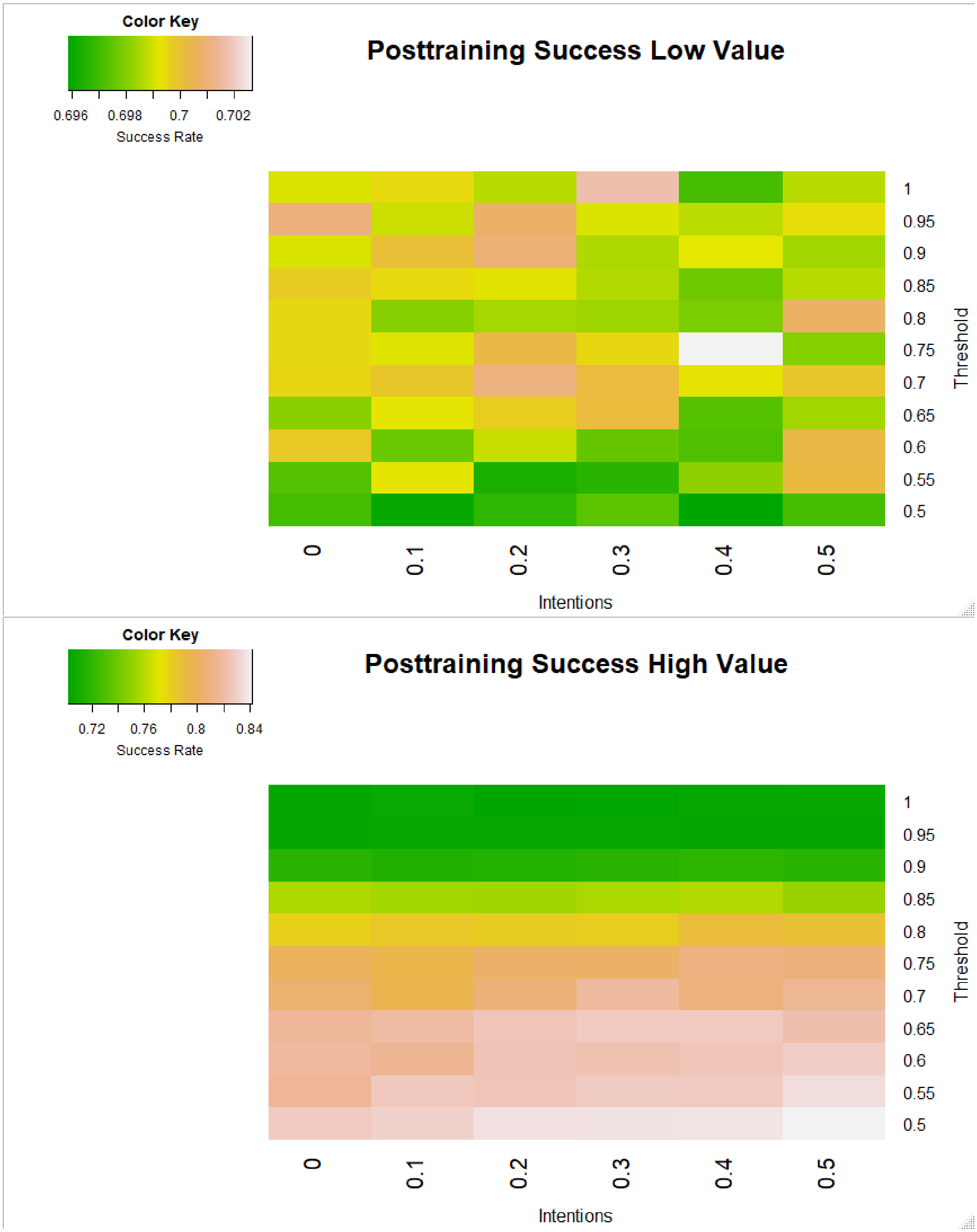


Figure 65. Heat map of three-way interaction of engagement thresholds, implementation intentions, and value change predicting pre-post training performance change in Experiment 4A (condition level).

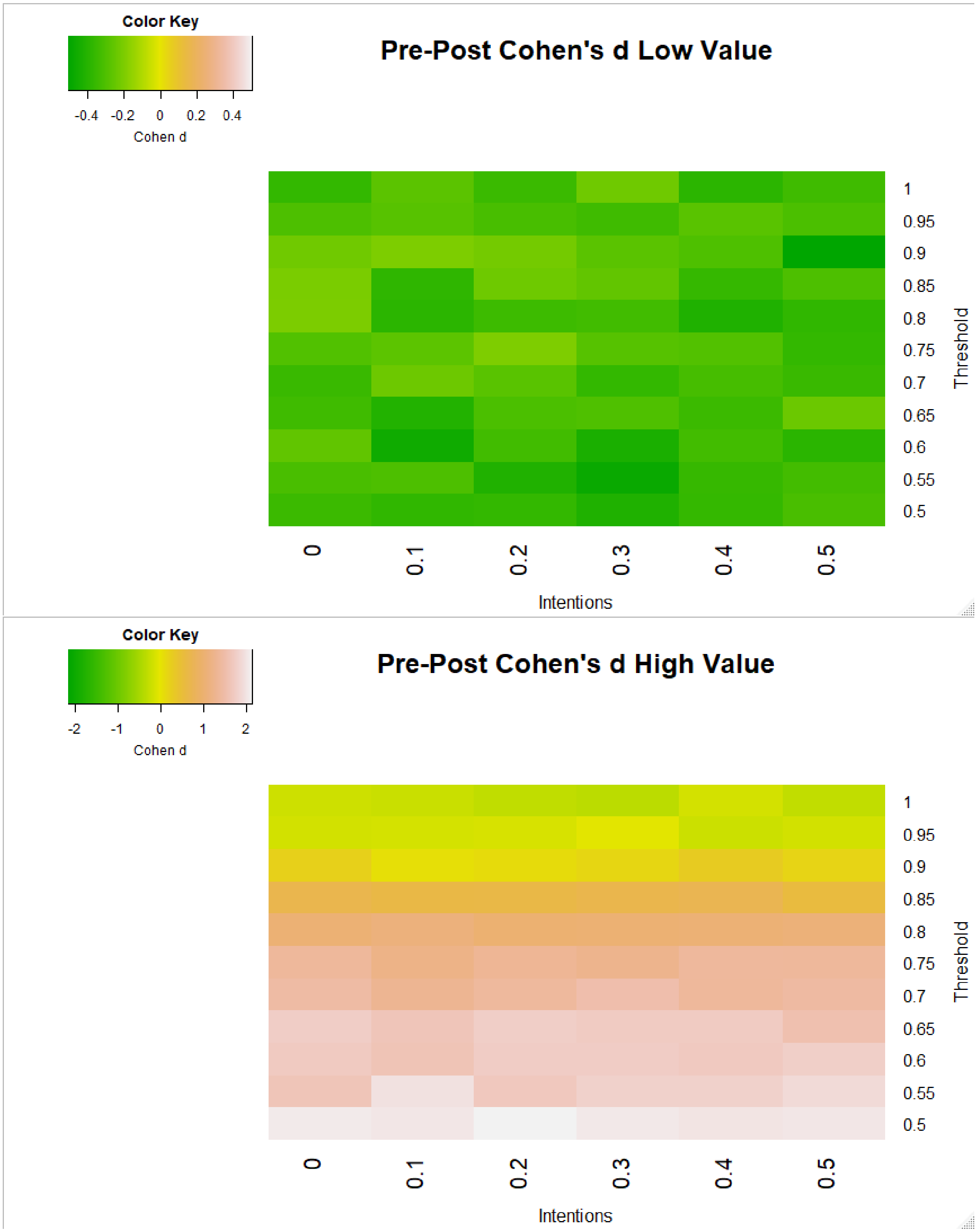


Figure 66. Three-way interaction of number of trainees, conformity, and goals predicting behavioral transfer in Experiment 4B (replication level).

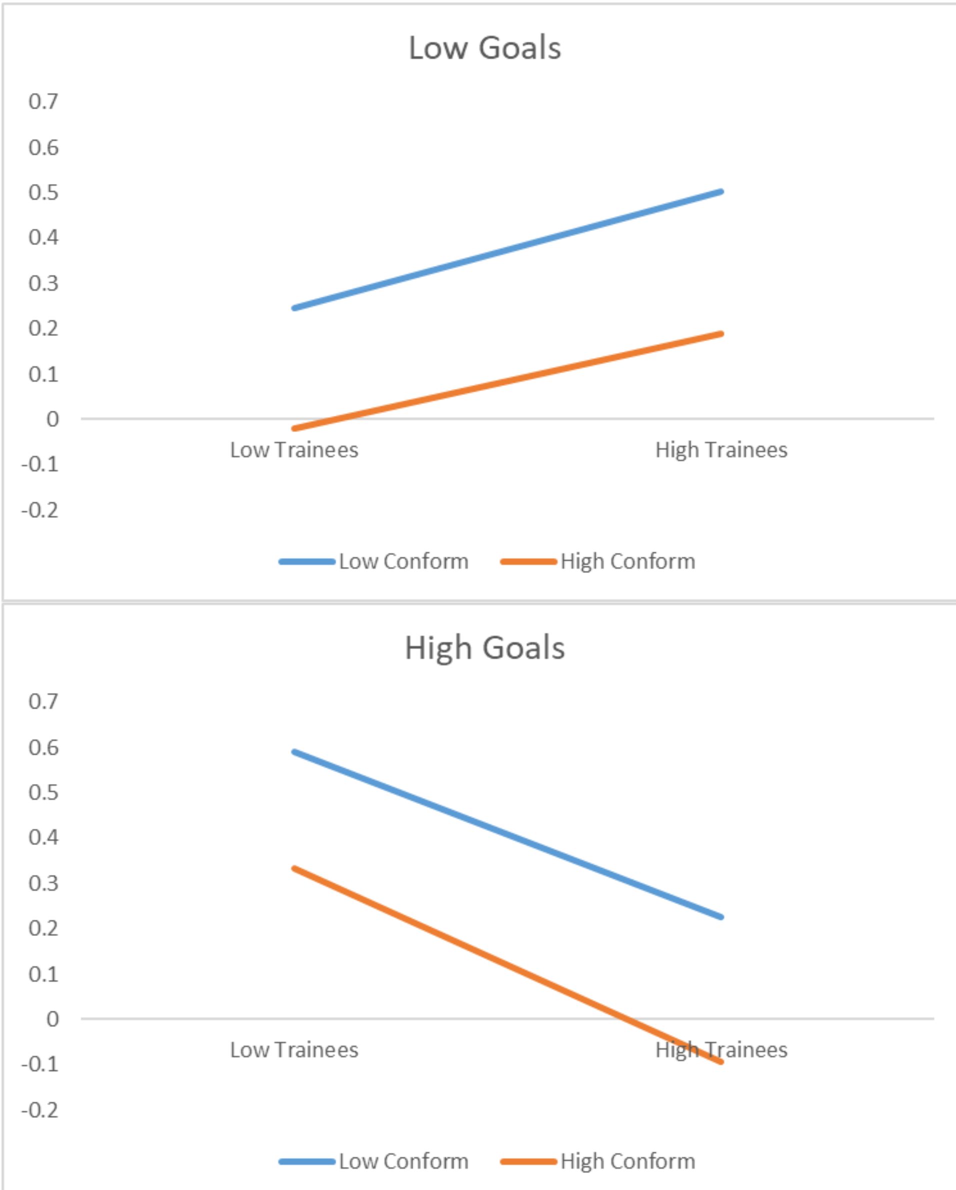
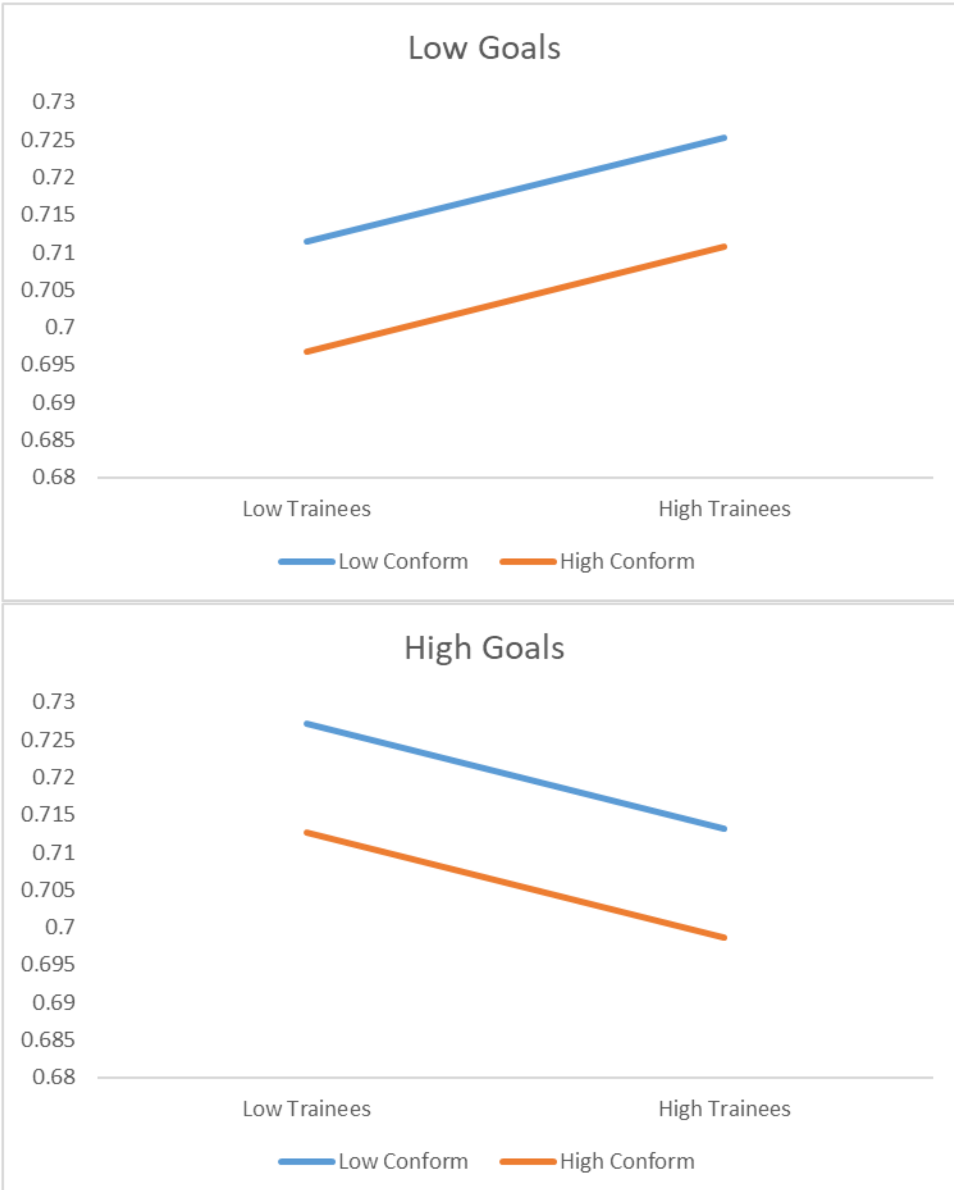
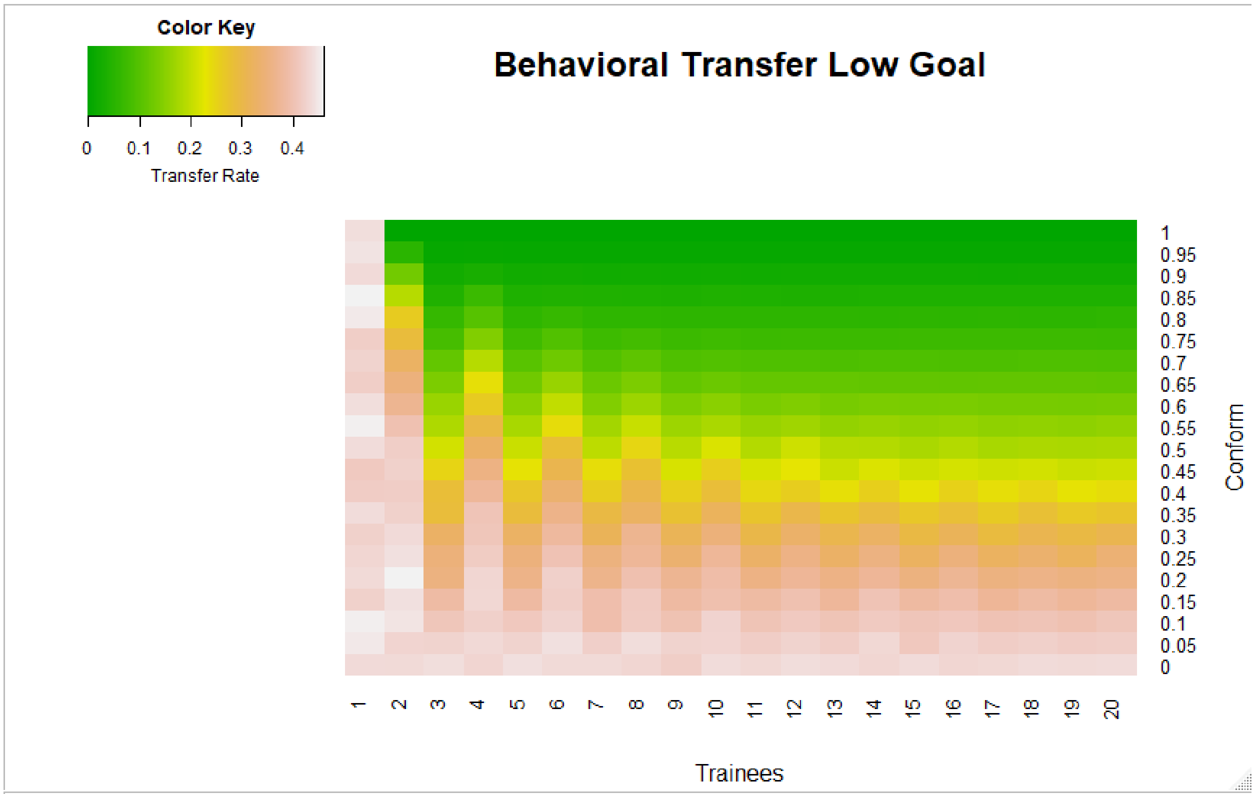


Figure 67. Three-way interaction of number of trainees, conformity, and goals predicting post training performance in Experiment 4B (replication level).



Note: Y axis does not start at 0 to better illustrate effect.

Figure 68. Heat maps of three-way interaction of number of trainees, conformity, and goals predicting behavioral transfer in Experiment 4B (replication level).



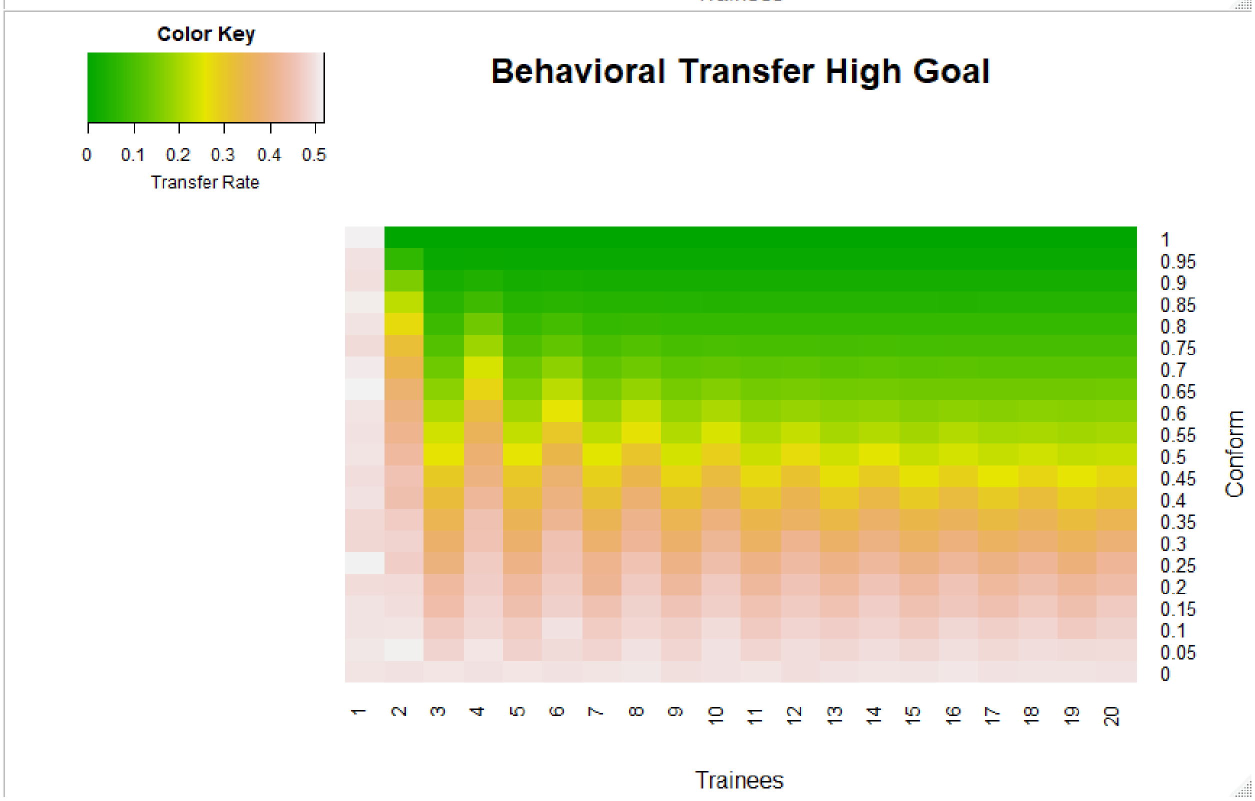


Figure 69. Heat maps of three-way interaction of number of trainees, conformity, and goals predicting post training performance in Experiment 4B (replication level).

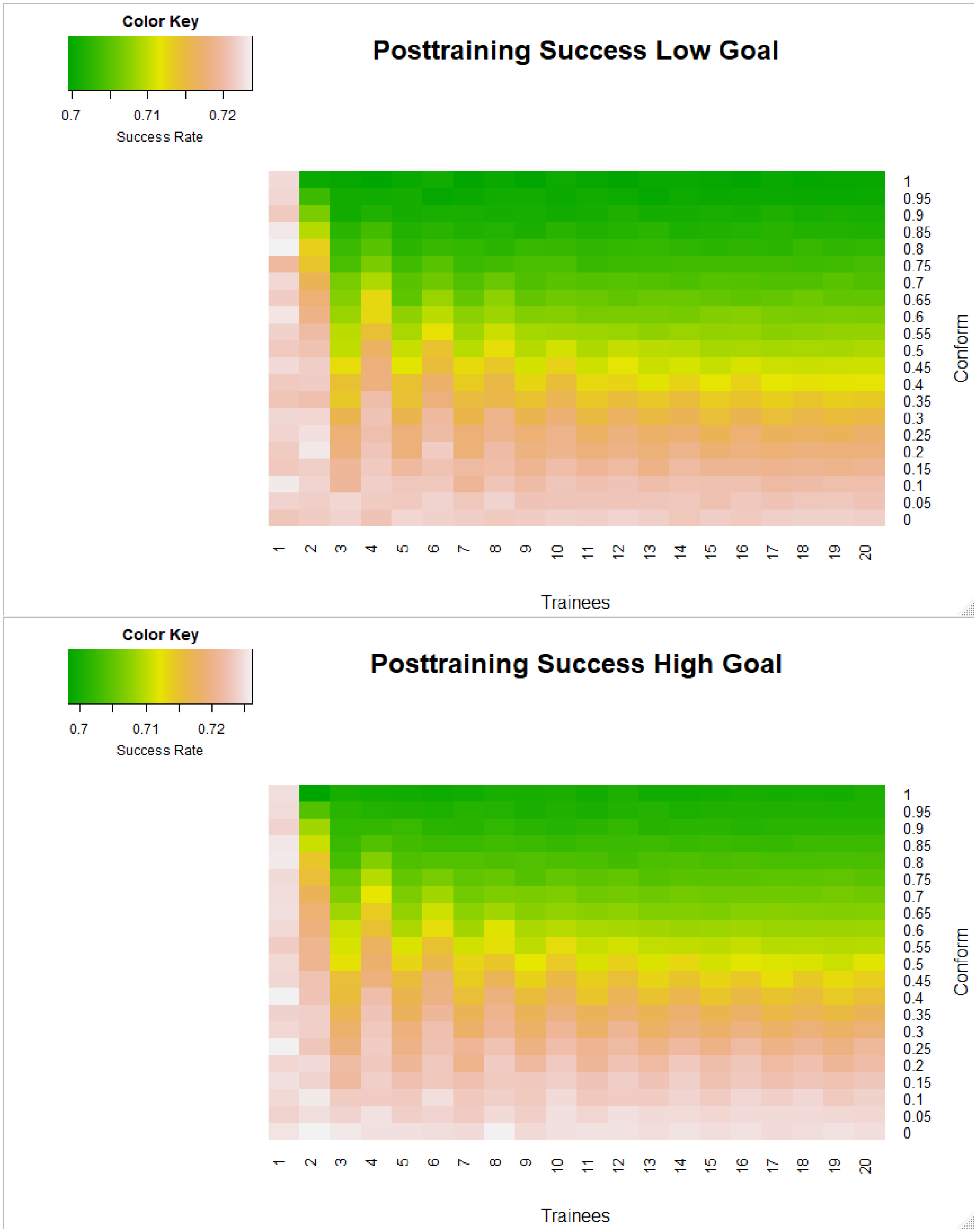


Figure 70. Three-way interaction of conformity, goals, and value change predicting behavioral transfer in Experiment 4C (replication level).

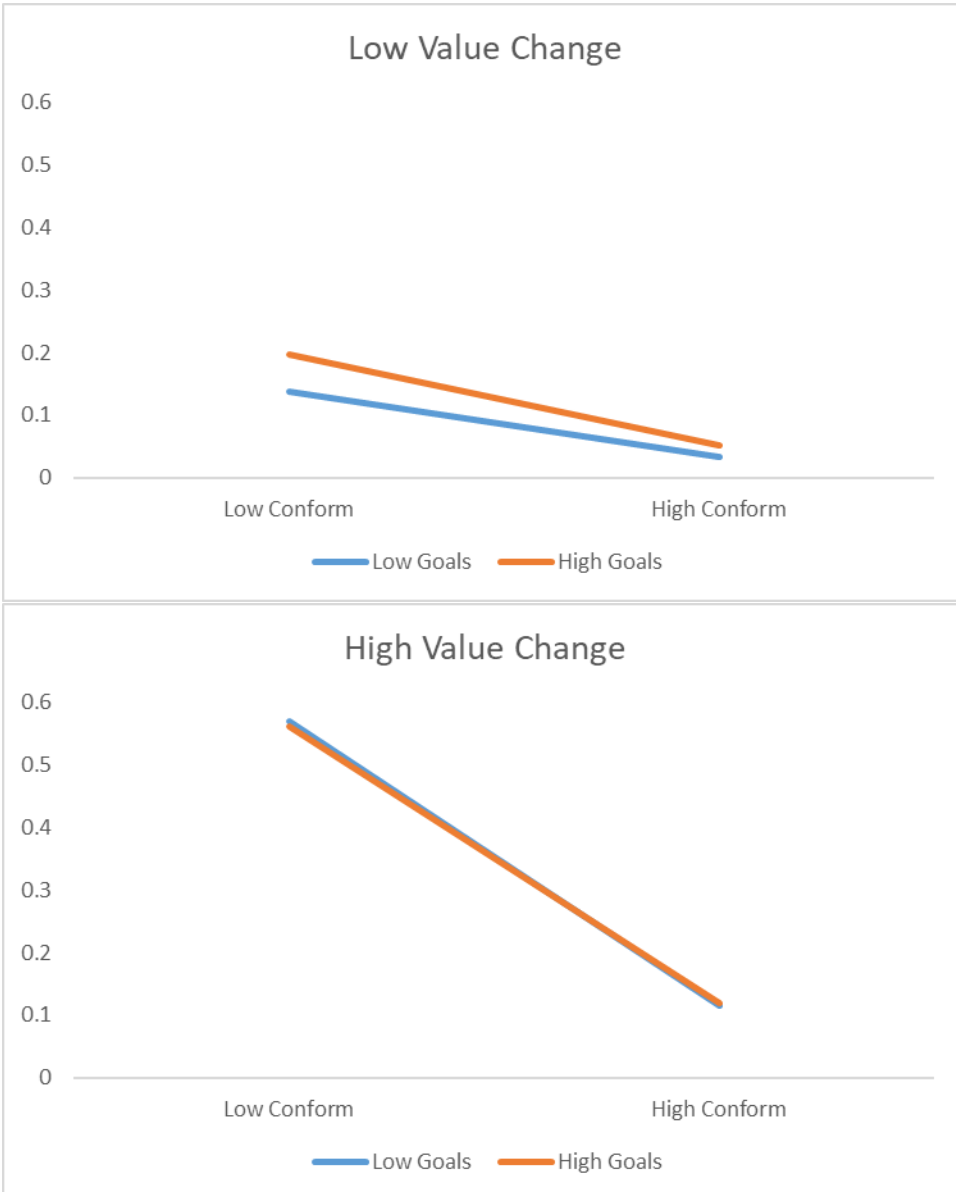


Figure 71. Three-way interaction of conformity, goals, and value change predicting post training performance in Experiment 4C (replication level).

Note: Y axis does not start at 0 to better highlight effect

Figure 72. Three-way interaction of conformity, goals, and value change predicting pre-post training performance change in Experiment 4C (condition level).

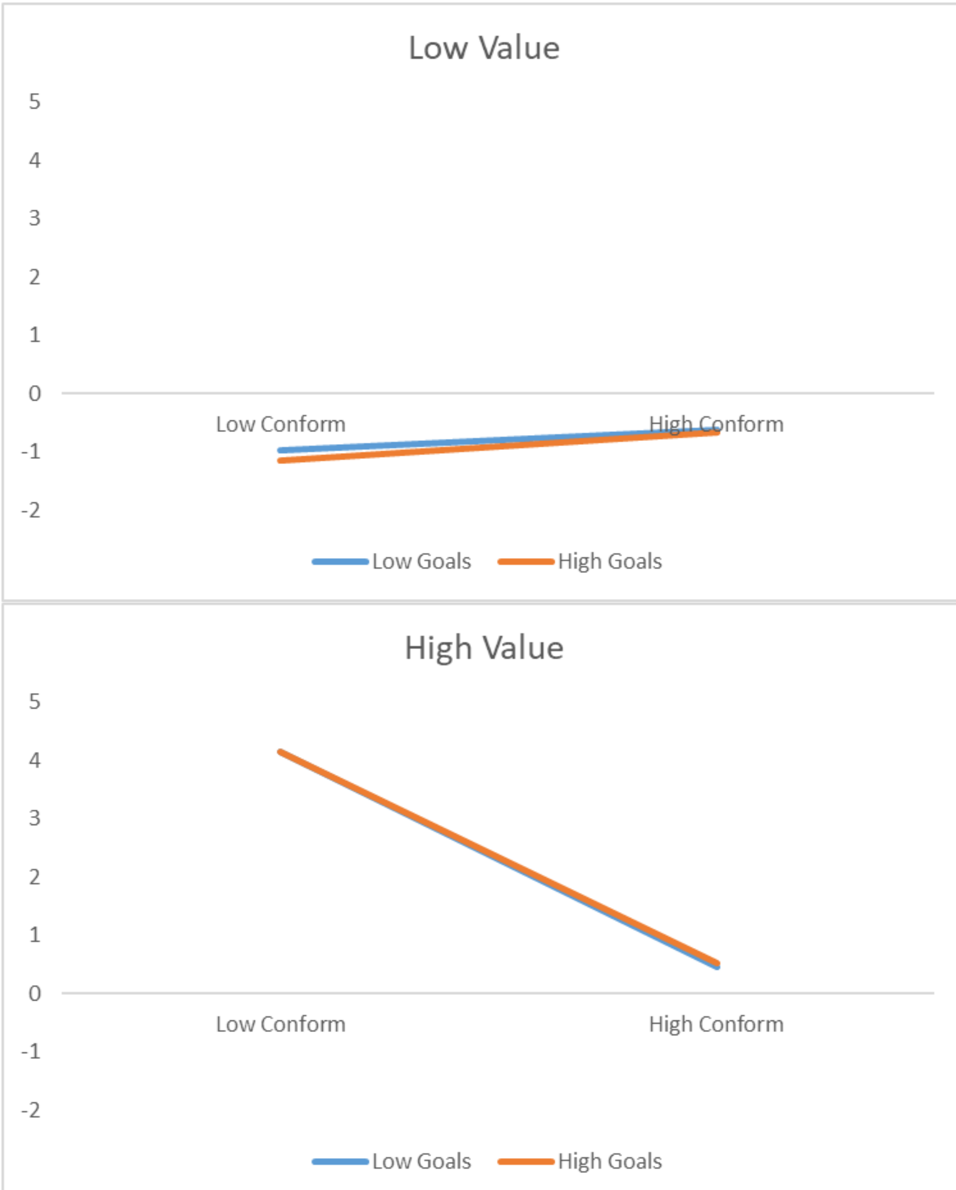


Figure 73. Heat map of three-way interaction of conformity, goals, and value change predicting behavioral transfer in Experiment 4C (replication level).

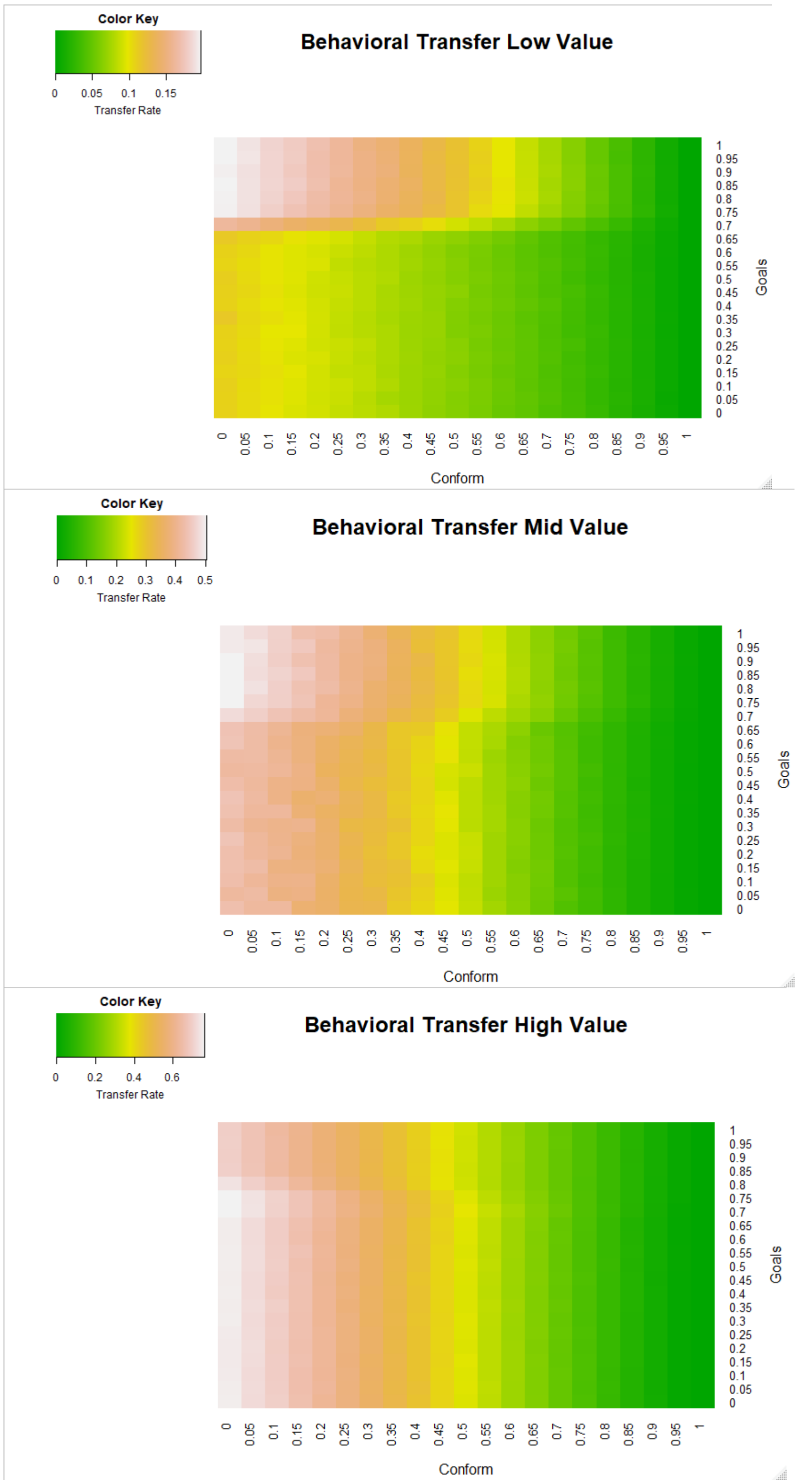


Figure 74. Heat map of three-way interaction of conformity, goals, and value change predicting post training performance in Experiment 4C (replication level).

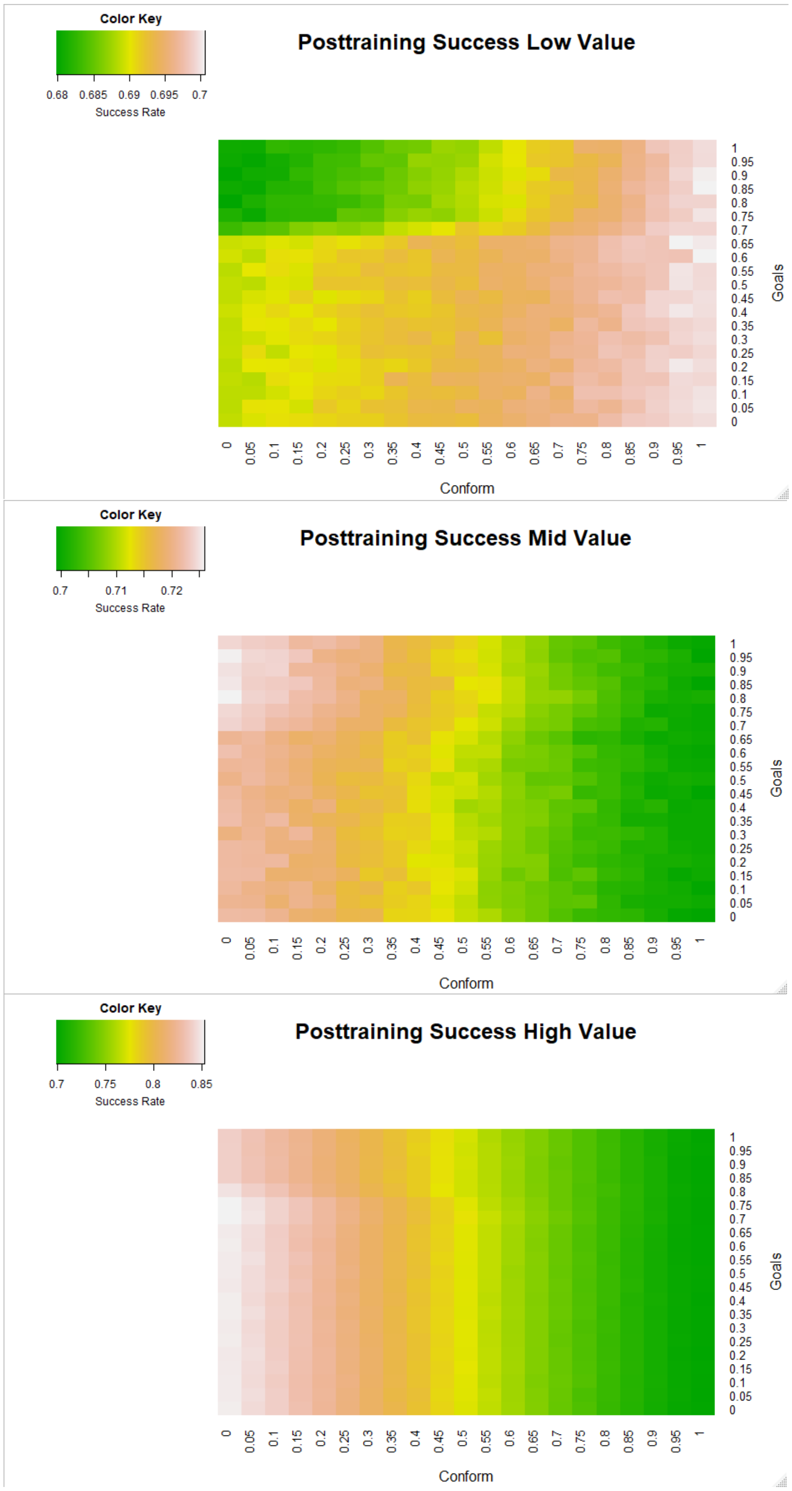


Figure 75. Heat map of three-way interaction of conformity, goals, and value change predicting pre-post training performance change in Experiment 4C (condition level).

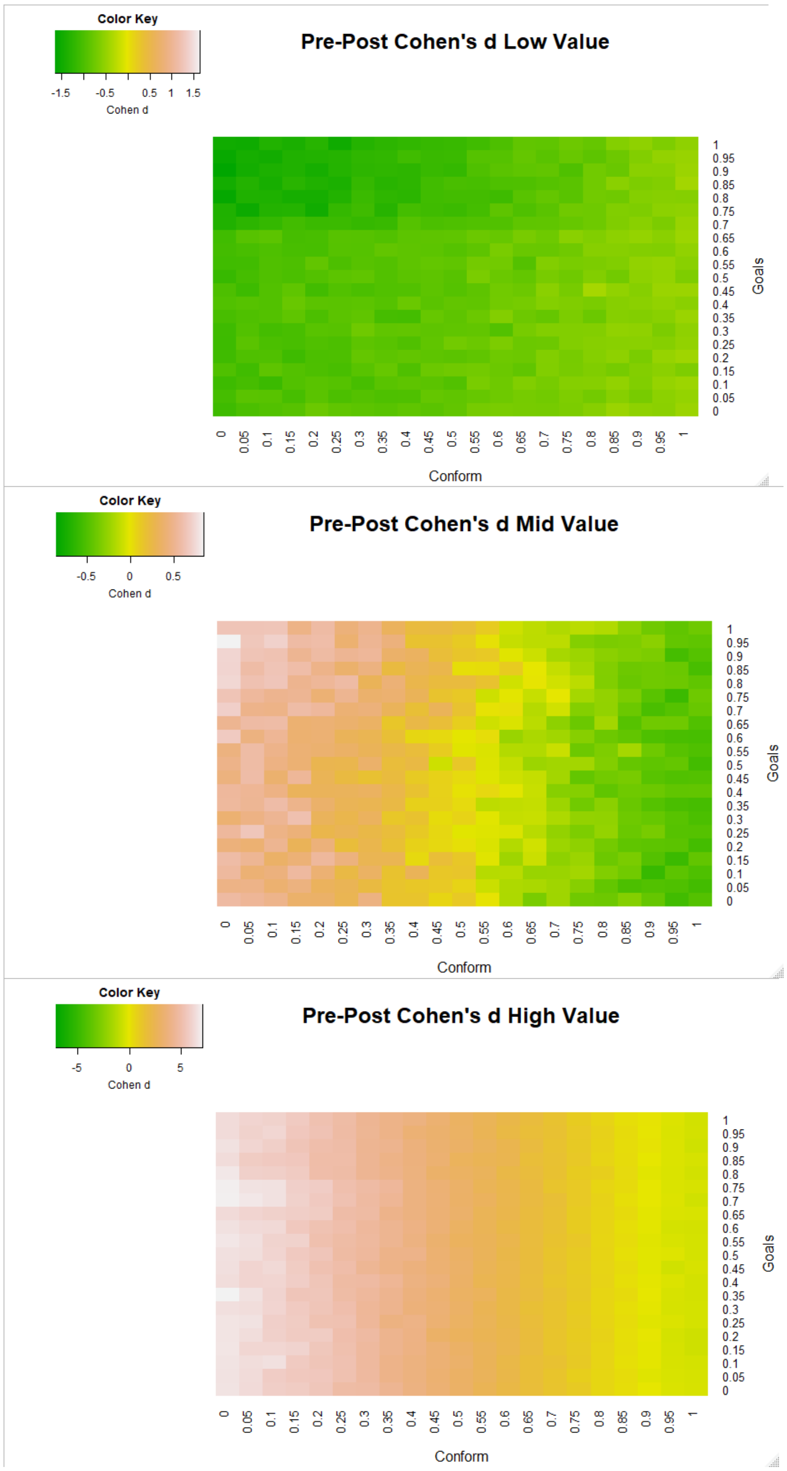


Figure 76. Three-way interaction of type 2 likelihood, conformity, and goals predicting behavioral transfer in Experiment 4D (replication level).

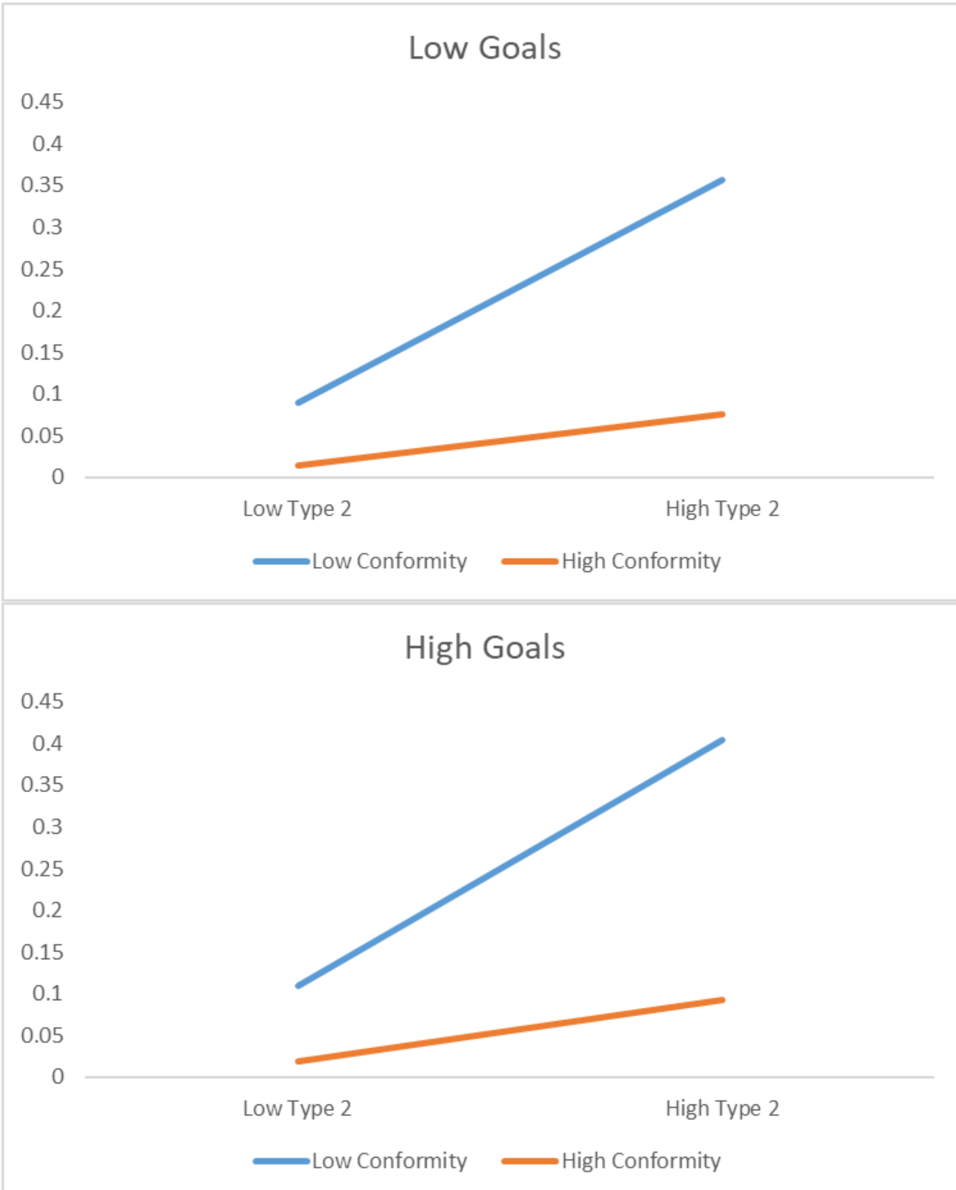
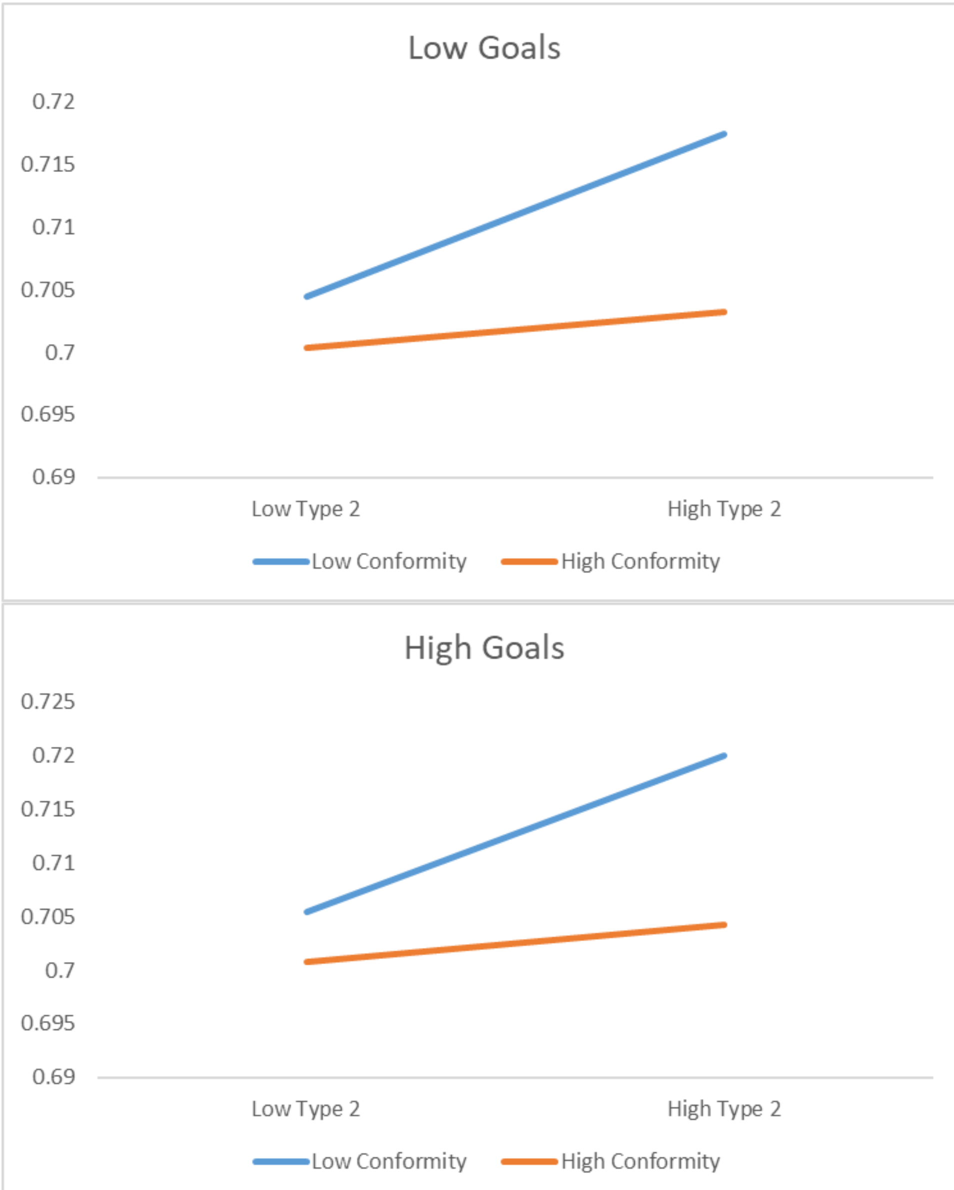


Figure 77. Three-way interaction of type 2 likelihood, conformity, and goals predicting post training performance in Experiment 4D (replication level).



Note: Y axis does not start at 0 to better highlight effect

Figure 78. Three-way interaction of type 2 likelihood, conformity, and goals predicting pre-post training performance change in Experiment 4D (condition level).

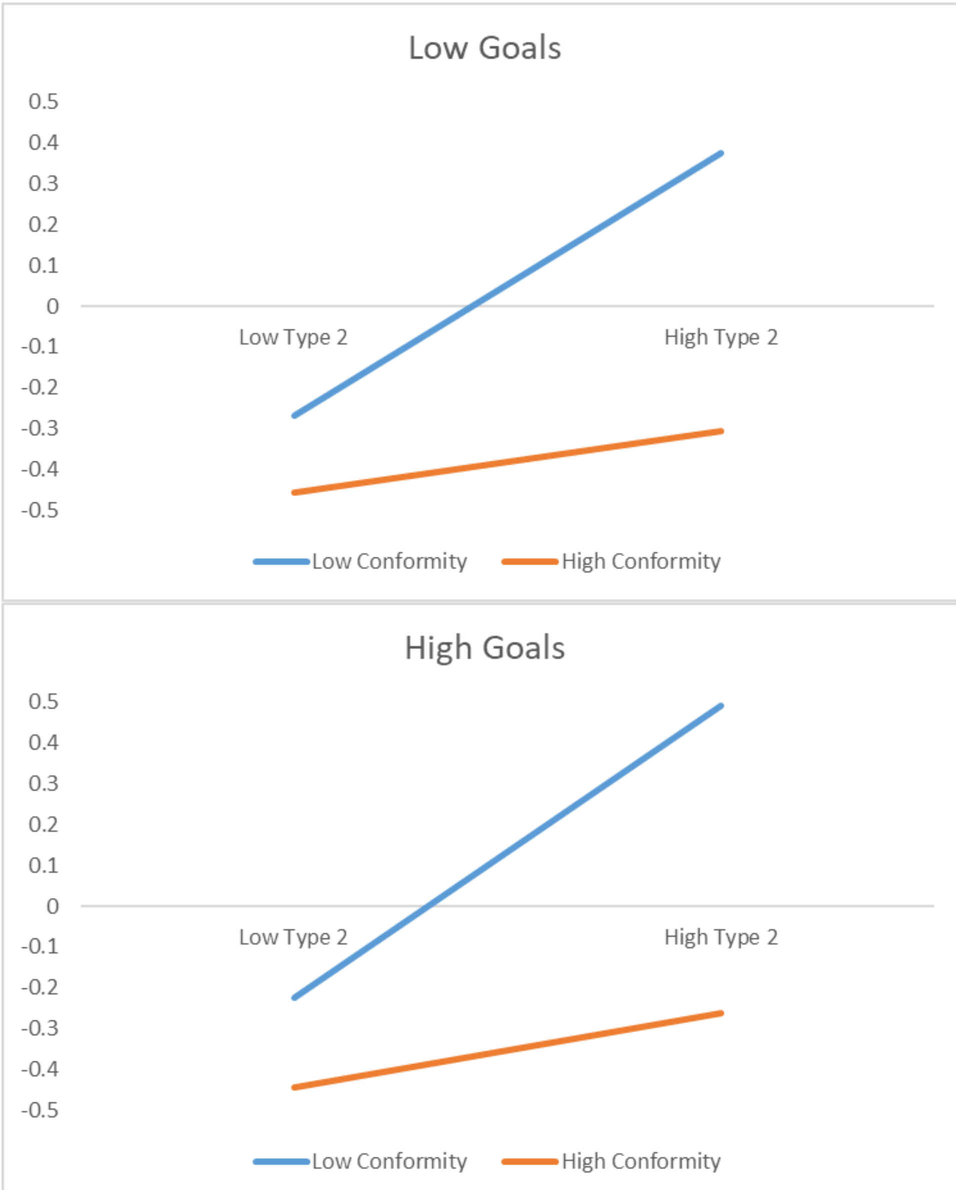
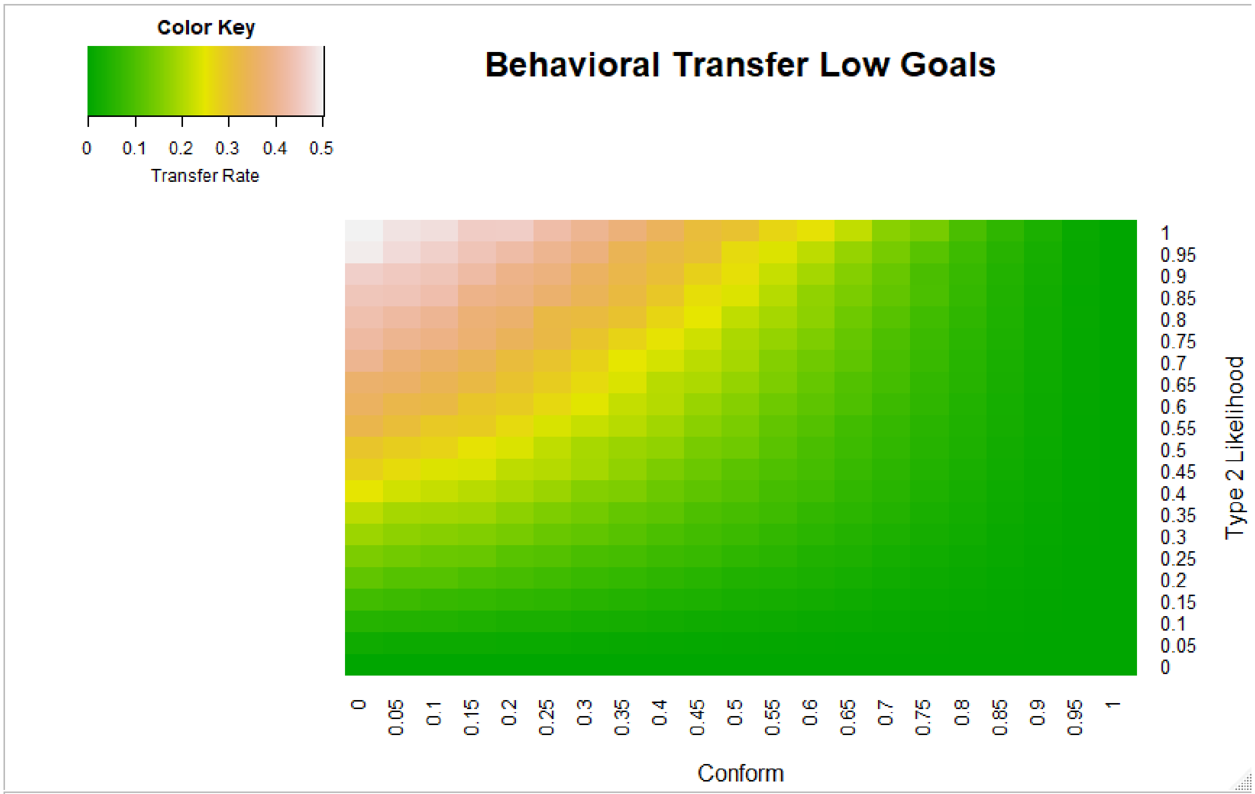


Figure 79. Heat map of three-way interaction type 2 likelihood, conformity, and goals predicting behavioral transfer in Experiment 4D (replication level).



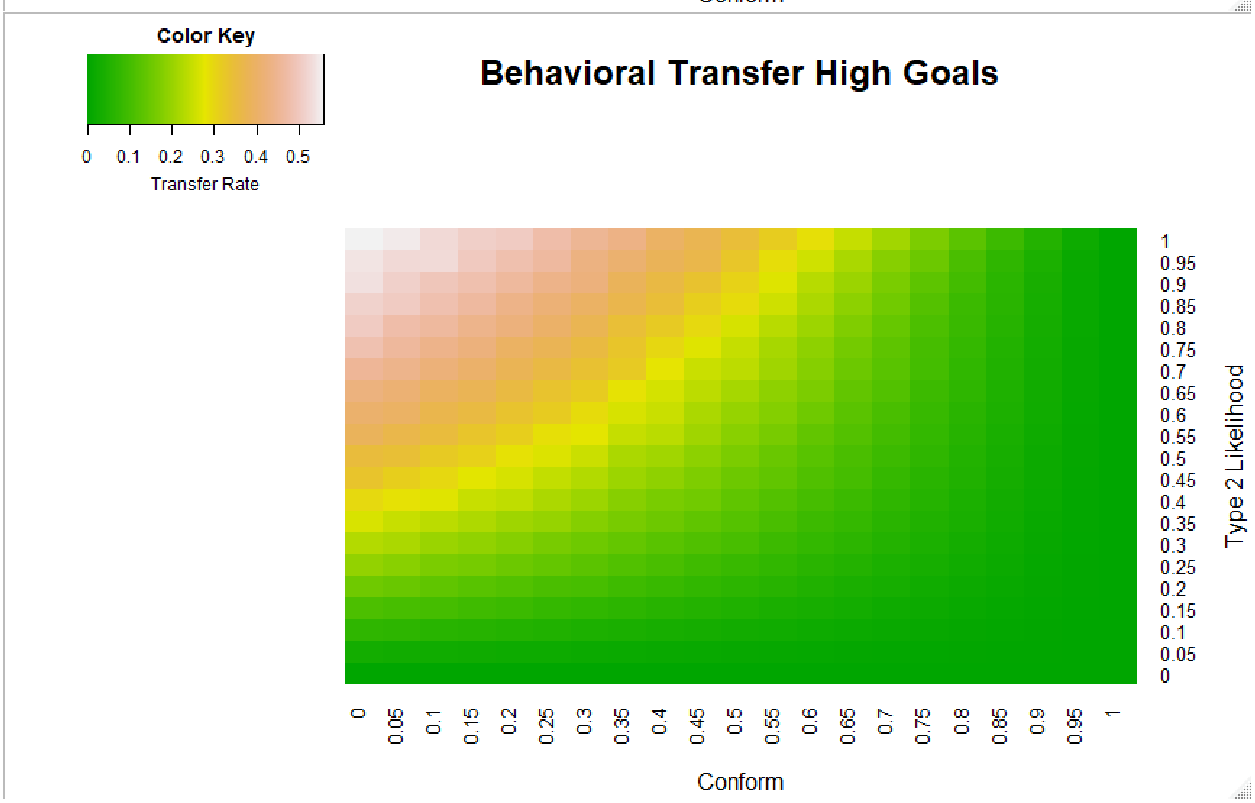


Figure 80. Heat map of three-way interaction of type 2 likelihood, conformity, and goals predicting post training performance in Experiment 4D (replication level).

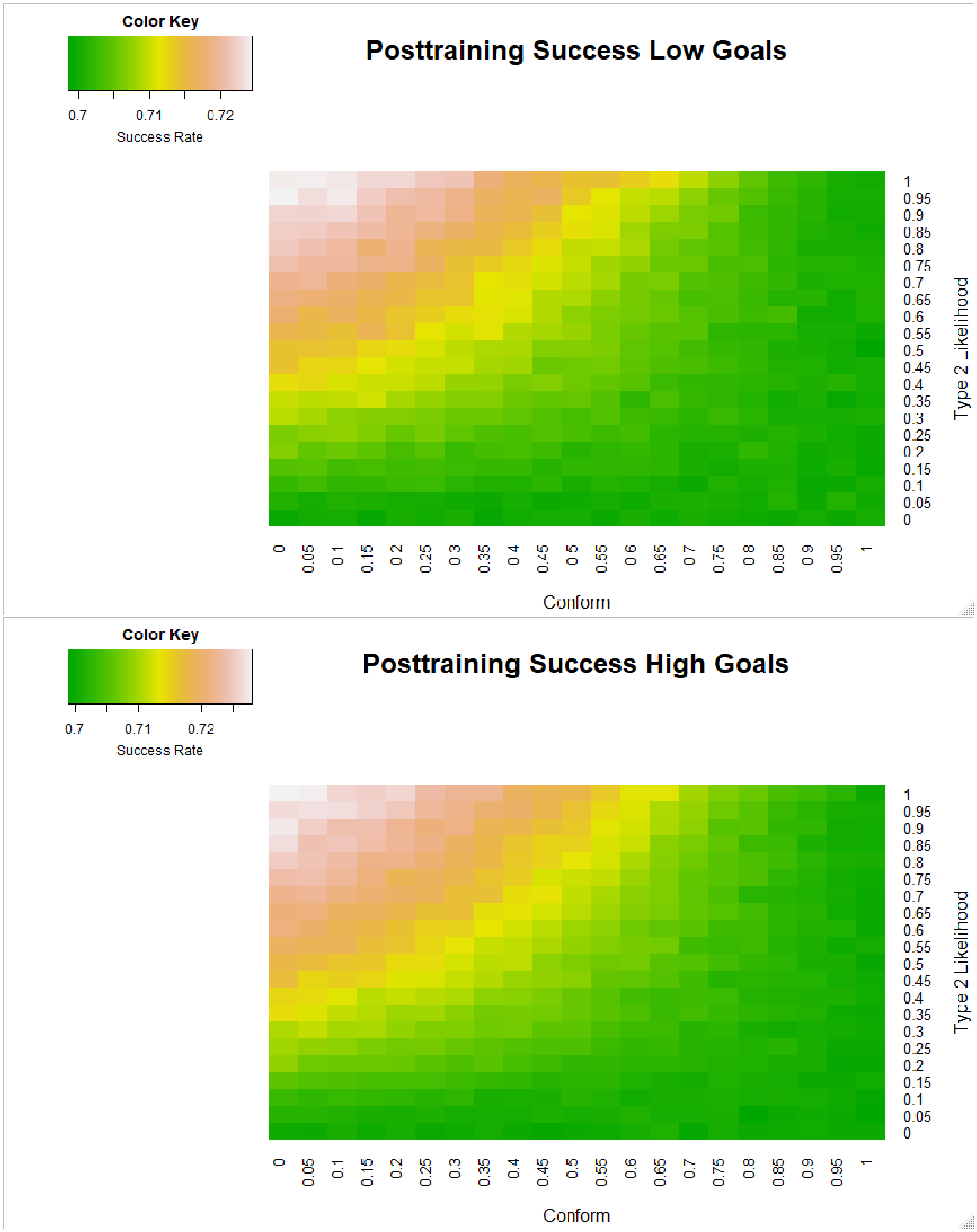
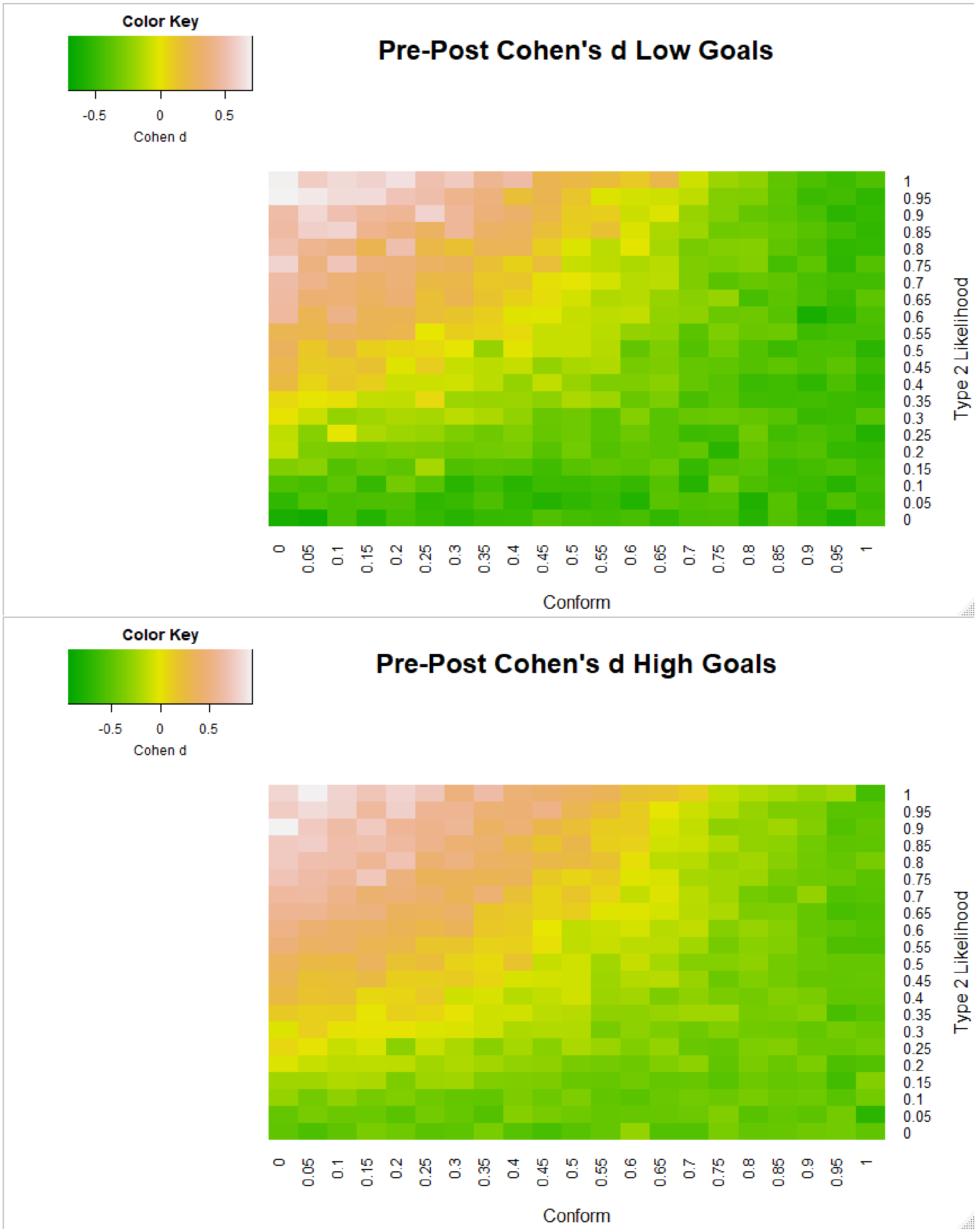


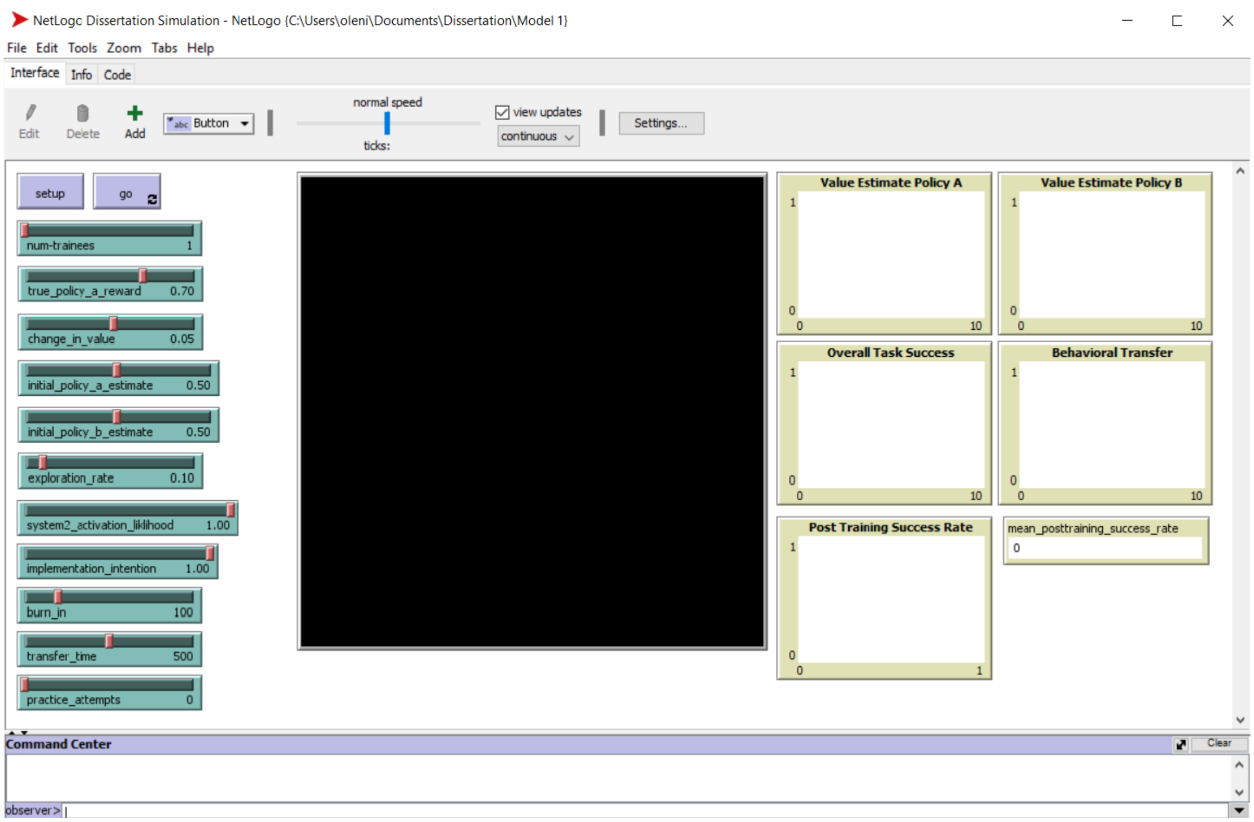
Figure 81. Heat map of three-way interaction of type 2 likelihood, conformity, and goals predicting pre-post training performance change in Experiment 4D (condition level).



APPENDICES

Appendix A: Study 1 Environment and Code

Figure 82. Snapshot of the modeling environment for Study 1 in NetLogo.



Algorithm 13. NetLogo Code for Study 1 Model

```
breed [trainees trainee] ;types of agents allowed in environment

  trainees-own [
    value_estimate_a ;estimated value of Policy A
    value_estimate_b ;estimated value of Policy B
    system1_choose_a ;liklihood of choosing Policy A as habitual response
    attempts_policy_a ;number times applied Policy A
    attempts_policy_b ;number time applied Policy B
    reward_a ;reward received on most recent attempt with Policy A
    reward_b ;reward received on most recent attempt with Policy B
    task_successes ;number of times successful at task overall
    post_training_successes ;number of times successful only post-training
    pretraining_success_rate ;success rate pretraining only
    posttraining_success_rate ; percentage of times successful in post-training environment
    behavioral_transfer_rate ;rate of choosing Policy B in transfer environment
    transfer_time_count ; ticks into transfer time
    ]

  globals [
    mean_value_estimate_a ;mean of agent value estimates for Policy A
    mean_value_estimate_b ;mean of agent value estimates for Policy B
    mean_overall_task_success ;task rate of success for full simulation
    mean_pretraining_success_rate ;success rate pretraining only all agents
    mean_posttraining_success_rate ;success rate posttraining only all agents
    mean_behavioral_transfer_rate ;rate of choosing Policy B in transfer environment all
agents
    true_policy_b_reward ;reward for Policy B after adjusting for policy value change
  ]

  to setup
    clear-all ;clears environment from previous simulation
    create-trainees num-trainees [ ;place specified number of agents at center of grid
      set value_estimate_a initial_policy_a_estimate ;set initial value estimate for Policy A
for each trainee
      set value_estimate_b initial_policy_b_estimate ;set initial value estimate for Policy B
for each trainee
      set attempts_policy_a 0 ;number times applied Policy A initial set to 0
      set attempts_policy_b 0 ;number time applied Policy B initial set to 0
      set task_successes 0 ;number of task successes initial set to 0
      set pretraining_success_rate 0 ;success rate pretraining only initial set to 0
      set post_training_successes 0 ;number of successes for post training initial set to 0
      set posttraining_success_rate 0 ;success rate in posttraining environment initial set to 0
      set behavioral_transfer_rate 0 ;percentage of time choosing trained policy initial set to
0
```

```
  ]
  set true_policy_b_reward (true_policy_a_reward + change_in_value)
  if true_policy_b_reward > 1 [set true_policy_b_reward 1]
  if true_policy_b_reward < 0 [set true_policy_b_reward 0]
  reset-ticks ;reset time count to 0
end

to go ;primary subroutines activated
  if ticks = (burn_in + transfer_time) [save-post-training] ;call subroutine to save post
training variables
  if ticks = (burn_in + transfer_time) [stop] ;control length of sim
  tick ;advance time
  if ticks <= burn_in [trainees-burn-in] ;call subroutine to have trainee engage in task
during burn in period
  if ticks > burn_in [trainees-transfer] ;call subroutine for trainee decisions post training
  if ticks = burn_in [save-burn-in] ;call subroutine to save pretraining performance
  update-globals ;call subroutine to calculate all global variables used to track sim
functioning
end

to trainees-burn-in ;agents engage in work task during burn in
  ask trainees [let success_a random 100 / 100
    ifelse success_a <= true_policy_a_reward [set reward_a 1
      set task_successes (task_successes + 1)]
      [set reward_a 0]
    set attempts_policy_a (attempts_policy_a + 1)
    set value_estimate_a (value_estimate_a + ((1 / attempts_policy_a) * (reward_a -
value_estimate_a))) ]
end

to update-globals ;calculate all global variables used to track sim functioning
  set mean_value_estimate_a mean [value_estimate_a] of trainees
  set mean_value_estimate_b mean [value_estimate_b] of trainees
  set mean_overall_task_success mean [task_successes] of trainees / ticks
  set mean_pretraining_success_rate mean [pretraining_success_rate] of trainees
  set mean_posttraining_success_rate mean [posttraining_success_rate] of trainees
  set mean_behavioral_transfer_rate mean [behavioral_transfer_rate] of trainees
end

to trainees-transfer ;call routine to choose which system will drive task
  system-choose
  ask trainees [set transfer_time_count (ticks - burn_in)]
  ask trainees [set behavioral_transfer_rate (attempts_policy_b / (transfer_time_count +
.000001))]
  ask trainees [set posttraining_success_rate (post_training_successes /
(transfer_time_count + .000001))]
```

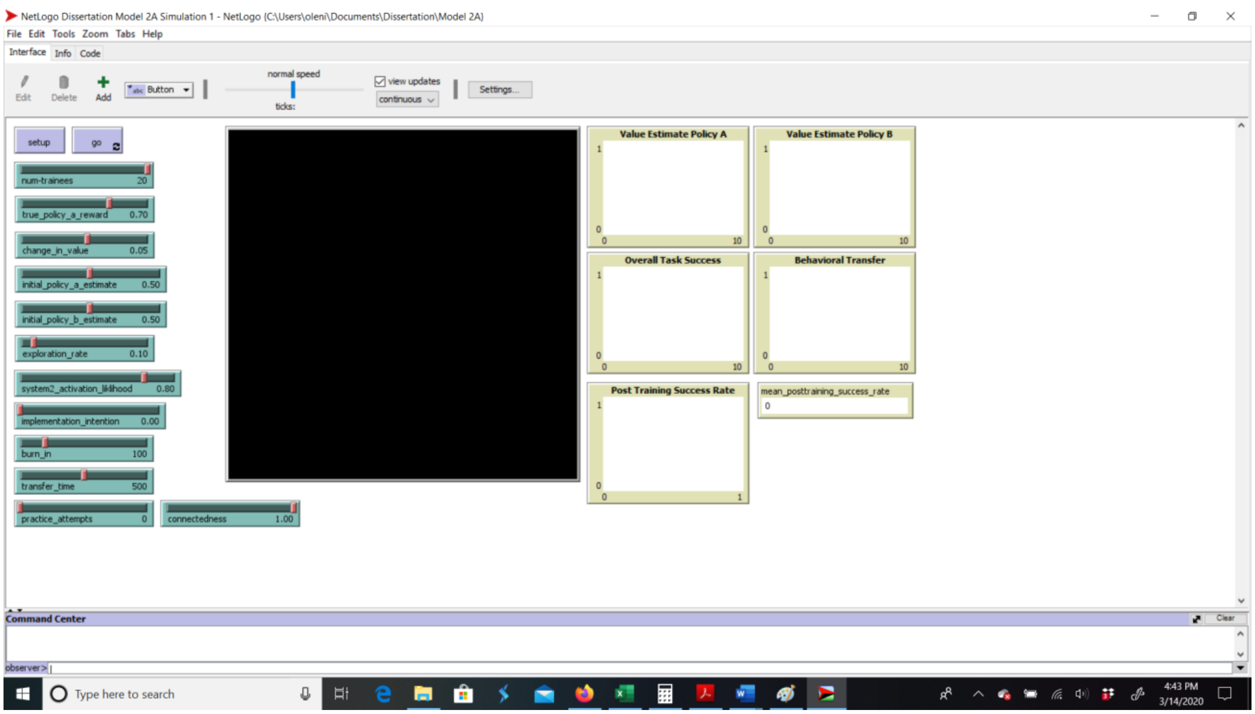
```
end

to system-choose ;decide if system2 will intervene, if not, rely on system 1
  ask trainees [
  let system_choose (random 100 / 100)
  if system_choose < system2_activation_liklihood [system2_decision]
  if system_choose >= system2_activation_liklihood [system1_decision]
  ]
end

to system1_decision ;agent makes automatic decision about which policy to apply
  set system1_choose_a ((attempts_policy_a / (attempts_policy_a + attempts_policy_b + practice_attempts + .000001)) - implementation_intention) ;update habitual decision rate ;note: all additions of .000001 are to avoid divisions by 0, number small so as not to affect simulation
  let choose_a random 100 / 100 ;generate random number to determine which policy to implement
    ifelse choose_a < system1_choose_a [ let success_a random 100 / 100 ;if Policy A chosen, determine if successful
      ifelse success_a < true_policy_a_reward [set reward_a 1 ;if successful receive reward
        set task_successes (task_successes + 1) ;update counts on task success
        set post_training_successes (post_training_successes + 1) ;update counts on task success
        set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
        set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a  + .000001)) * (reward_a - value_estimate_a))) ] ;update value estimate for Policy A
      [set reward_a 0 ;if unsuccessful set reward to 0 and update policy value estimate
        set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a + .000001)) * (reward_a - value_estimate_a))) ;update value estimate for Policy A
    set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
    ]
    ]
    [let success_b random 100 / 100 ;if Policy B chosen, determine if successful
      ifelse success_b < true_policy_b_reward [set reward_b 1 ;if successful receive reward
        set task_successes (task_successes + 1) ;update counts on task success
        set post_training_successes (post_training_successes + 1) ;update counts on task success
        set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
        set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b  + .000001)) * (reward_b - value_estimate_b))) ] ;update value estimate for Policy B
      [set reward_b 0 ;if unsuccessful set reward to 0 and update policy value estimate
        set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b + .000001)) * (reward_b - value_estimate_b))) ;update value estimate for Policy B
    set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
    ]
      ]
end
```

```
to system2_decision ;default to system 2 using highest value estimated policy except at
some error rate
  let e-greedy random 100 / 100
    ifelse e-greedy < exploration_rate [ run_low_value ] [ run_high_value ]
end

to save-burn-in ;save pretraining performance
  ask trainees [set pretraining_success_rate (task_successes / (burn_in + .000001))]
end

to run_low_value ;subroutine to choose and execute policy with lowest estimated value
   ifelse value_estimate_a <= value_estimate_b [ let success_a random 100 / 100 ;if
Policy A chosen, determine if successful
      ifelse success_a < true_policy_a_reward [set reward_a 1 ;if successful receive reward
        set task_successes (task_successes + 1) ;update counts on task success
        set post_training_successes (post_training_successes + 1) ;update counts on task
success
        set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
        set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a  + .000001)) *
(reward_a - value_estimate_a))) ] ;update value estimate for Policy A
      [set reward_a 0 ;if unsuccessful set reward to 0 and update policy value estimate
        set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a + .000001)) *
(reward_a - value_estimate_a))) ;update value estimate for Policy A
      if value_estimate_a < 0 [set value_estimate_a 0]
    set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
    ]
    ]
    [let success_b random 100 / 100 ;if Policy B chosen, determine if successful
      ifelse success_b < true_policy_b_reward [set reward_b 1 ;if successful receive reward
        set task_successes (task_successes + 1) ;update counts on task success
        set post_training_successes (post_training_successes + 1) ;update counts on task
success
        set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
        set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b  + .000001)) *
(reward_b - value_estimate_b))) ] ;update value estimate for Policy B
      [set reward_b 0 ;if unsuccessful set reward to 0 and update policy value estimate
        set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b + .000001)) *
(reward_b - value_estimate_b))) ;update value estimate for Policy B
      if value_estimate_b < 0 [set value_estimate_b 0]
    set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
    ]
      ]
end

to run_high_value ;subroutine to choose and execute policy with highest estimated value
```

```
    ifelse value_estimate_a >= value_estimate_b [ let success_a random 100 / 100 ;if
Policy A chosen, determine if successful
      ifelse success_a < true_policy_a_reward [set reward_a 1 ;if successful receive reward
        set task_successes (task_successes + 1) ;update counts on task success
        set post_training_successes (post_training_successes + 1) ;update counts on task
success
        set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
        set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a  + .000001)) *
(reward_a - value_estimate_a))) ] ;update value estimate for Policy A
      [set reward_a 0 ;if unsuccessful set reward to 0 and update policy value estimate
        set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a + .000001)) *
(reward_a - value_estimate_a))) ;update value estimate for Policy A
      if value_estimate_a < 0 [set value_estimate_a 0]
     set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
     ]
     ]
     [let success_b random 100 / 100 ;if Policy B chosen, determine if successful
      ifelse success_b < true_policy_b_reward [set reward_b 1 ;if successful receive reward
        set task_successes (task_successes + 1) ;update counts on task success
        set post_training_successes (post_training_successes + 1) ;update counts on task
success
        set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
        set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b  + .000001)) *
(reward_b - value_estimate_b))) ] ;update value estimate for Policy B
      [set reward_b 0 ;if unsuccessful set reward to 0 and update policy value estimate
        set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b + .000001)) *
(reward_b - value_estimate_b))) ;update value estimate for Policy B
      if value_estimate_b < 0 [set value_estimate_b 0]
     set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
     ]
      ]
   end

   to save-post-training ;save post training performance variables
     ask trainees [set posttraining_success_rate (post_training_successes / (transfer_time +
.000001))]
     ask trainees [set behavioral_transfer_rate (attempts_policy_b / (transfer_time +
.000001))]
end
```

Appendix B: Study 2A Environment and Code

Figure 83. Snapshot of the modeling environment for Study 2A in NetLogo.



Algorithm 14. NetLogo Code for Study 2A Model

```
breed [trainees trainee] ;types of agents allowed in environment

    trainees-own [
      value_estimate_a ;estimated value of Policy A
      value_estimate_b ;estimated value of Policy B
      system1_choose_a ;liklihood of choosing Policy A as habitual response
      attempts_policy_a ;number times applied Policy A
      attempts_policy_b ;number time applied Policy B
      reward_a ;reward received on most recent attempt with Policy A
      reward_b ;reward received on most recent attempt with Policy B
      task_successes ;number of times successful at task overall
      post_training_successes ;number of times successful only post-training
      pretraining_success_rate ;success rate pretraining only
      posttraining_success_rate ; percentage of times successful in post-training environment
      behavioral_transfer_rate ;rate of choosing Policy B in transfer environment
      transfer_time_count ; ticks into transfer time
      other_agent_estimate_a ;value estimate of other agents in model for Policy A
      other_agent_estimate_b ;value estimate of other agents in model for Policy B
      grouped_value_estimate_a ;combined value estimate of target agent and other agents for
Policy A
      grouped_value_estimate_b ;combined value estimate of target agent and other agents for
Policy B
      ]

    globals [
      mean_value_estimate_a ;mean of agent value estimates for Policy A
      mean_value_estimate_b ;mean of agent value estimates for Policy B
      mean_overall_task_success ;task rate of success for full simulation
      mean_pretraining_success_rate ;success rate pretraining only all agents
      mean_posttraining_success_rate ;success rate posttraining only all agents
      mean_behavioral_transfer_rate ;rate of choosing Policy B in transfer environment all
agents
      true_policy_b_reward ;reward for Policy B after adjusting for policy value change
    ]

    to setup
      clear-all ;clears environment from previous simulation
      create-trainees num-trainees [ ;place specified number of agents at center of grid
        set value_estimate_a initial_policy_a_estimate ;set initial value estimate for Policy A
for each trainee
        set value_estimate_b initial_policy_b_estimate ;set initial value estimate for Policy B
for each trainee
        set attempts_policy_a 0 ;number times applied Policy A initial set to 0
        set attempts_policy_b 0 ;number time applied Policy B initial set to 0
```

```
      set task_successes 0 ;number of task successes initial set to 0
      set pretraining_success_rate 0 ;success rate pretraining only initial set to 0
      set post_training_successes 0 ;number of successes for post training initial set to 0
      set posttraining_success_rate 0 ;success rate in posttraining environment initial set to 0
      set behavioral_transfer_rate 0 ;percentage of time choosing trained policy initial set to
0
    ]
    layout-circle (sort turtles) max-pxcor - 3
    set true_policy_b_reward (true_policy_a_reward + change_in_value)
    if true_policy_b_reward > 1 [set true_policy_b_reward 1]
    if true_policy_b_reward < 0 [set true_policy_b_reward 0]
    reset-ticks ;reset time count to 0
  end

  to go ;primary subroutines activated
    if ticks = (burn_in + transfer_time) [save-post-training] ;call subroutine to save post
training variables
    if ticks = (burn_in + transfer_time) [stop] ;control length of sim
    tick ;advance time
    if ticks <= burn_in [trainees-burn-in] ;call subroutine to have trainee engage in task
during burn in period
    if ticks > burn_in [trainees-transfer] ;call subroutine for trainee decisions post training
    if ticks = burn_in [save-burn-in] ;call subroutine to save pretraining performance
    ifelse num-trainees > 1 [pool_experiences] [no_pool_experiences] ;set group estimate
depending on if more than 1 agent or not
    update-globals ;call subroutine to calculate all global variables used to track sim
functioning
  end

  to trainees-burn-in ;agents engage in work task during burn in
    ask trainees [let success_a random 100 / 100
      ifelse success_a <= true_policy_a_reward [set reward_a 1
        set task_successes (task_successes + 1)]
        [set reward_a 0]
      set attempts_policy_a (attempts_policy_a + 1)
      set value_estimate_a (value_estimate_a + ((1 / attempts_policy_a) * (reward_a -
value_estimate_a))) ]
  end

  to update-globals ;calculate all global variables used to track sim functioning
    set mean_value_estimate_a mean [value_estimate_a] of trainees
    set mean_value_estimate_b mean [value_estimate_b] of trainees
    set mean_overall_task_success mean [task_successes] of trainees / ticks
    set mean_pretraining_success_rate mean [pretraining_success_rate] of trainees
    set mean_posttraining_success_rate mean [posttraining_success_rate] of trainees
    set mean_behavioral_transfer_rate mean [behavioral_transfer_rate] of trainees
```

```
end

to trainees-transfer ;call routine to choose which system will drive task
  system-choose
  ask trainees [set transfer_time_count (ticks - burn_in)]
  ask trainees [set behavioral_transfer_rate (attempts_policy_b / (transfer_time_count +
.000001))]
  ask trainees [set posttraining_success_rate (post_training_successes /
(transfer_time_count + .000001))]
end

to system-choose ;decide if system2 will intervene, if not, rely on system 1
  ask trainees [
  let system_choose (random 100 / 100)
  if system_choose < system2_activation_liklihood [system2_decision]
  if system_choose >= system2_activation_liklihood [system1_decision]
  ]
end

to system1_decision ;agent makes automatic decision about which policy to apply
  set system1_choose_a ((attempts_policy_a / (attempts_policy_a + attempts_policy_b +
practice_attempts + .000001)) - implementation_intention) ;update habitual decision rate ;note:
all additions of .000001 are to avoid divisions by 0, number small so as not to affect simulation
  let choose_a random 100 / 100 ;generate random number to determine which policy to
implement
    ifelse choose_a < system1_choose_a [ let success_a random 100 / 100 ;if Policy A
chosen, determine if successful
      ifelse success_a < true_policy_a_reward [set reward_a 1 ;if successful receive reward
        set task_successes (task_successes + 1) ;update counts on task success
        set post_training_successes (post_training_successes + 1) ;update counts on task
success
        set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
        set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a  + .000001)) *
(reward_a - value_estimate_a))) ] ;update value estimate for Policy A
      [set reward_a 0 ;if unsuccessful set reward to 0 and update policy value estimate
        set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a + .000001)) *
(reward_a - value_estimate_a))) ;update value estimate for Policy A
    set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
    ]
    ]
    [let success_b random 100 / 100 ;if Policy B chosen, determine if successful
      ifelse success_b < true_policy_b_reward [set reward_b 1 ;if successful receive reward
        set task_successes (task_successes + 1) ;update counts on task success
        set post_training_successes (post_training_successes + 1) ;update counts on task
success
        set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
```

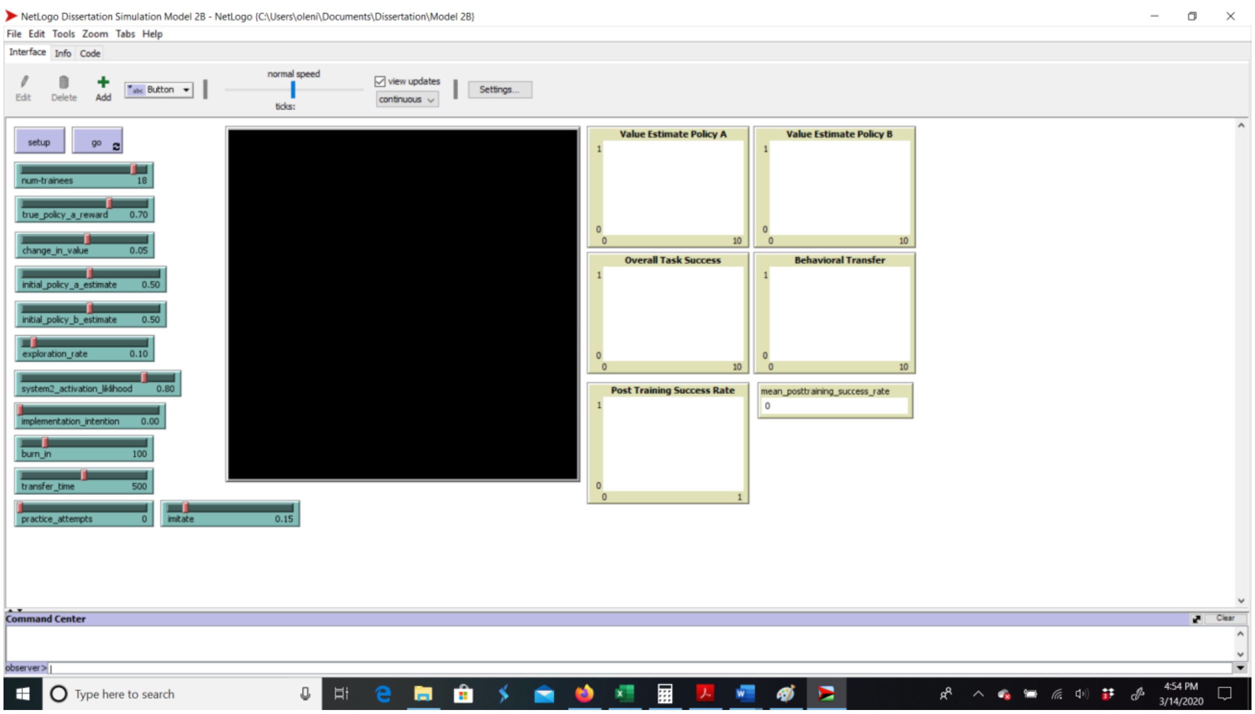
```
          set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b  + .000001)) *
(reward_b - value_estimate_b))) ] ;update value estimate for Policy B
         [set reward_b 0 ;if unsuccessful set reward to 0 and update policy value estimate
          set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b + .000001)) *
(reward_b - value_estimate_b))) ;update value estimate for Policy B
        set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
        ]
         ]
      end

      to system2_decision ;default to system 2 using highest value estimated policy except at
some error rate
       let e-greedy random 100 / 100
        ifelse e-greedy < exploration_rate [ run_low_value ] [ run_high_value ]
      end

      to save-burn-in ;save pretraining performance
       ask trainees [set pretraining_success_rate (task_successes / (burn_in + .000001))]
      end

      to run_low_value ;subroutine to choose and execute policy with lowest estimated value
       ifelse value_estimate_a <= value_estimate_b [ let success_a random 100 / 100 ;if
Policy A chosen, determine if successful
         ifelse success_a < true_policy_a_reward [set reward_a 1 ;if successful receive reward
          set task_successes (task_successes + 1) ;update counts on task success
          set post_training_successes (post_training_successes + 1) ;update counts on task
success
          set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
          set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a  + .000001)) *
(reward_a - value_estimate_a))) ] ;update value estimate for Policy A
         [set reward_a 0 ;if unsuccessful set reward to 0 and update policy value estimate
          set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a + .000001)) *
(reward_a - value_estimate_a))) ;update value estimate for Policy A
         if value_estimate_a < 0 [set value_estimate_a 0]
        set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
        ]
        ]
        [let success_b random 100 / 100 ;if Policy B chosen, determine if successful
         ifelse success_b < true_policy_b_reward [set reward_b 1 ;if successful receive reward
          set task_successes (task_successes + 1) ;update counts on task success
          set post_training_successes (post_training_successes + 1) ;update counts on task
success
          set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
          set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b  + .000001)) *
(reward_b - value_estimate_b))) ] ;update value estimate for Policy B
         [set reward_b 0 ;if unsuccessful set reward to 0 and update policy value estimate
```

```
          set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b + .000001)) *
(reward_b - value_estimate_b))) ;update value estimate for Policy B
         if value_estimate_b < 0 [set value_estimate_b 0]
        set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
        ]
         ]
      end

      to run_high_value ;subroutine to choose and execute policy with highest estimated value
        ifelse value_estimate_a >= value_estimate_b [ let success_a random 100 / 100 ;if
Policy A chosen, determine if successful
         ifelse success_a < true_policy_a_reward [set reward_a 1 ;if successful receive reward
          set task_successes (task_successes + 1) ;update counts on task success
          set post_training_successes (post_training_successes + 1) ;update counts on task
success
          set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
          set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a  + .000001)) *
(reward_a - value_estimate_a))) ] ;update value estimate for Policy A
         [set reward_a 0 ;if unsuccessful set reward to 0 and update policy value estimate
          set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a + .000001)) *
(reward_a - value_estimate_a))) ;update value estimate for Policy A
         if value_estimate_a < 0 [set value_estimate_a 0]
        set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
        ]
        ]
        [let success_b random 100 / 100 ;if Policy B chosen, determine if successful
         ifelse success_b < true_policy_b_reward [set reward_b 1 ;if successful receive reward
          set task_successes (task_successes + 1) ;update counts on task success
          set post_training_successes (post_training_successes + 1) ;update counts on task
success
          set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
          set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b  + .000001)) *
(reward_b - value_estimate_b))) ] ;update value estimate for Policy B
         [set reward_b 0 ;if unsuccessful set reward to 0 and update policy value estimate
          set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b + .000001)) *
(reward_b - value_estimate_b))) ;update value estimate for Policy B
         if value_estimate_b < 0 [set value_estimate_b 0]
        set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
        ]
         ]
      end

      to save-post-training ;save post training performance variables
       ask trainees [set posttraining_success_rate (post_training_successes / (transfer_time +
.000001))]
```

```
    ask trainees [set behavioral_transfer_rate (attempts_policy_b / (transfer_time +
.000001))]
  end

  to pool_experiences ;pool experiences from all agents for decision making
    ask trainees [set other_agent_estimate_a (mean [value_estimate_a] of other trainees)]
    ask trainees [set other_agent_estimate_b (mean [value_estimate_b] of other trainees)]
    ask trainees [set grouped_value_estimate_a (((1 -
connectedness)*(value_estimate_a))+(connectedness * other_agent_estimate_a))]
    ask trainees [set grouped_value_estimate_b (((1 -
connectedness)*(value_estimate_b))+(connectedness * other_agent_estimate_b))]
  end

  to no_pool_experiences ;if only 1 agent then group estimate is equal to personal estimate
    ask trainees [set grouped_value_estimate_a (value_estimate_a)]
    ask trainees [set grouped_value_estimate_b (value_estimate_b)]
end
```

Appendix C: Study 2B Environment and Code

Figure 84. Snapshot of the modeling environment for Study 2B in NetLogo.



Algorithm 15. NetLogo Code for Study 2B Model

```
  breed [trainees trainee] ;types of agents allowed in environment

trainees-own [
  value_estimate_a ;estimated value of Policy A
  value_estimate_b ;estimated value of Policy B
  system1_choose_a ;liklihood of choosing Policy A as habitual response
  attempts_policy_a ;number times applied Policy A
  attempts_policy_b ;number time applied Policy B
  reward_a ;reward received on most recent attempt with Policy A
  reward_b ;reward received on most recent attempt with Policy B
  task_successes ;number of times successful at task overall
  post_training_successes ;number of times successful only post-training
  pretraining_success_rate ;success rate pretraining only
  posttraining_success_rate ; percentage of times successful in post-training environment
  behavioral_transfer_rate ;rate of choosing Policy B in transfer environment
  transfer_time_count ; ticks into transfer time
  chose_b ;track behavioral choice of last task attempt, 0 = chose a, 1 = chose b
  other_success_rate ;success rate of most successful other trainee
  imitate_choice ;track decision to imitate on each time step
  other_chose_b ;behavioral choice of most successful other trainee
  ]

globals [
  mean_value_estimate_a ;mean of agent value estimates for Policy A
  mean_value_estimate_b ;mean of agent value estimates for Policy B
  mean_overall_task_success ;task rate of success for full simulation
  mean_pretraining_success_rate ;success rate pretraining only all agents
  mean_posttraining_success_rate ;success rate posttraining only all agents
  mean_behavioral_transfer_rate ;rate of choosing Policy B in transfer environment all agents
  true_policy_b_reward ;reward for Policy B after adjusting for policy value change
]

to setup
  clear-all ;clears environment from previous simulation
  create-trainees num-trainees [ setxy random-xcor random-ycor ;place specified number of
agents at random coordinates
    set value_estimate_a initial_policy_a_estimate ;set initial value estimate for Policy A for each
trainee
    set value_estimate_b initial_policy_b_estimate ;set initial value estimate for Policy B for each
trainee
    set attempts_policy_a 0 ;number times applied Policy A initial set to 0
    set attempts_policy_b 0 ;number time applied Policy B initial set to 0
    set task_successes 0 ;number of task successes initial set to 0
    set pretraining_success_rate 0 ;success rate pretraining only initial set to 0
```

```
    set post_training_successes 0 ;number of successes for post training initial set to 0
    set posttraining_success_rate 0 ;success rate in posttraining environment initial set to 0
    set behavioral_transfer_rate 0 ;percentage of time choosing trained policy initial set to 0
    set chose_b 0 ;set choice tracker to default of Policy A
    set other_success_rate 0 ;setup success rate of most successful other trainee
    set other_chose_b 0 ;setup choice made by other most successful trainee
  ]
  layout-circle (sort turtles) max-pxcor - 3
  set true_policy_b_reward (true_policy_a_reward + change_in_value)
  if true_policy_b_reward > 1 [set true_policy_b_reward 1]
  if true_policy_b_reward < 0 [set true_policy_b_reward 0]
  reset-ticks ;reset time count to 0
end

to go ;primary subroutines activated
  if ticks = (burn_in + transfer_time) [save-post-training] ;call subroutine to save post training
variables
  if ticks = (burn_in + transfer_time) [stop] ;control length of sim
  tick ;advance time
  if ticks <= burn_in [trainees-burn-in] ;call subroutine to have trainee engage in task during burn
in period
  if ticks > burn_in [trainees-transfer] ;call subroutine for trainee decisions post training
  if ticks = burn_in [save-burn-in] ;call subroutine to save pretraining performance
  update-globals ;call subroutine to calculate all global variables used to track sim functioning
end

to trainees-burn-in ;agents engage in work task during burn in
  ask trainees [let success_a random 100 / 100
    ifelse success_a <= true_policy_a_reward [set reward_a 1
      set task_successes (task_successes + 1)]
      [set reward_a 0]
    set attempts_policy_a (attempts_policy_a + 1)
    set value_estimate_a (value_estimate_a + ((1 / attempts_policy_a) * (reward_a -
value_estimate_a))) ]
end

to update-globals ;calculate all global variables used to track sim functioning
  set mean_value_estimate_a mean [value_estimate_a] of trainees
  set mean_value_estimate_b mean [value_estimate_b] of trainees
  set mean_overall_task_success mean [task_successes] of trainees / ticks
  set mean_pretraining_success_rate mean [pretraining_success_rate] of trainees
  set mean_posttraining_success_rate mean [posttraining_success_rate] of trainees
  set mean_behavioral_transfer_rate mean [behavioral_transfer_rate] of trainees
end

to trainees-transfer ;call routine to choose which system will drive task
```

```
  system-choose
  ask trainees [set transfer_time_count (ticks - burn_in)]
  ask trainees [set behavioral_transfer_rate (attempts_policy_b / (transfer_time_count +
.000001))]
  ask trainees [set posttraining_success_rate (post_training_successes / (transfer_time_count +
.000001))]
end

to system-choose ;decide if system2 will intervene, if not, rely on system 1
  ask trainees [
  let system_choose (random 100 / 100)
  if system_choose < system2_activation_liklihood [system2_decision]
  if system_choose >= system2_activation_liklihood [system1_decision]
  ]
end

to system1_decision ;agent makes automatic decision about which policy to apply
  set system1_choose_a ((attempts_policy_a / (attempts_policy_a + attempts_policy_b +
practice_attempts + .000001)) - implementation_intention) ;update habitual decision rate ;note:
all additions of .000001 are to avoid divisions by 0, number small so as not to affect simulation
  let choose_a random 100 / 100 ;generate random number to determine which policy to
implement
    ifelse choose_a < system1_choose_a [ let success_a random 100 / 100 ;if Policy A chosen,
determine if successful
      ifelse success_a < true_policy_a_reward [set reward_a 1 ;if successful receive reward
        set task_successes (task_successes + 1) ;update counts on task success
        set post_training_successes (post_training_successes + 1) ;update counts on task success
        set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
        set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a  + .000001)) * (reward_a
- value_estimate_a))) ] ;update value estimate for Policy A
      [set reward_a 0 ;if unsuccessful set reward to 0 and update policy value estimate
        set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a + .000001)) * (reward_a
- value_estimate_a))) ;update value estimate for Policy A
    set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
      set chose_b 0 ;update choice to Policy A
    ]
    ]
    [let success_b random 100 / 100 ;if Policy B chosen, determine if successful
      ifelse success_b < true_policy_b_reward [set reward_b 1 ;if successful receive reward
        set task_successes (task_successes + 1) ;update counts on task success
        set post_training_successes (post_training_successes + 1) ;update counts on task success
        set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
        set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b  + .000001)) *
(reward_b - value_estimate_b))) ] ;update value estimate for Policy B
      [set reward_b 0 ;if unsuccessful set reward to 0 and update policy value estimate
```

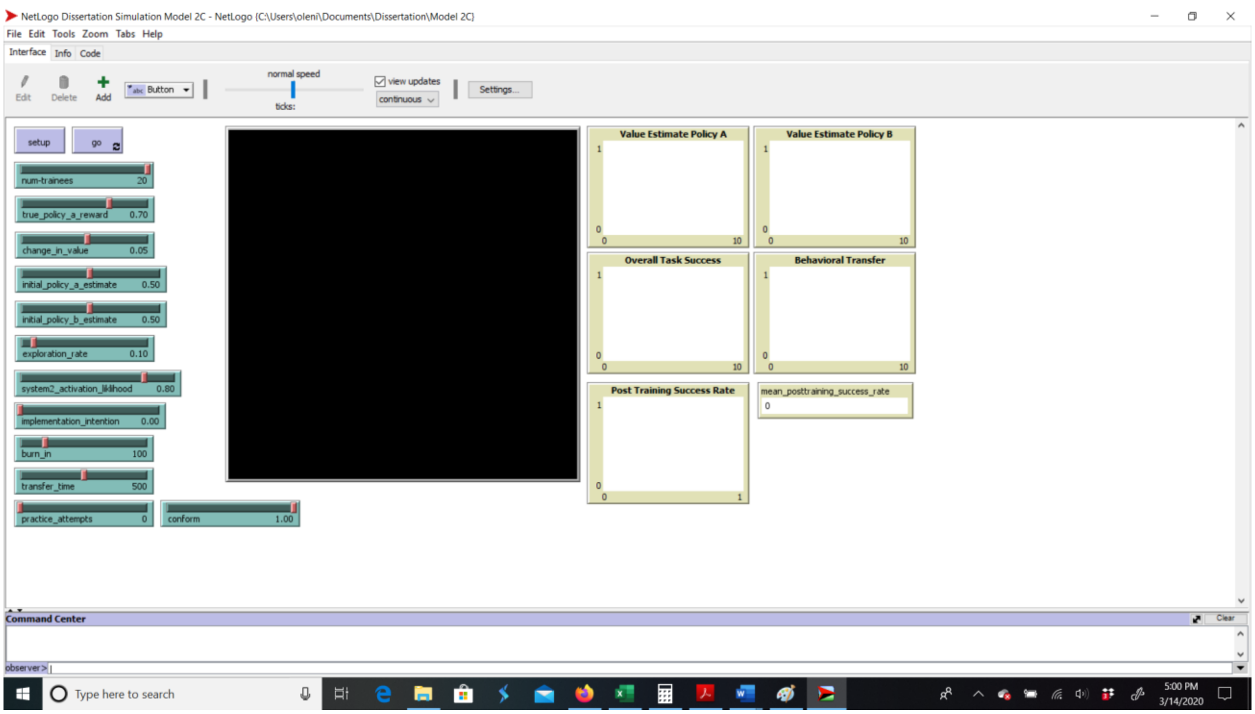
```
      set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b + .000001)) * (reward_b
- value_estimate_b))) ;update value estimate for Policy B
    set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
      set chose_b 1 ;update choice to Policy B
    ]
      ]
end

to system2_decision ;default to system 2 using highest value estimated policy except at some
error rate
  ifelse num-trainees > 1 [
    run-imitate ;have trainee choose if it will imitate or not if there are other trainees
    if imitate_choice = 0 [let e-greedy random 100 / 100 ;if not imitating run egreedy as normal
      ifelse e-greedy < exploration_rate [ run_low_value ] [ run_high_value ]]
  ]
  [
  let e-greedy random 100 / 100 ;run choice with some degree of error
    ifelse e-greedy < exploration_rate [ run_low_value ] [ run_high_value ]
  ]
end

to save-burn-in ;save pretraining performance
  ask trainees [set pretraining_success_rate (task_successes / (burn_in + .000001))]
end

to run_low_value ;subroutine to choose and execute policy with lowest estimated value
   ifelse value_estimate_a <= value_estimate_b [ let success_a random 100 / 100 ;if Policy A
chosen, determine if successful
      ifelse success_a < true_policy_a_reward [set reward_a 1 ;if successful receive reward
        set task_successes (task_successes + 1) ;update counts on task success
        set post_training_successes (post_training_successes + 1) ;update counts on task success
        set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
        set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a  + .000001)) * (reward_a
- value_estimate_a))) ] ;update value estimate for Policy A
      [set reward_a 0 ;if unsuccessful set reward to 0 and update policy value estimate
        set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a + .000001)) * (reward_a
- value_estimate_a))) ;update value estimate for Policy A
      if value_estimate_a < 0 [set value_estimate_a 0]
    set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
      set chose_b 0 ;update choice to Policy A
    ]
    ]
    [let success_b random 100 / 100 ;if Policy B chosen, determine if successful
      ifelse success_b < true_policy_b_reward [set reward_b 1 ;if successful receive reward
        set task_successes (task_successes + 1) ;update counts on task success
        set post_training_successes (post_training_successes + 1) ;update counts on task success
```

```
      set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
      set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b  + .000001)) *
(reward_b - value_estimate_b))) ] ;update value estimate for Policy B
     [set reward_b 0 ;if unsuccessful set reward to 0 and update policy value estimate
      set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b + .000001)) * (reward_b
- value_estimate_b))) ;update value estimate for Policy B
     if value_estimate_b < 0 [set value_estimate_b 0]
    set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
     set chose_b 1 ;update choice to Policy B
    ]
     ]
end

to run_high_value ;subroutine to choose and execute policy with highest estimated value
   ifelse value_estimate_a >= value_estimate_b [ let success_a random 100 / 100 ;if Policy A
chosen, determine if successful
     ifelse success_a < true_policy_a_reward [set reward_a 1 ;if successful receive reward
      set task_successes (task_successes + 1) ;update counts on task success
      set post_training_successes (post_training_successes + 1) ;update counts on task success
      set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
      set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a  + .000001)) * (reward_a
- value_estimate_a))) ] ;update value estimate for Policy A
     [set reward_a 0 ;if unsuccessful set reward to 0 and update policy value estimate
      set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a + .000001)) * (reward_a
- value_estimate_a))) ;update value estimate for Policy A
     if value_estimate_a < 0 [set value_estimate_a 0]
    set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
     set chose_b 0 ;update choice to Policy A
    ]
    ]
    [let success_b random 100 / 100 ;if Policy B chosen, determine if successful
     ifelse success_b < true_policy_b_reward [set reward_b 1 ;if successful receive reward
      set task_successes (task_successes + 1) ;update counts on task success
      set post_training_successes (post_training_successes + 1) ;update counts on task success
      set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
      set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b  + .000001)) *
(reward_b - value_estimate_b))) ] ;update value estimate for Policy B
     [set reward_b 0 ;if unsuccessful set reward to 0 and update policy value estimate
      set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b + .000001)) * (reward_b
- value_estimate_b))) ;update value estimate for Policy B
     if value_estimate_b < 0 [set value_estimate_b 0]
    set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
     set chose_b 1 ;update choice to Policy B
    ]
     ]
end
```

```
to save-post-training ;save post training performance variables
  ask trainees [set posttraining_success_rate (post_training_successes / (transfer_time +
.000001))]
  ask trainees [set behavioral_transfer_rate (attempts_policy_b / (transfer_time + .000001))]
end

to run-imitate ;make imitate decision based on specified rate and execute
  let imitate_yes random 100 / 100
  ifelse imitate_yes <= imitate [set imitate_choice 1] [set imitate_choice 0]
  set other_chose_b [chose_b] of other trainees with-max [posttraining_success_rate]
  if imitate_choice = 1 [
    ifelse other_chose_b = 0 [let success_a random 100 / 100 ;if Policy A chosen, determine if
successful
      ifelse success_a < true_policy_a_reward [set reward_a 1 ;if successful receive reward
        set task_successes (task_successes + 1) ;update counts on task success
        set post_training_successes (post_training_successes + 1) ;update counts on task success
        set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
        set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a  + .000001)) * (reward_a
- value_estimate_a))) ] ;update value estimate for Policy A
      [set reward_a 0 ;if unsuccessful set reward to 0 and update policy value estimate
        set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a + .000001)) * (reward_a
- value_estimate_a))) ;update value estimate for Policy A
      if value_estimate_a < 0 [set value_estimate_a 0]
    set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
      set chose_b 0 ;update choice to Policy A
    ]
    ]
    [let success_b random 100 / 100 ;if Policy B chosen, determine if successful
      ifelse success_b < true_policy_b_reward [set reward_b 1 ;if successful receive reward
        set task_successes (task_successes + 1) ;update counts on task success
        set post_training_successes (post_training_successes + 1) ;update counts on task success
        set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
        set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b  + .000001)) *
(reward_b - value_estimate_b))) ] ;update value estimate for Policy B
      [set reward_b 0 ;if unsuccessful set reward to 0 and update policy value estimate
        set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b + .000001)) * (reward_b
- value_estimate_b))) ;update value estimate for Policy B
      if value_estimate_b < 0 [set value_estimate_b 0]
    set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
      set chose_b 1 ;update choice to Policy B
    ]
      ]
      ]
end
```

Appendix D: Study 2C Environment and Code

Figure 85. Snapshot of the modeling environment for Study 2C in NetLogo.



Algorithm 16. NetLogo Code for Study 2C Model

```
breed [trainees trainee] ;types of agents allowed in environment

trainees-own [
  value_estimate_a ;estimated value of Policy A
  value_estimate_b ;estimated value of Policy B
  system1_choose_a ;liklihood of choosing Policy A as habitual response
  attempts_policy_a ;number times applied Policy A
  attempts_policy_b ;number time applied Policy B
  reward_a ;reward received on most recent attempt with Policy A
  reward_b ;reward received on most recent attempt with Policy B
  task_successes ;number of times successful at task overall
  post_training_successes ;number of times successful only post-training
  pretraining_success_rate ;success rate pretraining only
  posttraining_success_rate ; percentage of times successful in post-training environment
  behavioral_transfer_rate ;rate of choosing Policy B in transfer environment
  transfer_time_count ; ticks into transfer time
  chose_b ;track behavioral choice of last task attempt, 0 = chose a, 1 = chose b
  other_success_rate ;success rate of most successful other trainee
  conform_choice ;track decision to conform on each time step
  other_chose_b ;behavioral choice of most successful other trainee
  ]

globals [
  mean_value_estimate_a ;mean of agent value estimates for Policy A
  mean_value_estimate_b ;mean of agent value estimates for Policy B
  mean_overall_task_success ;task rate of success for full simulation
  mean_pretraining_success_rate ;success rate pretraining only all agents
  mean_posttraining_success_rate ;success rate posttraining only all agents
  mean_behavioral_transfer_rate ;rate of choosing Policy B in transfer environment all agents
  true_policy_b_reward ;reward for Policy B after adjusting for policy value change
]

to setup
  clear-all ;clears environment from previous simulation
  create-trainees num-trainees [ setxy random-xcor random-ycor ;place specified number of
agents at random coordinates
    set value_estimate_a initial_policy_a_estimate ;set initial value estimate for Policy A for each
trainee
    set value_estimate_b initial_policy_b_estimate ;set initial value estimate for Policy B for each
trainee
    set attempts_policy_a 0 ;number times applied Policy A initial set to 0
    set attempts_policy_b 0 ;number time applied Policy B initial set to 0
    set task_successes 0 ;number of task successes initial set to 0
    set pretraining_success_rate 0 ;success rate pretraining only initial set to 0
```

```
    set post_training_successes 0 ;number of successes for post training initial set to 0
    set posttraining_success_rate 0 ;success rate in posttraining environment initial set to 0
    set behavioral_transfer_rate 0 ;percentage of time choosing trained policy initial set to 0
    set chose_b 0 ;set choice tracker to default of Policy A
    set other_success_rate 0 ;setup success rate of most successful other trainee
    set other_chose_b 0 ;setup choice made by other most successful trainee
  ]
  layout-circle (sort turtles) max-pxcor - 3
  set true_policy_b_reward (true_policy_a_reward + change_in_value)
  if true_policy_b_reward > 1 [set true_policy_b_reward 1]
  if true_policy_b_reward < 0 [set true_policy_b_reward 0]
  reset-ticks ;reset time count to 0
end

to go ;primary subroutines activated
  if ticks = (burn_in + transfer_time) [save-post-training] ;call subroutine to save post training
variables
  if ticks = (burn_in + transfer_time) [stop] ;control length of sim
  tick ;advance time
  if ticks <= burn_in [trainees-burn-in] ;call subroutine to have trainee engage in task during burn
in period
  if ticks > burn_in [trainees-transfer] ;call subroutine for trainee decisions post training
  if ticks = burn_in [save-burn-in] ;call subroutine to save pretraining performance
  update-globals ;call subroutine to calculate all global variables used to track sim functioning
end

to trainees-burn-in ;agents engage in work task during burn in
  ask trainees [let success_a random 100 / 100
    ifelse success_a <= true_policy_a_reward [set reward_a 1
      set task_successes (task_successes + 1)]
      [set reward_a 0]
    set attempts_policy_a (attempts_policy_a + 1)
    set value_estimate_a (value_estimate_a + ((1 / attempts_policy_a) * (reward_a -
value_estimate_a))) ]
end

to update-globals ;calculate all global variables used to track sim functioning
  set mean_value_estimate_a mean [value_estimate_a] of trainees
  set mean_value_estimate_b mean [value_estimate_b] of trainees
  set mean_overall_task_success mean [task_successes] of trainees / ticks
  set mean_pretraining_success_rate mean [pretraining_success_rate] of trainees
  set mean_posttraining_success_rate mean [posttraining_success_rate] of trainees
  set mean_behavioral_transfer_rate mean [behavioral_transfer_rate] of trainees
end

to trainees-transfer ;call routine to choose which system will drive task
```

```
  system-choose
  ask trainees [set transfer_time_count (ticks - burn_in)]
  ask trainees [set behavioral_transfer_rate (attempts_policy_b / (transfer_time_count +
.000001))]
  ask trainees [set posttraining_success_rate (post_training_successes / (transfer_time_count +
.000001))]
end

to system-choose ;decide if system2 will intervene, if not, rely on system 1
  ask trainees [
  let system_choose (random 100 / 100)
  if system_choose < system2_activation_liklihood [system2_decision]
  if system_choose >= system2_activation_liklihood [system1_decision]
  ]
end

to system1_decision ;agent makes automatic decision about which policy to apply
  set system1_choose_a ((attempts_policy_a / (attempts_policy_a + attempts_policy_b +
practice_attempts + .000001)) - implementation_intention) ;update habitual decision rate ;note:
all additions of .000001 are to avoid divisions by 0, number small so as not to affect simulation
  let choose_a random 100 / 100 ;generate random number to determine which policy to
implement
    ifelse choose_a < system1_choose_a [ let success_a random 100 / 100 ;if Policy A chosen,
determine if successful
      ifelse success_a < true_policy_a_reward [set reward_a 1 ;if successful receive reward
        set task_successes (task_successes + 1) ;update counts on task success
        set post_training_successes (post_training_successes + 1) ;update counts on task success
        set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
        set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a  + .000001)) * (reward_a
- value_estimate_a))) ] ;update value estimate for Policy A
      [set reward_a 0 ;if unsuccessful set reward to 0 and update policy value estimate
        set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a + .000001)) * (reward_a
- value_estimate_a))) ;update value estimate for Policy A
    set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
      set chose_b 0 ;update choice to Policy A
    ]
    ]
    [let success_b random 100 / 100 ;if Policy B chosen, determine if successful
      ifelse success_b < true_policy_b_reward [set reward_b 1 ;if successful receive reward
        set task_successes (task_successes + 1) ;update counts on task success
        set post_training_successes (post_training_successes + 1) ;update counts on task success
        set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
        set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b  + .000001)) *
(reward_b - value_estimate_b))) ] ;update value estimate for Policy B
      [set reward_b 0 ;if unsuccessful set reward to 0 and update policy value estimate
```

```
    set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b + .000001)) * (reward_b
- value_estimate_b))) ;update value estimate for Policy B
  set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
   set chose_b 1 ;update choice to Policy B
  ]
   ]
end

to system2_decision ;default to system 2 using highest value estimated policy except at some
error rate
 ifelse num-trainees > 1 [
  run-conform ;have trainee choose if it will conform or not if there are other trainees
  if conform_choice = 0 [let e-greedy random 100 / 100 ;if not imitating run egreedy as normal
   ifelse e-greedy < exploration_rate [ run_low_value ] [ run_high_value ]]
 ]
 [
 let e-greedy random 100 / 100 ;run choice with some degree of error
  ifelse e-greedy < exploration_rate [ run_low_value ] [ run_high_value ]
 ]
end

to save-burn-in ;save pretraining performance
 ask trainees [set pretraining_success_rate (task_successes / (burn_in + .000001))]
end

to run_low_value ;subroutine to choose and execute policy with lowest estimated value
  ifelse value_estimate_a <= value_estimate_b [ let success_a random 100 / 100 ;if Policy A
chosen, determine if successful
   ifelse success_a < true_policy_a_reward [set reward_a 1 ;if successful receive reward
    set task_successes (task_successes + 1) ;update counts on task success
    set post_training_successes (post_training_successes + 1) ;update counts on task success
    set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
    set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a  + .000001)) * (reward_a
- value_estimate_a))) ] ;update value estimate for Policy A
   [set reward_a 0 ;if unsuccessful set reward to 0 and update policy value estimate
    set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a + .000001)) * (reward_a
- value_estimate_a))) ;update value estimate for Policy A
   if value_estimate_a < 0 [set value_estimate_a 0]
  set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
   set chose_b 0 ;update choice to Policy A
  ]
  ]
  [let success_b random 100 / 100 ;if Policy B chosen, determine if successful
   ifelse success_b < true_policy_b_reward [set reward_b 1 ;if successful receive reward
    set task_successes (task_successes + 1) ;update counts on task success
    set post_training_successes (post_training_successes + 1) ;update counts on task success
```

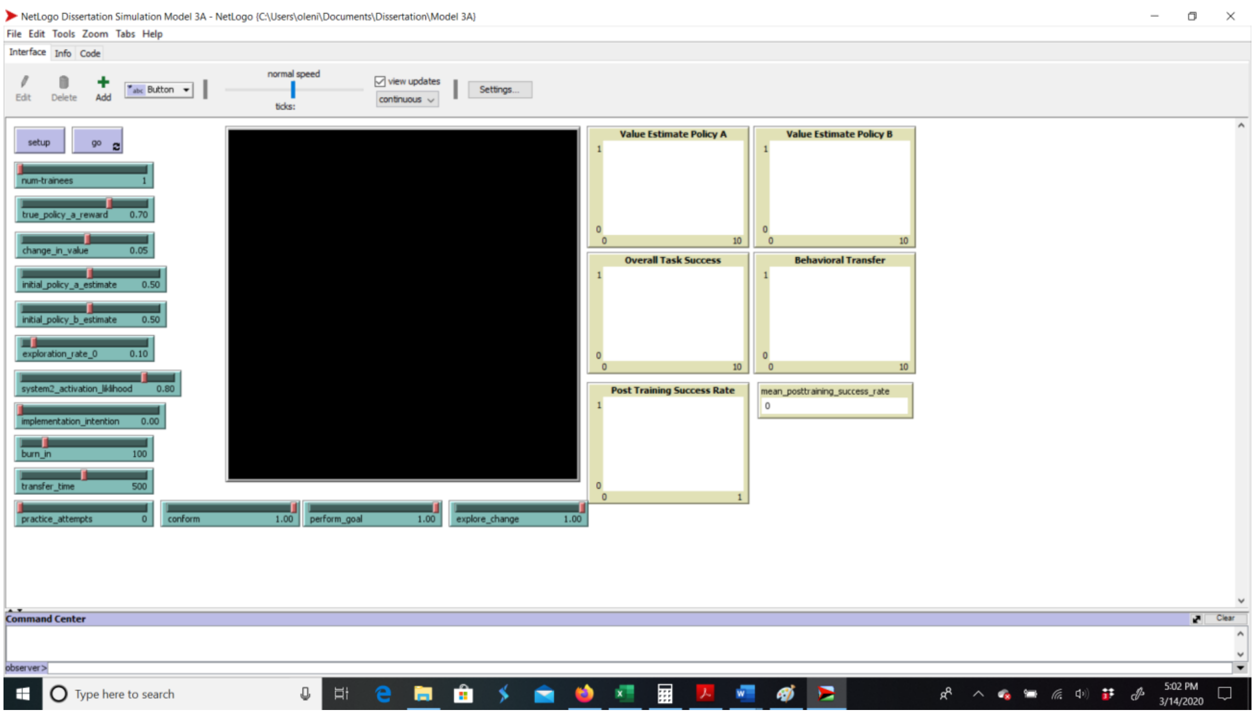
```
        set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
        set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b  + .000001)) *
(reward_b - value_estimate_b))) ] ;update value estimate for Policy B
      [set reward_b 0 ;if unsuccessful set reward to 0 and update policy value estimate
        set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b + .000001)) * (reward_b
- value_estimate_b))) ;update value estimate for Policy B
      if value_estimate_b < 0 [set value_estimate_b 0]
    set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
      set chose_b 1 ;update choice to Policy B
    ]
      ]
end

to run_high_value ;subroutine to choose and execute policy with highest estimated value
   ifelse value_estimate_a >= value_estimate_b [ let success_a random 100 / 100 ;if Policy A
chosen, determine if successful
      ifelse success_a < true_policy_a_reward [set reward_a 1 ;if successful receive reward
        set task_successes (task_successes + 1) ;update counts on task success
        set post_training_successes (post_training_successes + 1) ;update counts on task success
        set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
        set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a  + .000001)) * (reward_a
- value_estimate_a))) ] ;update value estimate for Policy A
      [set reward_a 0 ;if unsuccessful set reward to 0 and update policy value estimate
        set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a + .000001)) * (reward_a
- value_estimate_a))) ;update value estimate for Policy A
      if value_estimate_a < 0 [set value_estimate_a 0]
    set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
      set chose_b 0 ;update choice to Policy A
    ]
    ]
    [let success_b random 100 / 100 ;if Policy B chosen, determine if successful
      ifelse success_b < true_policy_b_reward [set reward_b 1 ;if successful receive reward
        set task_successes (task_successes + 1) ;update counts on task success
        set post_training_successes (post_training_successes + 1) ;update counts on task success
        set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
        set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b  + .000001)) *
(reward_b - value_estimate_b))) ] ;update value estimate for Policy B
      [set reward_b 0 ;if unsuccessful set reward to 0 and update policy value estimate
        set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b + .000001)) * (reward_b
- value_estimate_b))) ;update value estimate for Policy B
      if value_estimate_b < 0 [set value_estimate_b 0]
    set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
      set chose_b 1 ;update choice to Policy B
    ]
      ]
end
```

```
to save-post-training ;save post training performance variables
  ask trainees [set posttraining_success_rate (post_training_successes / (transfer_time +
.000001))]
  ask trainees [set behavioral_transfer_rate (attempts_policy_b / (transfer_time + .000001))]
end

to run-conform ;make conform decision based on specified rate and execute
  let conform_yes random 100 / 100
  ifelse conform_yes <= conform [set conform_choice 1] [set conform_choice 0] ;choose if
conforming or not
  set other_chose_b count other trainees with [chose_b = 1] ;count number of other trainees that
applied b on last step
  let majority_rule other_chose_b / num-trainees
  if conform_choice = 1 [
    ifelse majority_rule < .50 [let success_a random 100 / 100 ;if Policy A chosen, determine if
successful
      ifelse success_a < true_policy_a_reward [set reward_a 1 ;if successful receive reward
        set task_successes (task_successes + 1) ;update counts on task success
        set post_training_successes (post_training_successes + 1) ;update counts on task success
        set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
        set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a  + .000001)) * (reward_a
- value_estimate_a))) ] ;update value estimate for Policy A
      [set reward_a 0 ;if unsuccessful set reward to 0 and update policy value estimate
        set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a + .000001)) * (reward_a
- value_estimate_a))) ;update value estimate for Policy A
      if value_estimate_a < 0 [set value_estimate_a 0]
    set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
      set chose_b 0 ;update choice to Policy A
    ]
    ]
    [let success_b random 100 / 100 ;if Policy B chosen, determine if successful
      ifelse success_b < true_policy_b_reward [set reward_b 1 ;if successful receive reward
        set task_successes (task_successes + 1) ;update counts on task success
        set post_training_successes (post_training_successes + 1) ;update counts on task success
        set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
        set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b  + .000001)) *
(reward_b - value_estimate_b))) ] ;update value estimate for Policy B
      [set reward_b 0 ;if unsuccessful set reward to 0 and update policy value estimate
        set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b + .000001)) * (reward_b
- value_estimate_b))) ;update value estimate for Policy B
      if value_estimate_b < 0 [set value_estimate_b 0]
    set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
      set chose_b 1 ;update choice to Policy B
    ]
      ]
```

```
    ]
end
```

Appendix E: Study 3A Environment and Code

Figure 86. Snapshot of the modeling environment for Model 3A in NetLogo.



Algorithm 17. NetLogo Code for Model 3A

```
 breed [trainees trainee] ;types of agents allowed in environment

trainees-own [
  value_estimate_a ;estimated value of Policy A
  value_estimate_b ;estimated value of Policy B
  system1_choose_a ;liklihood of choosing Policy A as habitual response
  attempts_policy_a ;number times applied Policy A
  attempts_policy_b ;number time applied Policy B
  reward_a ;reward received on most recent attempt with Policy A
  reward_b ;reward received on most recent attempt with Policy B
  task_successes ;number of times successful at task overall
  post_training_successes ;number of times successful only post-training
  pretraining_success_rate ;success rate pretraining only
  posttraining_success_rate ; percentage of times successful in post-training environment
  behavioral_transfer_rate ;rate of choosing Policy B in transfer environment
  transfer_time_count ; ticks into transfer time
  chose_b ;track behavioral choice of last task attempt, 0 = chose a, 1 = chose b
  other_success_rate ;success rate of most successful other trainee
  conform_choice ;track decision to conform on each time step
  other_chose_b ;behavioral choice of most successful other trainee
  goal_difference ;difference between performance goal and actual performance
  j_goal_check ;is the agent short of goal or not?
  exploration_rate ;each trainees have own exploration rate
  ]

globals [
  mean_value_estimate_a ;mean of agent value estimates for Policy A
  mean_value_estimate_b ;mean of agent value estimates for Policy B
  mean_overall_task_success ;task rate of success for full simulation
  mean_pretraining_success_rate ;success rate pretraining only all agents
  mean_posttraining_success_rate ;success rate posttraining only all agents
  mean_behavioral_transfer_rate ;rate of choosing Policy B in transfer environment all agents
  true_policy_b_reward ;reward for Policy B after adjusting for policy value change
]

to setup
  clear-all ;clears environment from previous simulation
  create-trainees num-trainees [ setxy random-xcor random-ycor ;place specified number of
agents at random coordinates
    set value_estimate_a initial_policy_a_estimate ;set initial value estimate for Policy A for each
trainee
    set value_estimate_b initial_policy_b_estimate ;set initial value estimate for Policy B for each
trainee
    set attempts_policy_a 0 ;number times applied Policy A initial set to 0
```

```
    set attempts_policy_b 0 ;number time applied Policy B initial set to 0
    set task_successes 0 ;number of task successes initial set to 0
    set pretraining_success_rate 0 ;success rate pretraining only initial set to 0
    set post_training_successes 0 ;number of successes for post training initial set to 0
    set posttraining_success_rate 0 ;success rate in posttraining environment initial set to 0
    set behavioral_transfer_rate 0 ;percentage of time choosing trained policy initial set to 0
    set chose_b 0 ;set choice tracker to default of Policy A
    set other_success_rate 0 ;setup success rate of most successful other trainee
    set other_chose_b 0 ;setup choice made by other most successful trainee
    set exploration_rate exploration_rate_0 ;set initial exploration rate
  ]
  layout-circle (sort turtles) max-pxcor - 3
  set true_policy_b_reward (true_policy_a_reward + change_in_value)
  if true_policy_b_reward > 1 [set true_policy_b_reward 1]
  if true_policy_b_reward < 0 [set true_policy_b_reward 0]
  reset-ticks ;reset time count to 0
end

to go ;primary subroutines activated
  if ticks = (burn_in + transfer_time) [save-post-training] ;call subroutine to save post training
variables
  if ticks = (burn_in + transfer_time) [stop] ;control length of sim
  tick ;advance time
  if ticks <= burn_in [trainees-burn-in] ;call subroutine to have trainee engage in task during burn
in period
  if ticks > burn_in [trainees-transfer] ;call subroutine for trainee decisions post training
  if ticks = burn_in [save-burn-in] ;call subroutine to save pretraining performance
  update-globals ;call subroutine to calculate all global variables used to track sim functioning
end

to trainees-burn-in ;agents engage in work task during burn in
  ask trainees [let success_a random 100 / 100
    ifelse success_a <= true_policy_a_reward [set reward_a 1
      set task_successes (task_successes + 1)]
      [set reward_a 0]
    set attempts_policy_a (attempts_policy_a + 1)
    set value_estimate_a (value_estimate_a + ((1 / attempts_policy_a) * (reward_a -
value_estimate_a))) ]
end

to update-globals ;calculate all global variables used to track sim functioning
  set mean_value_estimate_a mean [value_estimate_a] of trainees
  set mean_value_estimate_b mean [value_estimate_b] of trainees
  set mean_overall_task_success mean [task_successes] of trainees / ticks
  set mean_pretraining_success_rate mean [pretraining_success_rate] of trainees
  set mean_posttraining_success_rate mean [posttraining_success_rate] of trainees
```

```
  set mean_behavioral_transfer_rate mean [behavioral_transfer_rate] of trainees
end

to trainees-transfer ;call routine to choose which system will drive task and update decision
variables and trackers
  system-choose
  ask trainees [set transfer_time_count (ticks - burn_in)]
  ask trainees [set behavioral_transfer_rate (attempts_policy_b / (transfer_time_count +
.000001))]
  ask trainees [set posttraining_success_rate (post_training_successes / (transfer_time_count +
.000001))]
  ask trainees [set goal_difference (perform_goal - (task_successes / ticks))]
  ask trainees [
    ifelse  goal_difference > 0 [set j_goal_check (1)] [set j_goal_check (0)]
  ]
  ask trainees [set exploration_rate (exploration_rate_0 + (explore_change * j_goal_check))]
end

to system-choose ;decide if system2 will intervene, if not, rely on system 1
  ask trainees [
  let system_choose (random 100 / 100)
  if system_choose < system2_activation_liklihood [system2_decision]
  if system_choose >= system2_activation_liklihood [system1_decision]
  ]
end

to system1_decision ;agent makes automatic decision about which policy to apply
  set system1_choose_a ((attempts_policy_a / (attempts_policy_a + attempts_policy_b +
practice_attempts + .000001)) - implementation_intention) ;update habitual decision rate ;note:
all additions of .000001 are to avoid divisions by 0, number small so as not to affect simulation
  let choose_a random 100 / 100 ;generate random number to determine which policy to
implement
    ifelse choose_a < system1_choose_a [ let success_a random 100 / 100 ;if Policy A chosen,
determine if successful
      ifelse success_a < true_policy_a_reward [set reward_a 1 ;if successful receive reward
        set task_successes (task_successes + 1) ;update counts on task success
        set post_training_successes (post_training_successes + 1) ;update counts on task success
        set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
        set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a  + .000001)) * (reward_a
- value_estimate_a))) ] ;update value estimate for Policy A
      [set reward_a 0 ;if unsuccessful set reward to 0 and update policy value estimate
        set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a + .000001)) * (reward_a
- value_estimate_a))) ;update value estimate for Policy A
    set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
      set chose_b 0 ;update choice to Policy A
    ]
```

```
    ]
    [let success_b random 100 / 100 ;if Policy B chosen, determine if successful
      ifelse success_b < true_policy_b_reward [set reward_b 1 ;if successful receive reward
        set task_successes (task_successes + 1) ;update counts on task success
        set post_training_successes (post_training_successes + 1) ;update counts on task success
        set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
        set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b  + .000001)) *
(reward_b - value_estimate_b))) ] ;update value estimate for Policy B
      [set reward_b 0 ;if unsuccessful set reward to 0 and update policy value estimate
        set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b + .000001)) * (reward_b
- value_estimate_b))) ;update value estimate for Policy B
    set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
      set chose_b 1 ;update choice to Policy B
    ]
      ]
end

to system2_decision ;default to system 2 using highest value estimated policy except at some
error rate
  ifelse num-trainees > 1 [
    run-conform ;have trainee choose if it will conform or not if there are other trainees
    if conform_choice = 0 [let e-greedy random 100 / 100 ;if not imitating run egreedy as normal
      ifelse e-greedy < exploration_rate [ run_low_value ] [ run_high_value ]]
  ]
  [
  let e-greedy random 100 / 100 ;run choice with some degree of error
    ifelse e-greedy < exploration_rate [ run_low_value ] [ run_high_value ]
  ]
end

to save-burn-in ;save pretraining performance
  ask trainees [set pretraining_success_rate (task_successes / (burn_in + .000001))]
end

to run_low_value ;subroutine to choose and execute policy with lowest estimated value
   ifelse value_estimate_a <= value_estimate_b [ let success_a random 100 / 100 ;if Policy A
chosen, determine if successful
      ifelse success_a < true_policy_a_reward [set reward_a 1 ;if successful receive reward
        set task_successes (task_successes + 1) ;update counts on task success
        set post_training_successes (post_training_successes + 1) ;update counts on task success
        set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
        set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a  + .000001)) * (reward_a
- value_estimate_a))) ] ;update value estimate for Policy A
      [set reward_a 0 ;if unsuccessful set reward to 0 and update policy value estimate
        set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a + .000001)) * (reward_a
- value_estimate_a))) ;update value estimate for Policy A
```

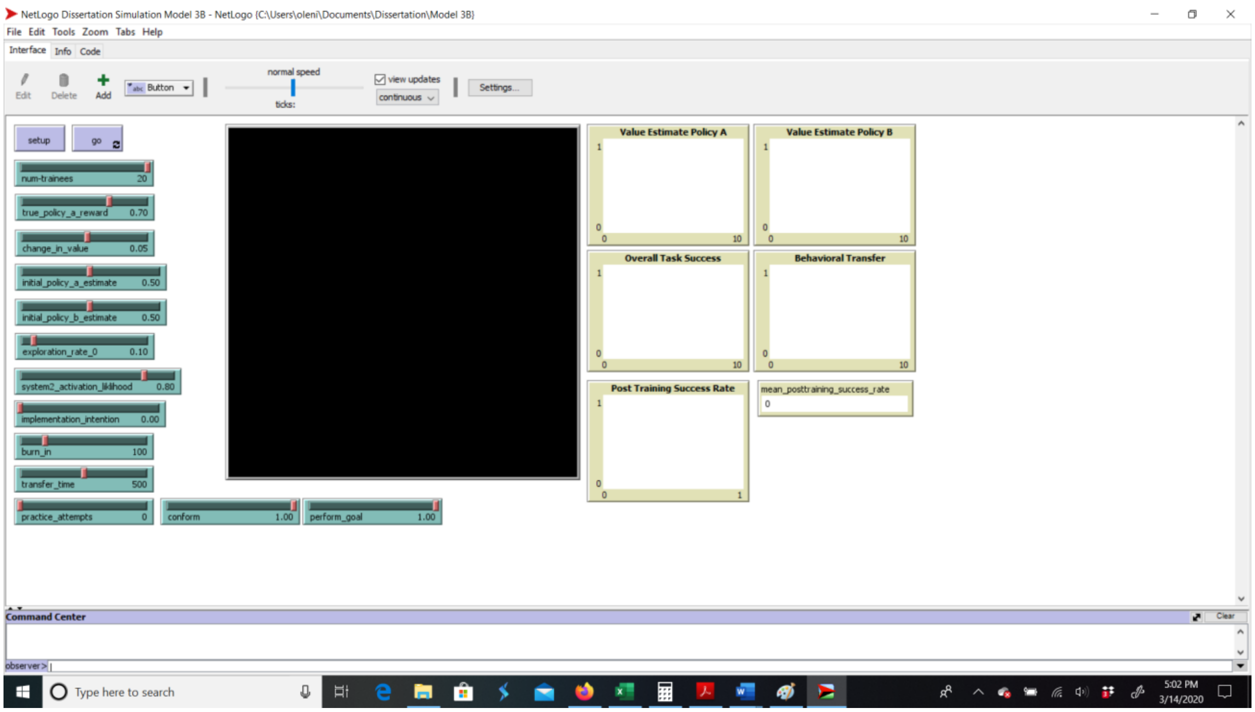
```
    if value_estimate_a < 0 [set value_estimate_a 0]
   set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
    set chose_b 0 ;update choice to Policy A
   ]
   ]
   [let success_b random 100 / 100 ;if Policy B chosen, determine if successful
    ifelse success_b < true_policy_b_reward [set reward_b 1 ;if successful receive reward
      set task_successes (task_successes + 1) ;update counts on task success
      set post_training_successes (post_training_successes + 1) ;update counts on task success
      set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
      set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b  + .000001)) *
(reward_b - value_estimate_b))) ] ;update value estimate for Policy B
    [set reward_b 0 ;if unsuccessful set reward to 0 and update policy value estimate
      set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b + .000001)) * (reward_b
- value_estimate_b))) ;update value estimate for Policy B
    if value_estimate_b < 0 [set value_estimate_b 0]
   set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
    set chose_b 1 ;update choice to Policy B
   ]
    ]
end

to run_high_value ;subroutine to choose and execute policy with highest estimated value
  ifelse value_estimate_a >= value_estimate_b [ let success_a random 100 / 100 ;if Policy A
chosen, determine if successful
    ifelse success_a < true_policy_a_reward [set reward_a 1 ;if successful receive reward
      set task_successes (task_successes + 1) ;update counts on task success
      set post_training_successes (post_training_successes + 1) ;update counts on task success
      set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
      set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a  + .000001)) * (reward_a
- value_estimate_a))) ] ;update value estimate for Policy A
    [set reward_a 0 ;if unsuccessful set reward to 0 and update policy value estimate
      set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a + .000001)) * (reward_a
- value_estimate_a))) ;update value estimate for Policy A
    if value_estimate_a < 0 [set value_estimate_a 0]
   set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
    set chose_b 0 ;update choice to Policy A
   ]
   ]
   [let success_b random 100 / 100 ;if Policy B chosen, determine if successful
    ifelse success_b < true_policy_b_reward [set reward_b 1 ;if successful receive reward
      set task_successes (task_successes + 1) ;update counts on task success
      set post_training_successes (post_training_successes + 1) ;update counts on task success
      set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
      set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b  + .000001)) *
(reward_b - value_estimate_b))) ] ;update value estimate for Policy B
```

```
      [set reward_b 0 ;if unsuccessful set reward to 0 and update policy value estimate
        set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b + .000001)) * (reward_b
- value_estimate_b))) ;update value estimate for Policy B
      if value_estimate_b < 0 [set value_estimate_b 0]
    set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
      set chose_b 1 ;update choice to Policy B
    ]
      ]
end

to save-post-training ;save post training performance variables
  ask trainees [set posttraining_success_rate (post_training_successes / (transfer_time +
.000001))]
  ask trainees [set behavioral_transfer_rate (attempts_policy_b / (transfer_time + .000001))]
end

to run-conform ;make conform decision based on specified rate and execute
  let conform_yes random 100 / 100
  ifelse conform_yes <= conform [set conform_choice 1] [set conform_choice 0] ;choose if
conforming or not
  set other_chose_b count other trainees with [chose_b = 1] ;count number of other trainees that
applied b on last step
  let majority_rule other_chose_b / num-trainees
  if conform_choice = 1 [
    ifelse majority_rule < .50 [let success_a random 100 / 100 ;if Policy A chosen, determine if
successful
      ifelse success_a < true_policy_a_reward [set reward_a 1 ;if successful receive reward
        set task_successes (task_successes + 1) ;update counts on task success
        set post_training_successes (post_training_successes + 1) ;update counts on task success
        set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
        set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a  + .000001)) * (reward_a
- value_estimate_a))) ] ;update value estimate for Policy A
      [set reward_a 0 ;if unsuccessful set reward to 0 and update policy value estimate
        set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a + .000001)) * (reward_a
- value_estimate_a))) ;update value estimate for Policy A
      if value_estimate_a < 0 [set value_estimate_a 0]
    set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
      set chose_b 0 ;update choice to Policy A
    ]
    ]
    [let success_b random 100 / 100 ;if Policy B chosen, determine if successful
      ifelse success_b < true_policy_b_reward [set reward_b 1 ;if successful receive reward
        set task_successes (task_successes + 1) ;update counts on task success
        set post_training_successes (post_training_successes + 1) ;update counts on task success
        set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
```

```
        set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b  + .000001)) *
(reward_b - value_estimate_b))) ] ;update value estimate for Policy B
      [set reward_b 0 ;if unsuccessful set reward to 0 and update policy value estimate
        set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b + .000001)) * (reward_b
- value_estimate_b))) ;update value estimate for Policy B
      if value_estimate_b < 0 [set value_estimate_b 0]
    set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
      set chose_b 1 ;update choice to Policy B
    ]
      ]
      ]
end
```

Appendix F: Studies 3B-1 and 3B-2 Environment and Code

Figure 87. Snapshot of the modeling environment for Models 3B-1 and 3B-2 in NetLogo.



Algorithm 18. NetLogo Code for Model 3B-1

```
breed [trainees trainee] ;types of agents allowed in environment

  trainees-own [
    value_estimate_a ;estimated value of Policy A
    value_estimate_b ;estimated value of Policy B
    system1_choose_a ;liklihood of choosing Policy A as habitual response
    attempts_policy_a ;number times applied Policy A
    attempts_policy_b ;number time applied Policy B
    reward_a ;reward received on most recent attempt with Policy A
    reward_b ;reward received on most recent attempt with Policy B
    task_successes ;number of times successful at task overall
    post_training_successes ;number of times successful only post-training
    pretraining_success_rate ;success rate pretraining only
    posttraining_success_rate ; percentage of times successful in post-training environment
    behavioral_transfer_rate ;rate of choosing Policy B in transfer environment
    transfer_time_count ; ticks into transfer time
    chose_b ;track behavioral choice of last task attempt, 0 = chose a, 1 = chose b
    other_success_rate ;success rate of most successful other trainee
    conform_choice ;track decision to conform on each time step
    other_chose_b ;behavioral choice of most successful other trainee
    goal_difference ;difference between performance goal and actual performance
    j_goal_check ;is the agent short of goal or not?
    exploration_rate ;each trainees have own exploration rate
    ]

  globals [
    mean_value_estimate_a ;mean of agent value estimates for Policy A
    mean_value_estimate_b ;mean of agent value estimates for Policy B
    mean_overall_task_success ;task rate of success for full simulation
    mean_pretraining_success_rate ;success rate pretraining only all agents
    mean_posttraining_success_rate ;success rate posttraining only all agents
    mean_behavioral_transfer_rate ;rate of choosing Policy B in transfer environment all
agents
    true_policy_b_reward ;reward for Policy B after adjusting for policy value change
  ]

  to setup
    clear-all ;clears environment from previous simulation
    create-trainees num-trainees [ setxy random-xcor random-ycor ;place specified number
of agents at random coordinates
      set value_estimate_a initial_policy_a_estimate ;set initial value estimate for Policy A
for each trainee
```

```
    set value_estimate_b initial_policy_b_estimate ;set initial value estimate for Policy B
for each trainee
    set attempts_policy_a 0 ;number times applied Policy A initial set to 0
    set attempts_policy_b 0 ;number time applied Policy B initial set to 0
    set task_successes 0 ;number of task successes initial set to 0
    set pretraining_success_rate 0 ;success rate pretraining only initial set to 0
    set post_training_successes 0 ;number of successes for post training initial set to 0
    set posttraining_success_rate 0 ;success rate in posttraining environment initial set to 0
    set behavioral_transfer_rate 0 ;percentage of time choosing trained policy initial set to
0
    set chose_b 0 ;set choice tracker to default of Policy A
    set other_success_rate 0 ;setup success rate of most successful other trainee
    set other_chose_b 0 ;setup choice made by other most successful trainee
    set exploration_rate exploration_rate_0 ;set initial exploration rate
  ]
  layout-circle (sort turtles) max-pxcor - 3
  set true_policy_b_reward (true_policy_a_reward + change_in_value)
  if true_policy_b_reward > 1 [set true_policy_b_reward 1]
  if true_policy_b_reward < 0 [set true_policy_b_reward 0]
  reset-ticks ;reset time count to 0
end

to go ;primary subroutines activated
  if ticks = (burn_in + transfer_time) [save-post-training] ;call subroutine to save post
training variables
  if ticks = (burn_in + transfer_time) [stop] ;control length of sim
  tick ;advance time
  if ticks <= burn_in [trainees-burn-in] ;call subroutine to have trainee engage in task
during burn in period
  if ticks > burn_in [trainees-transfer] ;call subroutine for trainee decisions post training
  if ticks = burn_in [save-burn-in] ;call subroutine to save pretraining performance
  update-globals ;call subroutine to calculate all global variables used to track sim
functioning
end

to trainees-burn-in ;agents engage in work task during burn in
  ask trainees [let success_a random 100 / 100
    ifelse success_a <= true_policy_a_reward [set reward_a 1
      set task_successes (task_successes + 1)]
      [set reward_a 0]
    set attempts_policy_a (attempts_policy_a + 1)
    set value_estimate_a (value_estimate_a + ((1 / attempts_policy_a) * (reward_a -
value_estimate_a))) ]
end

to update-globals ;calculate all global variables used to track sim functioning
```

```
set mean_value_estimate_a mean [value_estimate_a] of trainees
set mean_value_estimate_b mean [value_estimate_b] of trainees
set mean_overall_task_success mean [task_successes] of trainees / ticks
set mean_pretraining_success_rate mean [pretraining_success_rate] of trainees
set mean_posttraining_success_rate mean [posttraining_success_rate] of trainees
set mean_behavioral_transfer_rate mean [behavioral_transfer_rate] of trainees
end

to trainees-transfer ;call routine to choose which system will drive task and update
decision variables and trackers
  system-choose
  ask trainees [set transfer_time_count (ticks - burn_in)]
  ask trainees [set behavioral_transfer_rate (attempts_policy_b / (transfer_time_count +
.000001))]
  ask trainees [set posttraining_success_rate (post_training_successes /
(transfer_time_count + .000001))]
  ask trainees [set goal_difference (perform_goal - (task_successes / ticks))]
  ask trainees [
    ifelse  goal_difference > 0 [set j_goal_check (1)] [set j_goal_check (0)]
  ]
  ask trainees [set exploration_rate (exploration_rate_0 + goal_difference)]
end

to system-choose ;decide if system2 will intervene, if not, rely on system 1
  ask trainees [
  let system_choose (random 100 / 100)
  if system_choose < system2_activation_liklihood [system2_decision]
  if system_choose >= system2_activation_liklihood [system1_decision]
  ]
end

to system1_decision ;agent makes automatic decision about which policy to apply
  set system1_choose_a ((attempts_policy_a / (attempts_policy_a + attempts_policy_b +
practice_attempts + .000001)) - implementation_intention) ;update habitual decision rate ;note:
all additions of .000001 are to avoid divisions by 0, number small so as not to affect simulation
  let choose_a random 100 / 100 ;generate random number to determine which policy to
implement
    ifelse choose_a < system1_choose_a [ let success_a random 100 / 100 ;if Policy A
chosen, determine if successful
      ifelse success_a < true_policy_a_reward [set reward_a 1 ;if successful receive reward
        set task_successes (task_successes + 1) ;update counts on task success
        set post_training_successes (post_training_successes + 1) ;update counts on task
success
        set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
        set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a  + .000001)) *
(reward_a - value_estimate_a))) ] ;update value estimate for Policy A
```

```
            [set reward_a 0 ;if unsuccessful set reward to 0 and update policy value estimate
              set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a + .000001)) *
(reward_a - value_estimate_a))) ;update value estimate for Policy A
          set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
            set chose_b 0 ;update choice to Policy A
          ]
          ]
          [let success_b random 100 / 100 ;if Policy B chosen, determine if successful
            ifelse success_b < true_policy_b_reward [set reward_b 1 ;if successful receive reward
              set task_successes (task_successes + 1) ;update counts on task success
              set post_training_successes (post_training_successes + 1) ;update counts on task
success
              set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
              set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b  + .000001)) *
(reward_b - value_estimate_b))) ] ;update value estimate for Policy B
            [set reward_b 0 ;if unsuccessful set reward to 0 and update policy value estimate
              set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b + .000001)) *
(reward_b - value_estimate_b))) ;update value estimate for Policy B
          set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
            set chose_b 1 ;update choice to Policy B
          ]
            ]
        end

        to system2_decision ;default to system 2 using highest value estimated policy except at
some error rate
          ifelse num-trainees > 1 [
            run-conform ;have trainee choose if it will conform or not if there are other trainees
            if conform_choice = 0 [let e-greedy random 100 / 100 ;if not imitating run egreedy as
normal
              ifelse e-greedy < exploration_rate [ run_low_value ] [ run_high_value ]]
          ]
          [
          let e-greedy random 100 / 100 ;run choice with some degree of error
            ifelse e-greedy < exploration_rate [ run_low_value ] [ run_high_value ]
          ]
        end

        to save-burn-in ;save pretraining performance
          ask trainees [set pretraining_success_rate (task_successes / (burn_in + .000001))]
        end

        to run_low_value ;subroutine to choose and execute policy with lowest estimated value
          ifelse value_estimate_a <= value_estimate_b [ let success_a random 100 / 100 ;if
Policy A chosen, determine if successful
            ifelse success_a < true_policy_a_reward [set reward_a 1 ;if successful receive reward
```

```
            set task_successes (task_successes + 1) ;update counts on task success
            set post_training_successes (post_training_successes + 1) ;update counts on task
success
            set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
            set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a  + .000001)) *
(reward_a - value_estimate_a))) ] ;update value estimate for Policy A
           [set reward_a 0 ;if unsuccessful set reward to 0 and update policy value estimate
            set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a + .000001)) *
(reward_a - value_estimate_a))) ;update value estimate for Policy A
           if value_estimate_a < 0 [set value_estimate_a 0]
          set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
           set chose_b 0 ;update choice to Policy A
          ]
          ]
          [let success_b random 100 / 100 ;if Policy B chosen, determine if successful
           ifelse success_b < true_policy_b_reward [set reward_b 1 ;if successful receive reward
            set task_successes (task_successes + 1) ;update counts on task success
            set post_training_successes (post_training_successes + 1) ;update counts on task
success
            set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
            set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b  + .000001)) *
(reward_b - value_estimate_b))) ] ;update value estimate for Policy B
           [set reward_b 0 ;if unsuccessful set reward to 0 and update policy value estimate
            set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b + .000001)) *
(reward_b - value_estimate_b))) ;update value estimate for Policy B
           if value_estimate_b < 0 [set value_estimate_b 0]
          set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
           set chose_b 1 ;update choice to Policy B
          ]
           ]
      end

      to run_high_value ;subroutine to choose and execute policy with highest estimated value
         ifelse value_estimate_a >= value_estimate_b [ let success_a random 100 / 100 ;if
Policy A chosen, determine if successful
           ifelse success_a < true_policy_a_reward [set reward_a 1 ;if successful receive reward
            set task_successes (task_successes + 1) ;update counts on task success
            set post_training_successes (post_training_successes + 1) ;update counts on task
success
            set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
            set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a  + .000001)) *
(reward_a - value_estimate_a))) ] ;update value estimate for Policy A
           [set reward_a 0 ;if unsuccessful set reward to 0 and update policy value estimate
            set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a + .000001)) *
(reward_a - value_estimate_a))) ;update value estimate for Policy A
           if value_estimate_a < 0 [set value_estimate_a 0]
```

```
        set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
          set chose_b 0 ;update choice to Policy A
        ]
        ]
        [let success_b random 100 / 100 ;if Policy B chosen, determine if successful
          ifelse success_b < true_policy_b_reward [set reward_b 1 ;if successful receive reward
            set task_successes (task_successes + 1) ;update counts on task success
            set post_training_successes (post_training_successes + 1) ;update counts on task
success
            set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
            set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b  + .000001)) *
(reward_b - value_estimate_b))) ] ;update value estimate for Policy B
          [set reward_b 0 ;if unsuccessful set reward to 0 and update policy value estimate
            set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b + .000001)) *
(reward_b - value_estimate_b))) ;update value estimate for Policy B
          if value_estimate_b < 0 [set value_estimate_b 0]
        set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
          set chose_b 1 ;update choice to Policy B
        ]
          ]
    end

    to save-post-training ;save post training performance variables
      ask trainees [set posttraining_success_rate (post_training_successes / (transfer_time +
.000001))]
      ask trainees [set behavioral_transfer_rate (attempts_policy_b / (transfer_time +
.000001))]
    end

    to run-conform ;make conform decision based on specified rate and execute
      let conform_yes random 100 / 100
      ifelse conform_yes <= conform [set conform_choice 1] [set conform_choice 0] ;choose
if conforming or not
      set other_chose_b count other trainees with [chose_b = 1] ;count number of other
trainees that applied b on last step
      let majority_rule other_chose_b / num-trainees
      if conform_choice = 1 [
        ifelse majority_rule < .50 [let success_a random 100 / 100 ;if Policy A chosen,
determine if successful
          ifelse success_a < true_policy_a_reward [set reward_a 1 ;if successful receive reward
            set task_successes (task_successes + 1) ;update counts on task success
            set post_training_successes (post_training_successes + 1) ;update counts on task
success
            set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
            set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a  + .000001)) *
(reward_a - value_estimate_a))) ] ;update value estimate for Policy A
```

```
              [set reward_a 0 ;if unsuccessful set reward to 0 and update policy value estimate
                set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a + .000001)) *
(reward_a - value_estimate_a))) ;update value estimate for Policy A
              if value_estimate_a < 0 [set value_estimate_a 0]
            set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
              set chose_b 0 ;update choice to Policy A
            ]
            ]
            [let success_b random 100 / 100 ;if Policy B chosen, determine if successful
              ifelse success_b < true_policy_b_reward [set reward_b 1 ;if successful receive reward
                set task_successes (task_successes + 1) ;update counts on task success
                set post_training_successes (post_training_successes + 1) ;update counts on task
success
                set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
                set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b  + .000001)) *
(reward_b - value_estimate_b))) ] ;update value estimate for Policy B
              [set reward_b 0 ;if unsuccessful set reward to 0 and update policy value estimate
                set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b + .000001)) *
(reward_b - value_estimate_b))) ;update value estimate for Policy B
              if value_estimate_b < 0 [set value_estimate_b 0]
            set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
              set chose_b 1 ;update choice to Policy B
            ]
              ]
              ]
end
```

Algorithm 19. NetLogo Code for Model 3B-2

```
breed [trainees trainee] ;types of agents allowed in environment

trainees-own [
  value_estimate_a ;estimated value of Policy A
  value_estimate_b ;estimated value of Policy B
  system1_choose_a ;liklihood of choosing Policy A as habitual response
  attempts_policy_a ;number times applied Policy A
  attempts_policy_b ;number time applied Policy B
  reward_a ;reward received on most recent attempt with Policy A
  reward_b ;reward received on most recent attempt with Policy B
  task_successes ;number of times successful at task overall
  post_training_successes ;number of times successful only post-training
  pretraining_success_rate ;success rate pretraining only
  posttraining_success_rate ; percentage of times successful in post-training environment
  behavioral_transfer_rate ;rate of choosing Policy B in transfer environment
  transfer_time_count ; ticks into transfer time
  chose_b ;track behavioral choice of last task attempt, 0 = chose a, 1 = chose b
  other_success_rate ;success rate of most successful other trainee
  conform_choice ;track decision to conform on each time step
  other_chose_b ;behavioral choice of most successful other trainee
  goal_difference ;difference between performance goal and actual performance
  j_goal_check ;is the agent short of goal or not?
  exploration_rate ;each trainees have own exploration rate
  ]

globals [
  mean_value_estimate_a ;mean of agent value estimates for Policy A
  mean_value_estimate_b ;mean of agent value estimates for Policy B
  mean_overall_task_success ;task rate of success for full simulation
  mean_pretraining_success_rate ;success rate pretraining only all agents
  mean_posttraining_success_rate ;success rate posttraining only all agents
  mean_behavioral_transfer_rate ;rate of choosing Policy B in transfer environment all agents
  true_policy_b_reward ;reward for Policy B after adjusting for policy value change
]

to setup
  clear-all ;clears environment from previous simulation
  create-trainees num-trainees [ setxy random-xcor random-ycor ;place specified number of
agents at random coordinates
    set value_estimate_a initial_policy_a_estimate ;set initial value estimate for Policy A for each
trainee
    set value_estimate_b initial_policy_b_estimate ;set initial value estimate for Policy B for each
trainee
```

```
    set attempts_policy_a 0 ;number times applied Policy A initial set to 0
    set attempts_policy_b 0 ;number time applied Policy B initial set to 0
    set task_successes 0 ;number of task successes initial set to 0
    set pretraining_success_rate 0 ;success rate pretraining only initial set to 0
    set post_training_successes 0 ;number of successes for post training initial set to 0
    set posttraining_success_rate 0 ;success rate in posttraining environment initial set to 0
    set behavioral_transfer_rate 0 ;percentage of time choosing trained policy initial set to 0
    set chose_b 0 ;set choice tracker to default of Policy A
    set other_success_rate 0 ;setup success rate of most successful other trainee
    set other_chose_b 0 ;setup choice made by other most successful trainee
    set exploration_rate exploration_rate_0 ;set initial exploration rate
  ]
  layout-circle (sort turtles) max-pxcor - 3
  set true_policy_b_reward (true_policy_a_reward + change_in_value)
  if true_policy_b_reward > 1 [set true_policy_b_reward 1]
  if true_policy_b_reward < 0 [set true_policy_b_reward 0]
  reset-ticks ;reset time count to 0
end

to go ;primary subroutines activated
  if ticks = (burn_in + transfer_time) [save-post-training] ;call subroutine to save post training
variables
  if ticks = (burn_in + transfer_time) [stop] ;control length of sim
  tick ;advance time
  if ticks <= burn_in [trainees-burn-in] ;call subroutine to have trainee engage in task during burn
in period
  if ticks > burn_in [trainees-transfer] ;call subroutine for trainee decisions post training
  if ticks = burn_in [save-burn-in] ;call subroutine to save pretraining performance
  update-globals ;call subroutine to calculate all global variables used to track sim functioning
end

to trainees-burn-in ;agents engage in work task during burn in
  ask trainees [let success_a random 100 / 100
    ifelse success_a <= true_policy_a_reward [set reward_a 1
      set task_successes (task_successes + 1)]
      [set reward_a 0]
    set attempts_policy_a (attempts_policy_a + 1)
    set value_estimate_a (value_estimate_a + ((1 / attempts_policy_a) * (reward_a -
value_estimate_a))) ]
end

to update-globals ;calculate all global variables used to track sim functioning
  set mean_value_estimate_a mean [value_estimate_a] of trainees
  set mean_value_estimate_b mean [value_estimate_b] of trainees
  set mean_overall_task_success mean [task_successes] of trainees / ticks
  set mean_pretraining_success_rate mean [pretraining_success_rate] of trainees
```

```
  set mean_posttraining_success_rate mean [posttraining_success_rate] of trainees
  set mean_behavioral_transfer_rate mean [behavioral_transfer_rate] of trainees
end

to trainees-transfer ;call routine to choose which system will drive task and update decision
variables and trackers
  system-choose
  ask trainees [set transfer_time_count (ticks - burn_in)]
  ask trainees [set behavioral_transfer_rate (attempts_policy_b / (transfer_time_count +
.000001))]
  ask trainees [set posttraining_success_rate (post_training_successes / (transfer_time_count +
.000001))]
  ask trainees [set goal_difference (perform_goal - (task_successes / ticks))]
  ask trainees [
    ifelse  goal_difference > 0 [set j_goal_check (1)] [set j_goal_check (0)]
  ]
  ask trainees [set exploration_rate (exploration_rate_0 + (.5 - goal_difference))]
end

to system-choose ;decide if system2 will intervene, if not, rely on system 1
  ask trainees [
  let system_choose (random 100 / 100)
  if system_choose < system2_activation_liklihood [system2_decision]
  if system_choose >= system2_activation_liklihood [system1_decision]
  ]
end

to system1_decision ;agent makes automatic decision about which policy to apply
  set system1_choose_a ((attempts_policy_a / (attempts_policy_a + attempts_policy_b +
practice_attempts + .000001)) - implementation_intention) ;update habitual decision rate ;note:
all additions of .000001 are to avoid divisions by 0, number small so as not to affect simulation
  let choose_a random 100 / 100 ;generate random number to determine which policy to
implement
    ifelse choose_a < system1_choose_a [ let success_a random 100 / 100 ;if Policy A chosen,
determine if successful
      ifelse success_a < true_policy_a_reward [set reward_a 1 ;if successful receive reward
        set task_successes (task_successes + 1) ;update counts on task success
        set post_training_successes (post_training_successes + 1) ;update counts on task success
        set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
        set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a  + .000001)) * (reward_a
- value_estimate_a))) ] ;update value estimate for Policy A
      [set reward_a 0 ;if unsuccessful set reward to 0 and update policy value estimate
        set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a + .000001)) * (reward_a
- value_estimate_a))) ;update value estimate for Policy A
    set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
      set chose_b 0 ;update choice to Policy A
```

```
   ]
   ]
   [let success_b random 100 / 100 ;if Policy B chosen, determine if successful
     ifelse success_b < true_policy_b_reward [set reward_b 1 ;if successful receive reward
       set task_successes (task_successes + 1) ;update counts on task success
       set post_training_successes (post_training_successes + 1) ;update counts on task success
       set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
       set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b  + .000001)) *
(reward_b - value_estimate_b))) ] ;update value estimate for Policy B
     [set reward_b 0 ;if unsuccessful set reward to 0 and update policy value estimate
       set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b + .000001)) * (reward_b
- value_estimate_b))) ;update value estimate for Policy B
   set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
     set chose_b 1 ;update choice to Policy B
   ]
     ]
end

to system2_decision ;default to system 2 using highest value estimated policy except at some
error rate
  ifelse num-trainees > 1 [
   run-conform ;have trainee choose if it will conform or not if there are other trainees
   if conform_choice = 0 [let e-greedy random 100 / 100 ;if not imitating run egreedy as normal
     ifelse e-greedy < exploration_rate [ run_low_value ] [ run_high_value ]]
  ]
  [
  let e-greedy random 100 / 100 ;run choice with some degree of error
   ifelse e-greedy < exploration_rate [ run_low_value ] [ run_high_value ]
  ]
end

to save-burn-in ;save pretraining performance
  ask trainees [set pretraining_success_rate (task_successes / (burn_in + .000001))]
end

to run_low_value ;subroutine to choose and execute policy with lowest estimated value
   ifelse value_estimate_a <= value_estimate_b [ let success_a random 100 / 100 ;if Policy A
chosen, determine if successful
     ifelse success_a < true_policy_a_reward [set reward_a 1 ;if successful receive reward
       set task_successes (task_successes + 1) ;update counts on task success
       set post_training_successes (post_training_successes + 1) ;update counts on task success
       set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
       set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a  + .000001)) * (reward_a
- value_estimate_a))) ] ;update value estimate for Policy A
     [set reward_a 0 ;if unsuccessful set reward to 0 and update policy value estimate
```

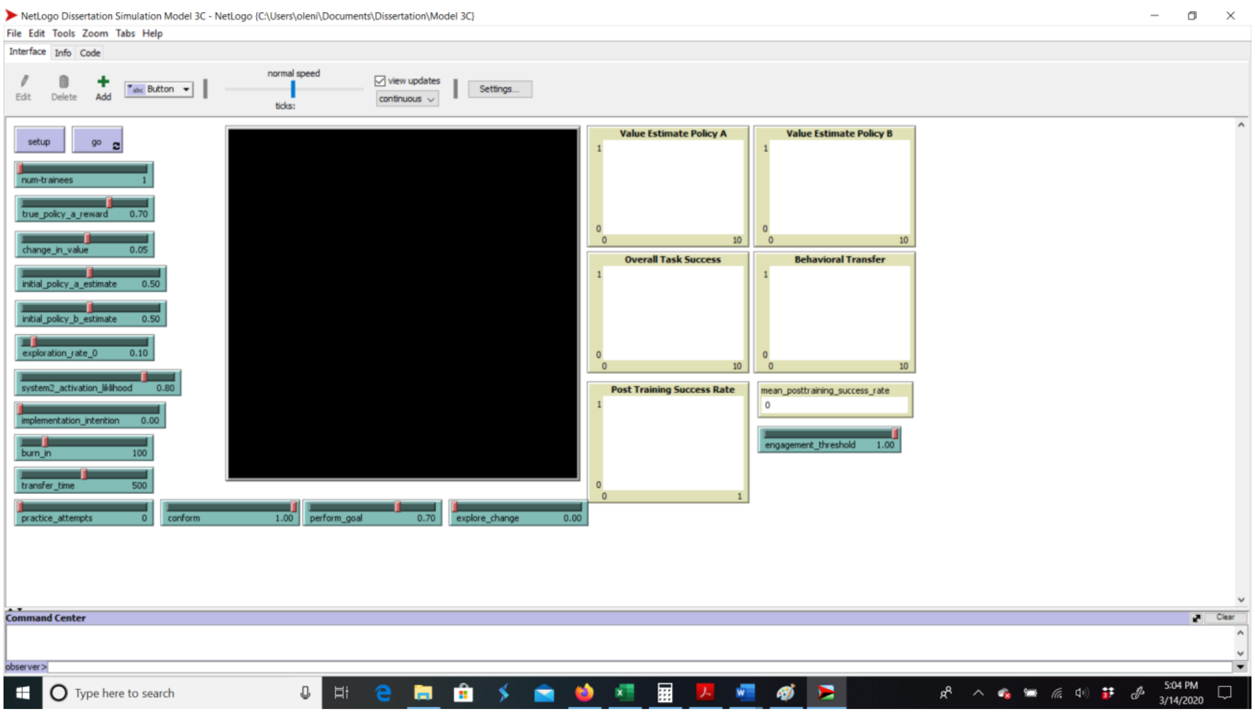
```
    set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a + .000001)) * (reward_a
- value_estimate_a))) ;update value estimate for Policy A
   if value_estimate_a < 0 [set value_estimate_a 0]
  set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
   set chose_b 0 ;update choice to Policy A
  ]
  ]
  [let success_b random 100 / 100 ;if Policy B chosen, determine if successful
   ifelse success_b < true_policy_b_reward [set reward_b 1 ;if successful receive reward
    set task_successes (task_successes + 1) ;update counts on task success
    set post_training_successes (post_training_successes + 1) ;update counts on task success
    set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
    set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b  + .000001)) *
(reward_b - value_estimate_b))) ] ;update value estimate for Policy B
   [set reward_b 0 ;if unsuccessful set reward to 0 and update policy value estimate
    set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b + .000001)) * (reward_b
- value_estimate_b))) ;update value estimate for Policy B
   if value_estimate_b < 0 [set value_estimate_b 0]
  set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
   set chose_b 1 ;update choice to Policy B
  ]
   ]
end

to run_high_value ;subroutine to choose and execute policy with highest estimated value
 ifelse value_estimate_a >= value_estimate_b [ let success_a random 100 / 100 ;if Policy A
chosen, determine if successful
   ifelse success_a < true_policy_a_reward [set reward_a 1 ;if successful receive reward
    set task_successes (task_successes + 1) ;update counts on task success
    set post_training_successes (post_training_successes + 1) ;update counts on task success
    set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
    set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a  + .000001)) * (reward_a
- value_estimate_a))) ] ;update value estimate for Policy A
   [set reward_a 0 ;if unsuccessful set reward to 0 and update policy value estimate
    set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a + .000001)) * (reward_a
- value_estimate_a))) ;update value estimate for Policy A
   if value_estimate_a < 0 [set value_estimate_a 0]
  set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
   set chose_b 0 ;update choice to Policy A
  ]
  ]
  [let success_b random 100 / 100 ;if Policy B chosen, determine if successful
   ifelse success_b < true_policy_b_reward [set reward_b 1 ;if successful receive reward
    set task_successes (task_successes + 1) ;update counts on task success
    set post_training_successes (post_training_successes + 1) ;update counts on task success
    set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
```

```
      set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b  + .000001)) *
(reward_b - value_estimate_b))) ] ;update value estimate for Policy B
     [set reward_b 0 ;if unsuccessful set reward to 0 and update policy value estimate
      set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b + .000001)) * (reward_b
- value_estimate_b))) ;update value estimate for Policy B
     if value_estimate_b < 0 [set value_estimate_b 0]
    set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
     set chose_b 1 ;update choice to Policy B
    ]
     ]
end

to save-post-training ;save post training performance variables
  ask trainees [set posttraining_success_rate (post_training_successes / (transfer_time +
.000001))]
  ask trainees [set behavioral_transfer_rate (attempts_policy_b / (transfer_time + .000001))]
end

to run-conform ;make conform decision based on specified rate and execute
  let conform_yes random 100 / 100
  ifelse conform_yes <= conform [set conform_choice 1] [set conform_choice 0] ;choose if
conforming or not
  set other_chose_b count other trainees with [chose_b = 1] ;count number of other trainees that
applied b on last step
  let majority_rule other_chose_b / num-trainees
  if conform_choice = 1 [
    ifelse majority_rule < .50 [let success_a random 100 / 100 ;if Policy A chosen, determine if
successful
     ifelse success_a < true_policy_a_reward [set reward_a 1 ;if successful receive reward
      set task_successes (task_successes + 1) ;update counts on task success
      set post_training_successes (post_training_successes + 1) ;update counts on task success
      set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
      set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a  + .000001)) * (reward_a
- value_estimate_a))) ] ;update value estimate for Policy A
     [set reward_a 0 ;if unsuccessful set reward to 0 and update policy value estimate
      set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a + .000001)) * (reward_a
- value_estimate_a))) ;update value estimate for Policy A
     if value_estimate_a < 0 [set value_estimate_a 0]
    set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
     set chose_b 0 ;update choice to Policy A
    ]
    ]
    [let success_b random 100 / 100 ;if Policy B chosen, determine if successful
     ifelse success_b < true_policy_b_reward [set reward_b 1 ;if successful receive reward
      set task_successes (task_successes + 1) ;update counts on task success
      set post_training_successes (post_training_successes + 1) ;update counts on task success
```

```
       set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
       set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b  + .000001)) *
(reward_b - value_estimate_b))) ] ;update value estimate for Policy B
     [set reward_b 0 ;if unsuccessful set reward to 0 and update policy value estimate
       set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b + .000001)) * (reward_b
- value_estimate_b))) ;update value estimate for Policy B
     if value_estimate_b < 0 [set value_estimate_b 0]
    set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
     set chose_b 1 ;update choice to Policy B
    ]
      ]
      ]
end
```

Appendix G: Study 3C Environment and Code

Figure 88. Snapshot of the modeling environment for Model 3C in NetLogo.



Algorithm 20. NetLogo Code for Model 3C

```
breed [trainees trainee] ;types of agents allowed in environment

trainees-own [
  value_estimate_a ;estimated value of Policy A
  value_estimate_b ;estimated value of Policy B
  system1_choose_a ;liklihood of choosing Policy A as habitual response
  attempts_policy_a ;number times applied Policy A
  attempts_policy_b ;number time applied Policy B
  reward_a ;reward received on most recent attempt with Policy A
  reward_b ;reward received on most recent attempt with Policy B
  task_successes ;number of times successful at task overall
  post_training_successes ;number of times successful only post-training
  pretraining_success_rate ;success rate pretraining only
  posttraining_success_rate ; percentage of times successful in post-training environment
  behavioral_transfer_rate ;rate of choosing Policy B in transfer environment
  transfer_time_count ; ticks into transfer time
  chose_b ;track behavioral choice of last task attempt, 0 = chose a, 1 = chose b
  other_success_rate ;success rate of most successful other trainee
  conform_choice ;track decision to conform on each time step
  other_chose_b ;behavioral choice of most successful other trainee
  goal_difference ;difference between performance goal and actual performance
  j_goal_check ;is the agent short of goal or not?
  exploration_rate ;each trainees have own exploration rate
  ]

globals [
  mean_value_estimate_a ;mean of agent value estimates for Policy A
  mean_value_estimate_b ;mean of agent value estimates for Policy B
  mean_overall_task_success ;task rate of success for full simulation
  mean_pretraining_success_rate ;success rate pretraining only all agents
  mean_posttraining_success_rate ;success rate posttraining only all agents
  mean_behavioral_transfer_rate ;rate of choosing Policy B in transfer environment all agents
  true_policy_b_reward ;reward for Policy B after adjusting for policy value change
]

to setup
  clear-all ;clears environment from previous simulation
  create-trainees num-trainees [ setxy random-xcor random-ycor ;place specified number of
agents at random coordinates
    set value_estimate_a initial_policy_a_estimate ;set initial value estimate for Policy A for each
trainee
    set value_estimate_b initial_policy_b_estimate ;set initial value estimate for Policy B for each
trainee
    set attempts_policy_a 0 ;number times applied Policy A initial set to 0
```

```
    set attempts_policy_b 0 ;number time applied Policy B initial set to 0
    set task_successes 0 ;number of task successes initial set to 0
    set pretraining_success_rate 0 ;success rate pretraining only initial set to 0
    set post_training_successes 0 ;number of successes for post training initial set to 0
    set posttraining_success_rate 0 ;success rate in posttraining environment initial set to 0
    set behavioral_transfer_rate 0 ;percentage of time choosing trained policy initial set to 0
    set chose_b 0 ;set choice tracker to default of Policy A
    set other_success_rate 0 ;setup success rate of most successful other trainee
    set other_chose_b 0 ;setup choice made by other most successful trainee
    set exploration_rate exploration_rate_0 ;set initial exploration rate
  ]
  layout-circle (sort turtles) max-pxcor - 3
  set true_policy_b_reward (true_policy_a_reward + change_in_value)
  if true_policy_b_reward > 1 [set true_policy_b_reward 1]
  if true_policy_b_reward < 0 [set true_policy_b_reward 0]
  reset-ticks ;reset time count to 0
end

to go ;primary subroutines activated
  if ticks = (burn_in + transfer_time) [save-post-training] ;call subroutine to save post training
variables
  if ticks = (burn_in + transfer_time) [stop] ;control length of sim
  tick ;advance time
  if ticks <= burn_in [trainees-burn-in] ;call subroutine to have trainee engage in task during burn
in period
  if ticks > burn_in [trainees-transfer] ;call subroutine for trainee decisions post training
  if ticks = burn_in [save-burn-in] ;call subroutine to save pretraining performance
  update-globals ;call subroutine to calculate all global variables used to track sim functioning
end

to trainees-burn-in ;agents engage in work task during burn in
  ask trainees [let success_a random 100 / 100
    ifelse success_a <= true_policy_a_reward [set reward_a 1
      set task_successes (task_successes + 1)]
      [set reward_a 0]
    set attempts_policy_a (attempts_policy_a + 1)
    set value_estimate_a (value_estimate_a + ((1 / attempts_policy_a) * (reward_a -
value_estimate_a))) ]
end

to update-globals ;calculate all global variables used to track sim functioning
  set mean_value_estimate_a mean [value_estimate_a] of trainees
  set mean_value_estimate_b mean [value_estimate_b] of trainees
  set mean_overall_task_success mean [task_successes] of trainees / ticks
  set mean_pretraining_success_rate mean [pretraining_success_rate] of trainees
  set mean_posttraining_success_rate mean [posttraining_success_rate] of trainees
```

```
  set mean_behavioral_transfer_rate mean [behavioral_transfer_rate] of trainees
end

to trainees-transfer ;call routine to choose which system will drive task and update decision
variables and trackers
  system-choose
  ask trainees [set transfer_time_count (ticks - burn_in)]
  ask trainees [set behavioral_transfer_rate (attempts_policy_b / (transfer_time_count +
.000001))]
  ask trainees [set posttraining_success_rate (post_training_successes / (transfer_time_count +
.000001))]
  ask trainees [set goal_difference (perform_goal - (task_successes / ticks))]
  ask trainees [
    ifelse  goal_difference > 0 [set j_goal_check (1)] [set j_goal_check (0)]
  ]
  ask trainees [set exploration_rate (exploration_rate_0 + (explore_change * j_goal_check))]
end

to system-choose ;decide if system2 will intervene, if not, rely on system 1
  ask trainees [
    ifelse value_estimate_b < engagement_threshold [run_policy_a]
    [
  let system_choose (random 100 / 100)
  ifelse system_choose < system2_activation_liklihood [system2_decision] [system1_decision]
    ]
  ]
end

to system1_decision ;agent makes automatic decision about which policy to apply
  set system1_choose_a ((attempts_policy_a / (attempts_policy_a + attempts_policy_b +
practice_attempts + .000001)) - implementation_intention) ;update habitual decision rate ;note:
all additions of .000001 are to avoid divisions by 0, number small so as not to affect simulation
  let choose_a random 100 / 100 ;generate random number to determine which policy to
implement
    ifelse choose_a < system1_choose_a [ let success_a random 100 / 100 ;if Policy A chosen,
determine if successful
      ifelse success_a < true_policy_a_reward [set reward_a 1 ;if successful receive reward
        set task_successes (task_successes + 1) ;update counts on task success
        set post_training_successes (post_training_successes + 1) ;update counts on task success
        set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
        set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a  + .000001)) * (reward_a
- value_estimate_a))) ] ;update value estimate for Policy A
      [set reward_a 0 ;if unsuccessful set reward to 0 and update policy value estimate
        set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a + .000001)) * (reward_a
- value_estimate_a))) ;update value estimate for Policy A
    set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
```

```
      set chose_b 0 ;update choice to Policy A
    ]
    ]
    [let success_b random 100 / 100 ;if Policy B chosen, determine if successful
      ifelse success_b < true_policy_b_reward [set reward_b 1 ;if successful receive reward
        set task_successes (task_successes + 1) ;update counts on task success
        set post_training_successes (post_training_successes + 1) ;update counts on task success
        set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
        set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b  + .000001)) *
(reward_b - value_estimate_b))) ] ;update value estimate for Policy B
      [set reward_b 0 ;if unsuccessful set reward to 0 and update policy value estimate
        set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b + .000001)) * (reward_b
- value_estimate_b))) ;update value estimate for Policy B
    set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
      set chose_b 1 ;update choice to Policy B
    ]
      ]
end

to system2_decision ;default to system 2 using highest value estimated policy except at some
error rate
  ifelse num-trainees > 1 [
    run-conform ;have trainee choose if it will conform or not if there are other trainees
    if conform_choice = 0 [let e-greedy random 100 / 100 ;if not imitating run egreedy as normal
      ifelse e-greedy < exploration_rate [ run_low_value ] [ run_high_value ]]
  ]
  [
  let e-greedy random 100 / 100 ;run choice with some degree of error
    ifelse e-greedy < exploration_rate [ run_low_value ] [ run_high_value ]
  ]
end

to save-burn-in ;save pretraining performance
  ask trainees [set pretraining_success_rate (task_successes / (burn_in + .000001))]
end

to run_low_value ;subroutine to choose and execute policy with lowest estimated value
   ifelse value_estimate_a <= value_estimate_b [ let success_a random 100 / 100 ;if Policy A
chosen, determine if successful
      ifelse success_a < true_policy_a_reward [set reward_a 1 ;if successful receive reward
        set task_successes (task_successes + 1) ;update counts on task success
        set post_training_successes (post_training_successes + 1) ;update counts on task success
        set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
        set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a  + .000001)) * (reward_a
- value_estimate_a))) ] ;update value estimate for Policy A
      [set reward_a 0 ;if unsuccessful set reward to 0 and update policy value estimate
```

```
        set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a + .000001)) * (reward_a
- value_estimate_a))) ;update value estimate for Policy A
      if value_estimate_a < 0 [set value_estimate_a 0]
    set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
      set chose_b 0 ;update choice to Policy A
    ]
    ]
    [let success_b random 100 / 100 ;if Policy B chosen, determine if successful
      ifelse success_b < true_policy_b_reward [set reward_b 1 ;if successful receive reward
        set task_successes (task_successes + 1) ;update counts on task success
        set post_training_successes (post_training_successes + 1) ;update counts on task success
        set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
        set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b  + .000001)) *
(reward_b - value_estimate_b))) ] ;update value estimate for Policy B
      [set reward_b 0 ;if unsuccessful set reward to 0 and update policy value estimate
        set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b + .000001)) * (reward_b
- value_estimate_b))) ;update value estimate for Policy B
      if value_estimate_b < 0 [set value_estimate_b 0]
    set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
      set chose_b 1 ;update choice to Policy B
    ]
      ]
end

to run_high_value ;subroutine to choose and execute policy with highest estimated value
   ifelse value_estimate_a >= value_estimate_b [ let success_a random 100 / 100 ;if Policy A
chosen, determine if successful
      ifelse success_a < true_policy_a_reward [set reward_a 1 ;if successful receive reward
        set task_successes (task_successes + 1) ;update counts on task success
        set post_training_successes (post_training_successes + 1) ;update counts on task success
        set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
        set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a  + .000001)) * (reward_a
- value_estimate_a))) ] ;update value estimate for Policy A
      [set reward_a 0 ;if unsuccessful set reward to 0 and update policy value estimate
        set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a + .000001)) * (reward_a
- value_estimate_a))) ;update value estimate for Policy A
      if value_estimate_a < 0 [set value_estimate_a 0]
    set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
      set chose_b 0 ;update choice to Policy A
    ]
    ]
    [let success_b random 100 / 100 ;if Policy B chosen, determine if successful
      ifelse success_b < true_policy_b_reward [set reward_b 1 ;if successful receive reward
        set task_successes (task_successes + 1) ;update counts on task success
        set post_training_successes (post_training_successes + 1) ;update counts on task success
        set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
```

```
    set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b  + .000001)) *
(reward_b - value_estimate_b))) ] ;update value estimate for Policy B
   [set reward_b 0 ;if unsuccessful set reward to 0 and update policy value estimate
    set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b + .000001)) * (reward_b
- value_estimate_b))) ;update value estimate for Policy B
   if value_estimate_b < 0 [set value_estimate_b 0]
  set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
   set chose_b 1 ;update choice to Policy B
  ]
   ]
end

to save-post-training ;save post training performance variables
 ask trainees [set posttraining_success_rate (post_training_successes / (transfer_time +
.000001))]
 ask trainees [set behavioral_transfer_rate (attempts_policy_b / (transfer_time + .000001))]
end

to run-conform ;make conform decision based on specified rate and execute
 let conform_yes random 100 / 100
 ifelse conform_yes <= conform [set conform_choice 1] [set conform_choice 0] ;choose if
conforming or not
 set other_chose_b count other trainees with [chose_b = 1] ;count number of other trainees that
applied b on last step
 let majority_rule other_chose_b / num-trainees
 if conform_choice = 1 [
  ifelse majority_rule < .50 [let success_a random 100 / 100 ;if Policy A chosen, determine if
successful
   ifelse success_a < true_policy_a_reward [set reward_a 1 ;if successful receive reward
    set task_successes (task_successes + 1) ;update counts on task success
    set post_training_successes (post_training_successes + 1) ;update counts on task success
    set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
    set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a  + .000001)) * (reward_a
- value_estimate_a))) ] ;update value estimate for Policy A
   [set reward_a 0 ;if unsuccessful set reward to 0 and update policy value estimate
    set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a + .000001)) * (reward_a
- value_estimate_a))) ;update value estimate for Policy A
   if value_estimate_a < 0 [set value_estimate_a 0]
  set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
   set chose_b 0 ;update choice to Policy A
  ]
  ]
  [let success_b random 100 / 100 ;if Policy B chosen, determine if successful
   ifelse success_b < true_policy_b_reward [set reward_b 1 ;if successful receive reward
    set task_successes (task_successes + 1) ;update counts on task success
    set post_training_successes (post_training_successes + 1) ;update counts on task success
```

```
       set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
       set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b  + .000001)) *
(reward_b - value_estimate_b))) ] ;update value estimate for Policy B
     [set reward_b 0 ;if unsuccessful set reward to 0 and update policy value estimate
       set value_estimate_b (value_estimate_b + ((1 / (attempts_policy_b + .000001)) * (reward_b
- value_estimate_b))) ;update value estimate for Policy B
     if value_estimate_b < 0 [set value_estimate_b 0]
    set attempts_policy_b attempts_policy_b + 1 ;update count on Policy B choice
     set chose_b 1 ;update choice to Policy B
    ]
     ]
     ]
end

to run_policy_a
  let success_a random 100 / 100 ;if Policy A chosen, determine if successful
     ifelse success_a < true_policy_a_reward [set reward_a 1 ;if successful receive reward
       set task_successes (task_successes + 1) ;update counts on task success
       set post_training_successes (post_training_successes + 1) ;update counts on task success
       set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
       set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a  + .000001)) * (reward_a
- value_estimate_a))) ] ;update value estimate for Policy A
     [set reward_a 0 ;if unsuccessful set reward to 0 and update policy value estimate
       set value_estimate_a (value_estimate_a + ((1 / (attempts_policy_a + .000001)) * (reward_a
- value_estimate_a))) ;update value estimate for Policy A
     if value_estimate_a < 0 [set value_estimate_a 0]
    set attempts_policy_a attempts_policy_a + 1 ;update count on Policy A choice
     set chose_b 0 ;update choice to Policy A
    ]
end
```

REFERENCES

REFERENCES

American Society of Training and Development. (2018). *2018 state of the industry report.* Alexandria, VA: ASTD Press.

Arthur, W., Bennett, W., Stanush, P., L., & McNelly, T. L. (1998). Factors that influence skill decay and retention: A quantitative review and analysis. *Human Performance*, *11*(1), 57-101.

Baard, S. K., Rench, T. A., & Kozlowski, S. W. J. (2014). Performance adaptation: A theoretical integration and review. *Journal of Management, 40*(1), 48-99.

Bago, B. and De Neys, W. (2017) Fast logic? Examining the time course assumption of dual process theory. *Cognition, 158,* 90–109

Baker, J. S., & Frey, P. W. (1980). A cusp catastrophe: Hysterisis, bimodality, and inaccessibility in rabbit eyelid conditioning. *Learning and Motivation, 10,* 520-535.

Baldwin, T., & Ford, J. K. (1988). Transfer of training: A review and directions for future research. *Personnel Psychology, 41*(1), 63-105.

Baldwin, T., Ford, J. K., & Blume, B. (2009). Transfer of training 1988-2008: An updated review and agenda for future research. *International Review of Industrial and Organizational Psychology, 24,* 41-70.

Baldwin, T., Magjuka, R. J., & Loher, B. (1991). The perils of participation: Effects of choice of training on trainee motivation and learning. *Personnel Psychology, 44,* 51-65.

Bandura, A. (1977). *Social Learning Theory*. Oxford, England: Prentice-Hall.

Bandura, A. (1989). Human agency in Social Cognitive Theory. *American Psychologist, 44*(9), 1175-1184).

Bandura, A. (1991). Social Cognitive Theory of Self-Regulation. *Organizational Behavior and Human Decision Processes, 50*, 248-287.

Bandura, A., & Cervone, D. (1983). Self-evaluative and self-efficacy mechanisms governing the motivational effects of goal systems. *Journal of Personality and Social Psychology, 45*(5), 1017-1028.

Banks, J., Carson, I. I., Nelson, B. L., & Nicol, D. M. (2005). *Discrete-event system simulation*. Pearson.

Bauer, K. N., Orvis, K. A., Ely, K., & Surface, E. A. (2016). Re-examination of motivation in learning contexts: Meta-analytically investigating the role type of motivation plays in the prediction of key training outcomes. *Journal of Business and Psychology, 31*(1), 33-50.

Beier, M. E., & Kanfer, R. (2010). Motivation in training and development: A phase perspective. In. S. W. J. Kozlowski & E. Salas (Eds.), *Learning, training, and development in organizations* (pp. 65-97). New York, NY: Routledge.

Bell, B., & Kozlowski, S. W. J. (2008). Active learning: Effects of core training design elements on self-regulatory processes, learning, and adaptability. *Journal of Applied Psychology, 93*(2), 296-316.

Bell, B., & Kozlowski, S. W. J. (2010). Toward a theory of learner-centered training design: An integrative framework of active learning. In. S. W. J. Kozlowski & E. Salas (Eds.), *Learning, training, and development in organizations* (pp. 263-300). New York, NY: Routledge.

Bell, B., Tannenbaum, S., Ford, J. K., Noe, R., & Kraiger, K. (2017). 100 years of training and development research: What we know and where we should go. *Journal of Applied Psychology, 102*(3), 305-323.

Benner, P. (1982). From novice to expert. *American Journal of Nursing, 82*(3), 402-407.

Blume, B., Ford, J. K., Baldwin, T., & Huang, J. (2010). Transfer of training: A meta-analytic review. *Journal of Management, 36*(4), 1065-1105.

Blume, B., Ford, J. K., Surface, E., & Olenick, J. (2019). A dynamic model of training transfer. *Human Resource Management Review, 29,* 270-283.

Box, G. E. P. (1976). Science and statistics. *Journal of the American Statistical Association, 71*(356), 791-799.

Brauer, M., Wasel, W., & Niedenthal, P. (2000). Implicit and explicit components of prejudice. *Review of General Psychology, 4,* 79-101.

Bronfenbrenner, U. (1977). Toward an experimental ecology of human development. *American Psychologist, 32,* 513–531.

Bronfenbrenner, U. (1979). *The ecology of human development: Experiments by nature and design.* Cambridge, MA: Harvard University Press.

Campion, M. & Lord, R. (1982). A Control-Syestems conceptualization of the goal-setting and changing process. *Organizational Behavior and Human Performance, 30,* 265-287.

Cannon-Bowers, J. A., & Salas, E., Tannenbaum, S. I., & Mathieu, J. E. (1995). Toward theoretically based principles of training effectiveness: A model and initial empirical investigation. *Military Psychology, 7*(3), 141-164.

Carver, C., & Scheier, M. (1998). On the self-regulation of behavior. New York, NY: Cambridge University Press.

Cascio, W. F. (2019). Training trends: Macro, micro, and policy issues. *Human Resource Management Review, 29,* 284-297.

Chen, G., Thomas, B., & Wallace, J. C. (2005). A multilevel examination of the relationships among training outcomes, mediating regulatory processes, and adaptive. *Journal of Applied Psychology*, *90*(5), 827-841.

Cheng, E. (2016). Maintaining the transfer of in-service teachers' training in the workplace. *Educational Psychology, 36*(3), 444-460.

Cheng, E., & Hampson, I. (2008). Transfer of training: A review and new insights. *International Journal of Management Reviews, 10*(4), 327-341.

Clark, F., Sanders, K., Carlson, M. Blanche, E., & Jackson, J. (2007). Synthesis of habit theory. *OTJR: Occupation, Participation and Health, 27,* 75-235.

Colquitt, J. A., LePine, J. A., & Noe, R. A. (2000). Toward an integrative theory of training motivation: A meta-analytic path analysis of 20 years of research. *Journal of Applied Psychology, 85*(3), 679-707.

Cumming, G. (2014). The new statistics: Why and how. *Psychological Science*, *25*(1), 7-29.

DeShon, R. P. (2012). Multivariate dynamics in organizational science. In S. W. J. Kozlowski (Ed.), *The Oxford Handbook of Organizational Psychology, Vol. 1* (pp. 117-142). New York, NY: Oxford University Press.

DeShon, R. P., & Rench, T. A. (2009). Clarifying the notion of self-regulation in organizational behavior. *International Review of Industrial and Organizational Psychology, 24,* 217-247.

Dickinson, A. (1980). *Contemporary Animal Learning Theory.* Cambridge University Press.

Dickinson, A. (1985). Actions and habits: The development of behavioral autonomy. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 67-78.

Dienes, Z. (2019). How do I know what my theory predicts? *Advances in Methods and Practices in Psychological Science, 2*(4), 364-377.

Dienes, Z., & Perner, J. (1999). A theory of implicit and explicit knowledge. *Behavioral and Brain Sciences, 22,* 735-808.

Dierdorff, E., & Surface, E. (2008). If you pay for skills, will they learn? Skill change and maintenance under a skill-based pay system. *Journal of Management, 34*(4), 721-743.

Dishop, C., Olenick, J., & DeShon, R. (in press). Principles for Taking a Dynamic Perspective. In Y. Griep, S. D. Hansen, T. Vantilborgh, and J. Hofmans (Eds.), *Handbook of*

Temporal Dynamic Organizational Behavior, Vol. 1: A Dynamic Look At Organizational Behavior Topics. Edward Elgar.

Donovan, J. & Radosevich, D. (1999). A meta-analytic review of the distribution of practice effect: Now you see it, now you don't. *Journal of Applied Psychology, 84*, 795-805.

Driskell, J. E., Willis, R. P., & Copper, C. (1992). Effect of overlearning on retention. *Journal of Applied Psychology, 77*(5), 615-622.

Dunlosky, J., Rawson, K. A., Mash, E. J., Nathan, M. J., & Willingham, D. T. (2013). Improving students' learning with effective learning techniques: Promising directions from cognitive and educational psychology. *Psychological Science in the Public Interest, 14*(1), 4-58.

Dweck, C. (1986). Motivational processes affecting learning. *American Psychologist, 41*(10), 1040-1048.

Elliot, A. (2006). The hierarchical model of approach-avoidance orientation. *Motivation and Emotion, 30*, 111-116.

Epstein, J.M. (1999). Agent-based computational models and generative social science. *Complexity, 4*(5), 41-60.

Ericsson, K. A. (2006). The influence of experience and deliberate practice on the development of superior expert performance. In K. A. Ericsson, N. Charness, P. J. Feltovich, and R. R. Hoffman (Eds.), *The Cambridge Handbook of Expertise and Expert Performance*. Cambridge: Cambridge University Press.

Ericsson, K. A., Krampe, R. T., & Tesch-Romer, C. (1992). The role of deliberate practice in the acquisition of expert performance. *Psychological Review, 100*(3), 363-406.

Evans, J., & Stanovich, K. E. (2013). Dual-process theories of higher cognition: Advancing the debate. *Perspectives on Psychological Science, 8*(3), 223-241.

Ford, J. K., Kraiger, K., & Merritt, S. M. (2010). An updated review of the multidimensionality of training outcomes: New directions for training evaluation research. In. S. W. J. Kozlowski & E. Salas (Eds.), *Learning, training, and development in organizations* (pp. 135-165). New York, NY: Routledge.

Ford, J. K., Quinones, M. A., Sego, D. J., & Sorra, J. S. (1992). Factors affecting the opportunity to perform trained tasks on the job. *Personnel Psychology, 45,* 511-527.

Ford, J. K., Yelon, S. L., & Billington, A. Q. (2011). How much is transferred from training to the job? The 10% delusion as a catalyst for thinking about transfer. *Performance Improvement Quarterly, 24*(2), 7-24.

Fork, J. K., Bhatia, S., & Yelon, S. L. (in press). Beyond direct application as an indicator of transfer: A demonstration of five types of use. *Performance Improvement Quarterly*.

Foxon, M. (1997). The influence of motivation to transfer, action planning, and manager support on the transfer process. *Performance Improvement Quarterly, 10*(2), 42-63.

Friedman, S., & Ronen, S. (2015). The effect of implementation intentions on transfer of training. *European Journal of Social Psychology, 45,* 409-416.

Gentner, D. R. (1988). Thoughts on expertise. In C. Schooler & K. W. Schaie (Eds.), *Cognitive functioning and social structure over the life course* (pp. 81-94). Norwood, NJ: Ablex.

Gist, M., Stevens, C., & Bavetta, A. (1991). Effects of self-efficacy and post-training intervention on the acquisition and maintenance of complex interpersonal skills. *Personnel Psychology, 44*, 837-861.

Goldstein, I. L. (1986). *Training in organizations: Needs assessment, development, and evaluation.* Pacific Grove, CA: Brooks/Cole.

Goldstein, I. L., & Ford, J. K. (2002). *Training in organizations: Needs assessment, development, and evaluation, 4th ed.* Belmont, CA: Wadsworth.

Gollwitzer, P. M. (1999). Implementation intentions: Strong effects of simple plans. *American Psychologist, 54*, 493–503.

Gollwitzer, P. M., & Sheeran, P. (2006). Implementation intentions and goal achievement: A meta-analysis of effects and processes. *Advances in Experimental and Social Psychology, 38,* 69-119.

Grand, J. A. (in press). A general response process theory for situational judgement tests. *Journal of Applied Psychology.*

Grand, J. A., Braun, M. T., Kuljanin, G., Kozlowski, S. W. J., & Chao, G. T. (2016). The dynamics of team cognition: A process-oriented theory of knowledge emergence in teams. *Journal of Applied Psychology, 101*(10), 1352-1385.

Grant, A. M. (2008). The significance of task significance: Job performance effects, relational mechanisms, and boundary conditions. *Journal of Applied Psychology, 93*(1), 108-124.

Greenwald, A., & Banaji, M. (1995). Implicit social cognition: Attitudes, self-esteem, and stereotypes. *Psychological Review*, *102*(1), 4-27.

Guastello, S. J. (1987). A butterfly catastrophe model of motivation in organizations: Academic performance. *Journal of Applied Psychology, 72*(1), 165-182.

Hackman, J. R. (2003). Learning more by crossing levels: Evidence from airplanes, hospitals, and orchestras. *Journal of Organizational Behavior, 24*(8), 905-922.

Hanges, P. J., & Wang, M. (2012). Seeking the Holy Grail in organizational science: Uncovering causality through research design. In S.W.J. Kozlowski (Ed.), *The Oxford Handbook of*

Organizational Psychology, Vol. 1 (pp. 79-116). New York, NY: Oxford University Press.

Harris, P., Brearley, I., Sheeran, P., Barker, M., Klein, W., Creswell, J., LeVine, J., & Bond, R. (2014). Combining self-affirmation with implementation intentions to promote fruit and vegetable consumption. *Health Psychology, 33*(7), 729-736.

Hattrup, K., & Jackson, S. E. (1996). Learning about individual differences by taking situations seriously. In K. R. Murphy (Ed.), *Individual differences and behavior in organizations* (pp. 507–547). San Francisco: Jossey-Bass.

Hausknecht, J., Halpert, J., Di Paolo, N., & Moriarti Gerrard, M. (2007). Retesting in selection: A meta-analysis of coaching and practice effects for tests of cognitive ability. *Journal of Applied Psychology*, *92,* 373–385.

Healy, K. (2017). Fuck nuance. *Sociological Theory, 35*(2), 118-127.

Hollenbeck, J. R., Colquitt, J. A., Ilgen, D. R., LePine, J. A., & Hedlund, J. (1998). Accuracy decomposition and team decision making: Testing theoretical boundary conditions. *Journal of Applied Psychology, 83*(3), 494-500.

Holton, E. F., Bates, R. A., & Ruona, W. E. A. (2000). Development of a generalized learning transfer system inventory. *Human Resource Development Quarterly, 11*(4), 333-360.

Holton, E. G., Bates, R. A., Seyler, D. L., & Carvalho, M. B. (1997). Toward construct validation of a transfer climate instrument. *Human Resource Development Quarterly, 11*(4), 333-360.

Huang, J. L., Blume, B. D., Ford, J. K., & Baldwin, T. T. (2015). A tale of two transfers: Disentangling maximum and typical transfer and their respective predictors. *Journal of Business Psychology, 30,* 709-732.

Huang, J. L., Ford, J. K., & Ryan, A. M. (2017). Ignored no more: Within-person variability enables better understanding of training transfer. *Personnel Psychology, 70*(3), 557-596.

Jaeggi, S. M., Buschkuehl, M., Shah, P., & Jonides, J. (2014). The role of individual differences in cognitive training and transfer. *Memory & Cognition, 42,* 464-480.

Jaidev, U. P., & Chirayath, S. (2012). Pre-training, during-training and post-training activities as predictors of transfer of training. *The IUP Journal of Management Research, 11*(4), 54-70.

Judge, T. A., & Zapata, C. P. (2015). The person-situation debate revisited: Effect of situation strength and trait activation on the validity of the Big Five personality traits in predicting job performance. *Academy of Management Journal, 58*(4), 1149-1179.

Kahneman, D. (2011). *Thinking Fast and Slow.* New York, NY: Farar, Straus, & Giroux.

Kalinoski, Z. T., Steele-Johnson, D., Peyton, E. J., Leas, K. A., Steinke, J., & Bowling, N. A. (2013). A meta-analytic evaluation of diversity training outcomes. *Journal of Organizational Behavior, 34*(8), 1076-1104.

Karoly, P. (1993). Mechanisms of self-regulation: A systems view. *Annual Review of Psychology*, *44*, 23–52.

Keith, N., & Frese, M. (2008). Effectiveness of error management training: A meta-analysis. *Journal of Applies Psychology, 93*(1), 59-69.

Kendzierski, D., Ritter, R., Stump, T., Anglin, C. (2015). The effectiveness of an implementations intention intervention for fruit and vegetable consumption as moderated by self-schema status. *Appetite, 95*, 228-238.

Kenny, D. A. (2005). *Cross-lagged panel design*. Hoboken, NJ: Wiley.

Kessler, R. C. (1992). Perceived support and adjustment to stress: Methodological considerations. In H. O. F. Viel & U. Baumann (Eds.), *The meaning and measurement of social support* (pp. 259-271). New York, NY: Hemisphere.

Kim, Y., & Ployhart, R. E. (2014). The effects of staffing and training on firm productivity and profit growth before, during, and after the Great Recession. *Journal of Applied Psychology, 99*(3), 361-389.

Kirkpatrick, D. L. (1994). *Evaluating training programs: The four levels.* San Francisco, CA: Berrett-Koehler.

Knowles, M. S. (1984). *Andragogy in action: Applying modern principles of adult learning.* San Francisco: Jossey-Bass.

Kolb, D. (1984). *Experiential learning.* Englewood Cliffs, NJ: Prentice Hall.

Kozlowski, S. W. J., & Chao, G. T. (2012). The dynamics of emergence: Cognition and cohesion in work teams. *Managerial and Decision Economics, 33,* 335-354.

Kozlowski, S. W. J., & Klein, K. J. (2000). A multilevel approach to theory and research in organizations: Contextual, temporal, and emergent processes. In K. J. Klein & S. W. J. Kozlowski (Eds.), *Multilevel theory, research and methods in organizations: Foundations, extensions, and new directions* (pp. 3-90). San Francisco, CA: Jossey-Bass.

Kozlowski, S. W. J., Gully, S. M., Brown, K. G., Salas, E., Smith, E. M., & Nason, E. R. (2001). Effects of training goals and goal orientation traits on multidimensional training outcomes and performance adaptability. *Organizational Behavior and Human Decision Processes, 85,* 1-31.

Kraiger, K., & Ford, J. K. (2007). The history of training in industrial/organizational psychology. In L. Koppes (Ed.), *The science and practice of industrial and organizational* *psychology: Historical aspects from the first 100 years.* Mahwah, NJ: Lawrence Erlbaum Associates.

Kraiger, K., Ford, J. K., & Salas, E. D. (1993). Application of cognitive, skill-based, and affective theories of learning outcomes to new methods of training evaluation. *Journal of Applied Psychology*, *78,* 311-328.

Lai, C., Hoffman, K., & Nosek, B. (2013). Reducing implicit prejudice. *Social and Personality Psychology Compass*, *7*(5), 315-330.

Lai, C., Skinner, A., Cooley, E., Murrar, S., Brauer, M., … Nosek, B. (2016). Reducing implicit racial preferences: II. Intervention effectiveness across time. *Journal of Experimental Psychology: General, 145*(5), 1001-1016.

Laker, D. R., & Powell, J. L. (2011). The differences between hard and soft skills and their relative impact on training transfer. *Human Resource Development Quarterly, 22*(1), 111-122.

Lancaster, S., Di Milia, L., & Cameron, R. (2013). Supervisor behaviours that facilitate training transfer. *Journal of Workplace Learning, 25*(1), 6-22.

Langdon, D. G. (1997). Selecting interventions. *Performance Improvement, 36,* 11-15.

Leavitt, K., Qiu, F., & Shapiro, D. L. (in press). Using electronic confederates for experimental research in organizational science. *Organizational Research Methods.*

Lewin, K. (1943). Psychology and the process of group living. *Journal of Social Psychology, 17,* 113-131.

Lindsley, D., Brass, D., & Thomas, J. (1995). Efficacy-performance spirals: A multi-level perspective. *The Academy of Management Review*, *20*(3), 645-678.

Locke, E. (1968). Toward a theory of task motivation and incentives. *Organizational behavior and human performance, 3,* 157-189.

Locke, E. (1975). Personnel attitudes and motivation. *Annual Review of Psychology, 26,* 457-480.

Locke, E., & Latham, G. (1990). *A Theory of Goal Setting and Task Performance.* Englewood Cliffs, NJ: Prentice-Hall.

London, M. (2012). Lifelong learning. In S. W. J. Kozlowski (Ed.), *The Oxford Handbook of Organizational Psychology, Vol. 2* (pp. 1199-1227). New York, NY: Oxford University Press.

Lopes, M., Melo, F. S., Kenward, B., & Santos-Victor, J. (2009). A computational model of social-learning mechanisms. *Adaptive Behavior, 17*(6), 467-183.

Lord, R. G., & Hanges, P. J. (1987). A control system model of organizational motivation: Theoretical development and applied implications. *Behavioral Science*, *32*(3), 161-178.

Lord, R., Diefendorff, J., Schmidt, A., & Hall, R. (2010). Self-regulation at work. *Annual Review of Psychology*, *61*, 543-68.

Ludvig, E. A., Bellemare, M. G., & Pearson, K. G. (2011). A primer on reinforcement learning in the brain: Psychological, computational, and neural perspectives. In E. Alonso and E. Mondragon (Eds.), *Computational neuroscience for advancing artificial intelligence: Models, methods, and applications,* (pp. 111-144). Hershey, PA: IGI Global.

March, J. G. (1991). Exploration and exploitation in organizational learning. *Organization Science*, *2*(1), 71-87.

Martocchio, J. J. (1992). Microcomputer usage as an opportunity: The influence of context in employee training. *Personnel Psychology, 45,* 529-552.

Mathieu, J. E., & Tesluk, P. E. (2010). A multilevel perspective on training and development effectiveness. In. S. W. J. Kozlowski & E. Salas (Eds.), *Learning, training, and development in organizations* (pp. 405-440). New York, NY: Routledge.

Mathieu, J. E., Tannenbaum, S. I., & Salas, E. (1992). Influences of individual and situational characteristics on measures of training effectiveness. *Academy of Management Journal, 35,* 828-847.

McCrae, R. R., & Costa, P. T. (1987). Validation of the five-factor model of personality across instruments and observers. *Journal of Personality and Social Psychology, 52*(1), 81–90.

Melnikoff, D. E. and Bargh, J. A. (2018a) The mythical number two. *Trends in Cognitive Science, 22*, 280–293.

Melnikoff, D. E. and Bargh, J. A. (2018b) The insidious number two. *Trends in Cognitive Science, 22*, 668-669.

Meyer, R. D., Dalal, R. S., & Hermida, R. (2010). A review and synthesis of situational strength in the organizational sciences. *Journal of Management, 36,* 121-140.

Miller, J. H., & Page, S. E. (2007). *Complex adaptive systems: An introduction to computational models of social life.* Princeton, NJ: Princeton University Press.

Muthukrishna, M., & Henrich, J. (2019). A problem in theory. *Nature: Human Behaviour*. doi: 10.1038/s41562-018-0522-1

Myers, C. G. (in press). Performance benefits of reciprocal vicarious learning in teams. *Academy of Management Journal.*

Myers, D. G. (2004). *Psychology* (7th ed.). New York: Worth.

Neal, J. W., & Neal, Z. P. (2013). Nested or networked? Future directions for ecological systems theory. *Social Development, 22*, 722 –737.

Neal, D. T., Wood, W., & Drolet, A. (2013). How do people adhere to goals when willpower is low? The profits (and pitfalls) of strong habits. *Journal of Personality and Social Psychology*, *104*(6), 959-975.

Neal, D. T., Wood, W., & Quinn, J. M. (2006). Habits: A repeat performance. *Current Directions in Psychological Science, 15,* 198-202.

Newell, A., & Rosenbloom, P. S. (1981). Mechanisms of skill acquisition and the law of practice. In J.R. Anderson (Ed.), *Cognitive Skills and their Acquisition* (pp. 1-56). Hillsdale, NJ: Lawrence Earlbaum Associates.

Nijman, D. J. M., Nijhof, W. J., Wognum, A. A. M., & Veldkamp, B. P. (2006). Exploring differential effects of supervisor support on transfer of training. *Journal of European Industrial Training, 30*(7), 529-549.

Noe, R. A. (2017). *Employee Training & Development, Seventh Edition.* New York, NY: McGraw-Hill Education.

Nye, C., Prasad, J., Bradburn, J., & Elizondo, F. (2018). Improving the operationalization of interest congruence using polynomial regression. *Journal of Vocational Behavior, 104,* 154-169.

Olenick, J., Bhatia, S., & Ryan, A. M. (2016). Effects of *g*-loading and time lag on retesting in job selection. *International Journal of Selection and Assessment*, *24*(4), 324-336.

Olenick, J., Blume, B., & Ford, J. K. (in press). A nonlinear framework for understanding employee training and transfer. *European Journal of Work and Organizational Psychology*.

Olenick, J., Walker, R., Bradburn, J., & DeShon, R. (2018). A systems view of the scientist-practitioner gap. *Industrial and Organizational Psychology*, *11*(2), 220-227.

Pavlov, P.I. (1927). *Conditioned Reflexes.* London: Oxford University Press.

Payne, S., Youngcourt, S., & Beaubien, J. (2007). A meta-analytic examination of the goal-orientation nomological net. *Journal of Applied Psychology*, *92*(1), 128-150.

Peetz, J., Wilson, A. E., & Strahan, E. J. (2009). So far away: The role of subjective temporal distance to future goals in motivation and behavior. *Social Cognition, 27*(4), 475-495.

Pennycock, G., De Neys, W., Evans, J., Stanovich, K. E., & Thompson, V. A. (2018). The mythical dual-process typology. *Trends in Cognitive Science*, *22*(8), 667-668.

Pennycook, G., Fugelsang, J. A., & Koehler, D. J. (2015). What makes us think? A three-stage dual-process model of analytic engagement. *Cognitive psychology*, *80*, 34-72.

Ployhart, R. E., & Moliterno, T. P. (2011). Emergence of the human capital resource: A multilevel model. *Academy of Management Review, 36*(1), 127-150.

Ployhart, R. E., & Vandenberg, R. J. (2010). Longitudinal research: The theory, design, and analysis of change. *Journal of management*, *36*(1), 94-120.

Popper, K. R. (1959). *The Logic of Scientific Discovery*. London: Hutchinson.

Powers, W. (1973). *Behavior: The Control of Perception*. New York: Aldine/DeGruyter.

Railsback, S. F., & Grimm, V. (2012). *Agent-Based and Individual-Based Modeling: A Practical Introduction.* Princeton, NJ: Princeton University Press.

Rogers, E. (2003). *Diffusion of Innovations, Fifth Edition.* New York, NY: Free Press.

Rouiller, J. Z., & Goldstein, I. L. (1993). The relationship between organizational transfer climate and positive transfer of training. *Human Resource Development Quarterly, 4,* 377-390.

Rummler, G. (1996). In search of the holy performance grail. *Training and Development*, 26-31.

Ruona, W., Leimbach, M., F. Holton III, E., & Bates, R. (2002). The relationship between learner utility reactions and predicted learning transfer among trainees. *International Journal of Training and Development, 6*(4), 218-228.

Salas, E., & Kozlowski, S. W. J. (2010). Learning, training, and development in organizations: Much progress and a peek over the horizon. In. S. W. J. Kozlowski & E. Salas (Eds.), *Learning, training, and development in organizations* (pp. 461-476). New York, NY: Routledge.

Salas, E., Milham, L. M., & Bowers, C. A. (2003). Training evaluation in the military: Misconceptions, opportunities, and challenges. *Military Psychology, 15,* 3-16.

Salas, E., Weaver, S. J., & Shuffler, M. L. (2012). Learning, training, and development in organizations. In S. W. J. Kozlowski (Ed.), *The Oxford Handbook of Organizational Psychology, Vol. 1* (pp. 330-372). New York, NY: Oxford University Press.

Samuel, A. L. (1967). Some studies in machine learning using the game of checkers. II – Recent progress. *IBM Journal on Research and Development, 11*(6), 601-617.

Schmidt, A. M., & DeShon, R. P. (2007). What to do? The effects of goal-performance discrepancies, superordinate goals, and time on dynamic goal prioritization. *Journal of Applied Psychology, 92,* 928-941.

Schniehotta, F., Scholz, U., & Schwarzer, R. (2005). Bridging the intention-behavior gap: Planning, self-efficacy, and action control in the adoption and maintenance of physical exercise. *Psychology & Health, 20*(2), 143-160.

Scholz, U., Nagy, G., Schuz, B., & Ziegelman, J. (2008). The role of motivational and volitional factors for self-regulated running training: Associations on the between and within-person level. *British Journal of Social Psychology, 47*, 421-439.

Schunk, D. & Usher, E. (2012). Social Cognitive Theory and Motivation. In R. Ryan (Ed.) *The Oxford Handbook of Human Motivation,* (pp. 3-27). New York, NY: Oxford University Press.

Sheeran, P., Webb, T., & Gollwitzer, P. (2005). The interplay between goal intentions and implementation intentions. *Personality and social psychology bulletin, 31*(1), 87-98.

Shen, Y., Tobia, M. J., Sommer, T., & Obermayer, K. (2014). Risk-sensitive reinforcement learning. *Neural Computation, 26*(7), 1298-1328.

Singh, V., Dong, A., & Gero, J. S. (2013). Developing a computational model to understand the contributions of social learning modes to task coordination in teams. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing. 27,* 3-17.

Sitzmann, T., & Weinhardt, J. M. (2019). Approaching evaluation from a multilevel perspective: A comprehensive analysis of the indicators of training effectiveness. *Human Resource Management Review, 29,* 253-269.

Sitzmann, T., & Yeo, G. (2013). A meta-analytic investigation of the within-person self-efficacy domain: Is self-efficacy a product of past performance or a driver of future performance? *Personnel Psychology, 66*, 531-568.

Skinner, B. F. (1938). *The Behavior of Organisms: An Experimental Analysis*. New York, NY: Appleton-Century.

Skinner, B. F. (1963). Operant behavior. *American Psychologist, 18*(8), 503-515.

Smith, E. M., Ford, J. K., & Kozlowski, S. W. J. (1997). Building adaptive expertise: Implications for training design strategies. In M. A. Quiñones & A. Ehrenstein (Eds.), *Training for a rapidly changing workplace: Applications of psychological research* (pp. 89-118). Washington, DC, US: American Psychological Association.

Soltis, S. M., Brass, D. J., & Lepak, D. P. (2018). Social resource management: Integrating social network theory and human resource management. *Academy of Management Annals, 12*(2), 537-573.

Southerton, D. (2012). Habits, routines and temporalities of consumption: From individual behaviours to the reproduction of everyday practices. *Time & Society, 22*(3), 335-355.

Spicer, S. G., Mitchell, C. J., Wills, A. J., & Jones, P. M. (2020). Theory protection in associative learning: Humans maintain certain beliefs in a manner that violates prediction error. *Journal of Experimental Psychology: Animal Learning and Cognition, 46*(2), 151-161.

Stajkovic, A., & Luthans, F. (1998). Self-efficacy and work-related performance: A meta-analysis. *Psychological Bulletin, 124,* 240-261.

Starns, J. J., Cataldo, A. M., Rotello, C. M., Annis, J., Aschenbrenner, A., …Wilson, J. (2019). Assessing theoretical conclusions with blinded inference to investigate a potential inference crisis. *Advances in Methods and Practices in Psychological Science*, *2*(4), 335-349.

Steel, P., & Konig, C. J. (2006). Integrating theories of motivation. *Academy of Management Review, 31*(4), 889-913.

Steele-Johnson, D.., Narayan, A., Delgado, K. M., & Cole, P. (2010). Pretraining influences and readiness to change dimensions: A focus on static versus dynamic issues. *The Journal of Applied Behavioral Science, 46*(2), 245-274.

Stonedahl, F. and Wilensky, U. (2008). NetLogo Diffusion on a Directed Network model. http://ccl.northwestern.edu/netlogo/models/DiffusiononaDirectedNetwork. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

Sun, R., Slusarz, P., & Terry, C. (2005). The interaction and of the explicit and the implicit in skill learning: A dual-process approach. *Psychological Review, 112*(1), 159-192.

Sun, S., Vancouver, J., & Weinhardt, J. (2014). Goal choices as planning: Distinct expectancies and value effects in two goal processes. *Organizational Behavior and Human Decision Processes, 125*, 220-233.

Surface, E., & Olenick, J. (forthcoming). A mechanistic model of training transfer.

Susskind, D., & Susskind, R. (2017). *The Future of the Professions: How Technology Will Transform the Work of Human Experts.* New York, NY: Oxford University Press.

Sutton, R. I., & Staw, B. M. (1995). What theory is not. *Administrative Science Quarterly, 40*(3), 371-384.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction, Second Edition.* Cambridge, MA: The MIT Press.

Tannenbaum, S. I., Beard, R. L., McNall, L. A., & Salas, E. (2010). Informal learning and development in organizations. In. S. W. J. Kozlowski & E. Salas (Eds.), *Learning, training, and development in organizations* (pp. 303-331). New York, NY: Routledge.

Tannenbaum, S. I., Mathieu, J. E., Salas, E., & Cohen, D. (2012). Teams are changing: Are research and practice evolving fast enough? *Industrial and Organizational Psychology, 5*, 2-24.

Tesauro, G. (2002). Programming backgammon using self-teaching neural nets. *Artificial Intelligence, 134*(1-2), 181-199.

Tesauro, G., Gondek, D. C., Lenchner, J., Fan, J., & Prager, J. M. (2013). Analysis of WATSON's strategies for playing Jeopardy! *Journal of Artificial Intelligence Research, 47,* 205-251.

Thayer, P. W., & Teachout, M. S. (1995). *A Climate for transfer model.* (Rep No. ALM-TP-1995-0035). Brooks Air Force Base, TX: Air Force Material Command.

Thorndike, E. L. (1898). Animal intelligence: An experimental study of the associative processes in animals. *The Psychological Review, Series of Monograph Supplements, II*(4).

Trentin, G. (2001). From Formal Training to Communities of Practice via Network-Based Learning. *Educational Technology,41*(2), 5-14.

Turton, R., Bruidegom, K., Cardi, V., Hirsch, C. R., & Treasure, J. (2016). Novel methods to help develop healthier eating habits for eating and weight disorders: A systematic review and meta-analysis. *Neuroscience and Biobehavioral Reviews*, *61*, 132-155.

Tversky, A., & Kahneman, D. (1992). Advances in prospect theory: Cumulative representation of uncertainty. *Journal of Risk and Uncertainty, 5,* 297-323.

Vancouver, J. & Day, D. (2005). Industrial and Organization research on self-regulation: From constructs to applications. *Applied Psychology: An International Review*, *54*(2), 155-185.

Vancouver, J. (2008). Integrating self-regulation theories of work motivation into a dynamic process theory. *Human Resource Management Review*, *18,* 1-18.

Vancouver, J. (2012). Rhetorical reckoning: A response to Bandura. *Journal of Management, 38*(2), 465-474.

Vancouver, J., & Kendall, L. (2006). When self-efficacy negatively relates to motivation and performance in a learning context. *Journal of Applied Psychology, 91*(5), 1146-1153.

Vancouver, J., & Weinhardt, J. (2015). Modeling the mind and the milieu: Computational modeling for micro-level organizational researchers. *Organizational Research Methods, 15*(4), 602-623.

Vancouver, J., Gullekson, N., Morse, B., & Warren, M. (2014). Finding a between person negative effect of self-efficacy on performance: Not just a within-person effect anymore. *Human Performance, 27*(3), 243-261.

Vancouver, J., Moore, K., & Yoder, R. (2008). Self-efficacy and resource allocation: Support for a nonmonotonic, discontinuous model. *Journal of Applied Psychology, 93*(1), 35-47.

Vancouver, J., Weinhardt, J., & Vigo, R. (2014). Change one can believe in: Adding learning to computational models of self-regulation. *Organizational Behavior and Human Decision Processes, 124,* 56-74.

Vermeulen, R. C. M. (2002). Narrowing the transfer gap: The advantages of "as if" situations in training. *Journal of European Industrial Training, 26*(8), 366-374.

Verplanken, B., & Orbell, S. (2003). Reflections on past behavior: A self-report index of habit strength. *Journal of Applied Social Psychology, 33*(6), 1313-1330.

Vignoli, M., & Depolo, M. (2019). Transfer of training process. When proactive personality matters? A three-wave investigation of proactive personality as a trigger of the transfer of training process. *Personality and Individual Differences, 141*, 62-67.

Vroom, V. (1964). *Work and Motivation*. New York, NY: John Wiley & Sons, Inc.

Weichart, E. R., Turner, B. M., & Sederberg, P. B. (in press). A model of dynamic, within-trial conflict resolution for decision making. *Psychological Review.*

Weick, K. E. (1976). Educational organizations as loosely coupled systems. *Administrative science quarterly, 21*(1), 1-19.

Wieber, F., Thurmer, J., Gollwitzer, P. (2015). Promoting the translation of intentions into actions by implementation intentions: Behavioral effects and psychological correlates. *Frontiers in Human Neuroscience, 9*, 1-18.

Wilensky, U. (1999). NetLogo. http://ccl.northwestern.edu/netlogo/. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

Wilson, T., Lindsey, S., & Schooler, T. (2000). A model of dual attitudes. *Psychological Bulletin, 107*(1), 101-126.

Wilson, T., Lindsey, S., & Schooler, T. (2000). A model of dual attitudes. *Psychological Bulletin, 107*(1), 101-126.

Wood, W., Quinn, J. M., & Kashy, D. (2002). Habits in everyday life: Thought, emotions, and action. *Journal of Personality and Social Psychology, 83*, 1281-1297.

Yammarino, F. J., & Dubinsky, A. J. (1994). Transformational leadership theory: Using levels of analysis to determine boundary conditions. *Personnel Psychology, 47*, 787-811.

Yeh, C-H., & Chen, S-H. (2001). Toward an integration of social learning and individual learning in agent-based computational stock markets: The approach based on population genetic programming. *Journal of Management and Economics, 5*(5).

Yelon, S. L., & Ford, J. K. (1999). Pursuing a multidimensional view of transfer. *12*(3), 58-78.

Zerres, A., Huffmeier, J., Freund, P., Backhaus, K., & Hertel, G. (2013). Does it take two to tango? Longitudinal effects of unilateral and bilateral integrative negotiation training. *Journal of Applied Psychology, 98*(3), 478-491.

Zhang, Y., Olenick, J., Chang, C-H., Kozlowski, S. W. J., & Hung, H. (2018). The I in team: Mining personal social interaction routine with topic models from long-term team data. *Proceedings of the 23rd International Conference on Intelligent User Interfaces*, 421-426.