A COMPARISON OF TWO MEDIATION ANALYSIS METHODS WITH SEQUENTIAL MEDIATORS

By

Kyle Bennett

A THESIS

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

Biostatistics – Master of Science

2019

ABSTRACT

A COMPARISON OF TWO MEDIATION ANALYSIS METHODS WITH SEQUENTIAL MEDIATORS

By

Kyle Bennett

Two methods for mediation analysis with sequential mediators were compared using multiple simulation scenarios. The performances of each method were assessed using three key metrics: relative bias, root mean square error, and coverage. The methods shared both similarities and key differences and some modification and adjustment were necessary to perform comparable simulations across the scenarios. Overall performance was assessed primarily using relative bias, where each simulated effect estimate was compared to a “true” effect generated by simulating from a theoretical super population. Simulation scenarios included correctly specified models using both methods and various mis-specified estimation models by incorrectly specifying a critical parameter in the model to assess the performance and robustness of each mediation analysis method. The results of the simulations suggest that one method was particularly more resilient to mis-specification of the model over the other, and that proper specification of the marginal structural model is also critical to minimizing bias and maximizing coverage.

Key Words: mediation analysis; sequential mediators; marginal structural models; relative bias; data simulation; directed acyclic graph; causal inference

This thesis is dedicated to several vital and irreplaceable people in my life: My father, for his relentless encouragement to achieve more and never settle. My wife Amanda, daughter Isobel, and canine companion Willow for their love and support while I dedicated myself to this project. And Anthony Lupo, my supervisor, for it was he that helped accommodate this fruitful endeavor.

I could not have completed this effort without each one of you.

ACKNOWLEDGEMENTS

I would like to first express my appreciation for my academic adviser, Dr. Zhehui Luo, of Michigan State University. Your door was always open to me and no question was too trivial. Your encouragement to pursue this project over “others” will be something I am forever grateful for.

I also owe gratitude to my employer, Neogen Corporation. Their support of my efforts and belief in the skillset I sought to obtain helped make this possible.

Many thanks to Dr. Joseph Gardiner and Dr. Honglei Chen, both of Michigan State University, for their willingness to participate as committee members for this thesis, and Dr. Ahnalee Brincks, also of Michigan State University, for her readiness to audit the project. I could not have asked for better expertise to be part of my team.

Lastly, I would like to acknowledge my friends and family, all of whom allowed me, without question, to change plans and take rainchecks when needed. Their support and understanding never wavered and for that I am indebted. Thank you for being you.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Section 1. Introduction

Mediation analysis in epidemiology is a critical piece for performing causal inference. Because the pathway from an exposure to an outcome is not always clearly defined, we must utilize analytic methods to help us understand the impact that mediating variables have on the pathways from said exposures to said outcomes (1). Substantial academic work on mediation analysis has been published prior to this thesis across a wide range of social and life science applications (2–6). Two specific methods (2,3) will be scrutinized here and discussed in detail.

The purpose of this thesis is to execute a thorough comparison of two mediation analysis methods. In no particular order, the first method is proposed by Lange *et al.* (2), which builds on work introduced by Lange, Vansteelandt, & Bekaert (7). The second, brought forth by Steen *et al.* (3), focuses heavily on the decomposition of models including multiple mediators, an idea previously presented by Daniel *et al.* (8). For this current work, the focus is on two sequential mediators evaluated using Monte Carlo data simulation. The specifics of the data simulation will be described in later sections, but in short, model components were incorrectly specified in various ways to explore the properties of the estimators by each statistical method. Statistical performance was assessed using several metrics: (i) relative bias, expressed as a percent difference from a “true” effect obtained from a very large population; (ii) root mean square error (RMSE); and (iii) coverage, expressed as a percent of the instances where the confidence interval generated by each simulation replicate covers the “true” effect. A discussion on the “true” effect and how it was generated occurs later, but it can be summarily thought of as a reasonable estimate of the effect if it were generated from a large population of 1,000,000 observations. We refer to this large population throughout the paper as a ‘Super Population’ and assume that data generated from the Super Population closely reflect true effects supported by Bernoulli’s law of large numbers.

Some motivating background can be found in Liu *et al.’s* work on the association between poor olfaction and mortality (9). In this study, two mediators are presented sequentially, and the variables are treated as binary in nature. Part of the statistical analysis was to measure the strength of association between the exposure (olfaction impairment), mediators (weight loss and dementia and/or Parkinson disease) and outcome (mortality) with respect to explaining the overall total effect (TE) via natural direct effect (NDE), natural indirect effect (NIE), and partial indirect effect (PIE). Since multiple methods are present in the literature for explaining TE and the related direct and indirect effects, this created an ideal opportunity to compare method performance using simulated data constructed to resemble Liu *et al.’s* setting.

The aforementioned effects of NDE, NIE, PIE, and TE, which are also the primary effects of interest in this current study, are the parameters researchers use to measure the impacts of variables on an outcome within natural (or counterfactual) effects models (2,3,10). In terms of mediation analysis, the NDE (natural direct effect) explains the effect of an exposure when a mediator is set to the mediator level that an individual would have experienced without the exposure (11). Pearl (12) describes *direct effects* as quantities measuring the impact of exposures on outcomes that are not mediated by other intermediates in a model. The natural and partial indirect effects constitute the effects represented in a model that explain the extent to which a response variable changes had the exposure been held fixed and the mediator variable is allowed to change as it would have if the exposure shifts to a different level (13).

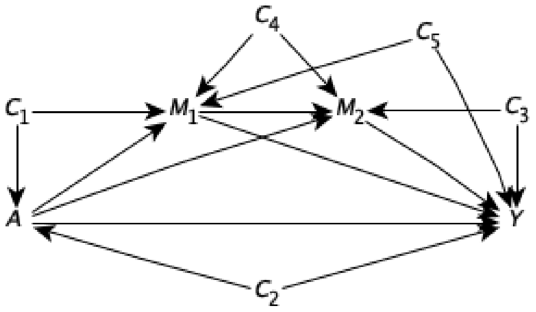
Indirect effects can also be thought of as “path-specific” effects since they are measured through a specific combination of mediators and confounders from exposure to outcome (14). Direct and indirect effects together constitute total effects (11). As Pearl (14) notes, total effects are often the easiest to define and interpret since most experiments are designed to measure the effect of an exposure variable on a response variable. However, in order to conduct a thorough mediation analysis—that is, to attempt to understand the direct and indirect effects that sum to the total effects—a proper decomposition of the total effect into its smaller parts is appropriate and preferred (2,3,15).

This thesis will compare Lange *et al.’s* and Steen *et al.’s* proposed algorithms for quantifying and modeling component and total effects. The performances of each will be assessed through multiple data simulation scenarios, each carrying their own unique parameterization and specifications in the structural models. The results of the simulations will be tabulated concisely to display how well each algorithm, with respect to the specific scenario, manages the correct or incorrect specification of the component models. The strengths and weaknesses of each method, again with respect to specific scenarios and specifications, will be discussed and an aim of this effort will be to provide suggestions on the causes of each pitfall. It is important to note, however, that the set of effects chosen for evaluation here are simply one subset of all component effects in a natural effects model with multiple mediators. The justification for this is that this subset theoretically provides the closest estimation of the natural effects in our proposed model, therefore it was believed this was the best point to explore a total effect decomposition.

Section 2. Description of Deployed Methods

A primary foundation of this simulation study is depicted in the directed acyclic graph (DAG) shown in Figure 1. The causal assumptions shown in the DAG were the basis for establishing the true data generating process for both the super population estimates and the properly specified models for each statistical method. Of note, the mediators in this causal diagram share a sequential relationship, where M_1 precedes M_2 in all facets and scenarios. The mediator pathway is never reversed. Additionally, each confounding variable directly affects two other variables. These details will be important to recall when the discussion regarding the ramifications of the simulations occurs later. All variables in this DAG are binary.

Figure 1. Directed Acyclic Graph for Data Generating Process



Notably, Lange *et al.'s* method (2) was slightly modified as a consequence of the chosen design of the data generating process. Since the mediators for the simulation were to be established sequentially, it was necessary to model mediator two (M_2) using mediator one (M_1) as a covariate. This effectively renders a step in the published Lange *et al.* method irrelevant—testing the mediators for mutual independence (2). Since M_2 is modeled from M_1, we would expect dependence in the mediators. A proof for how the newly modified procedure featuring sequential mediators provides validity is shown in Appendix A.

The Lange *et al.* (2) and Steen *et al.* (3) approaches share several similarities in their respective procedures. Both methods require modeling of at least one mediator. Both require an expansion of the original dataset. The two methods also require generating weights for at least one mediator and using those weights to fit a suitable model to the outcome variable. Each approach is also based on comparing counterfactual outcomes to quantify the causal effects.

Despite these similarities, the Lange *et al.* and Steen *et al.* statistical methods also carry some contrasts to each other. Perhaps most notably, prior to data expansion and weight generation, the Lange *et al.* method requires that both mediators be modeled, while disregarding any model for the response variable. Contrary to this, the Steen *et al.* method requires only one of the two mediators be modeled along *with* modeling the response. Additionally, the weights for Lange *et al.'s* method must be generated using prediction models based on both mediators M_1 and M_2. This is unlike Steen *et al.'s* approach where regression weights are generated from either M_1 *or* M_2, dependent entirely on which mediator was modeled in step one of their procedure. Finally, since the Steen *et al.* method requires preemptively modeling the mean outcome prior to data expansion, fitted values for the expectation of Y (the response) are used to impute outcomes and fit the final natural effects model including given confounders and measured covariates. In Lange *et al.*, the response variable is not modeled prior to data expansion, therefore the expectation of Y is modeled using only the observed values for Y and the exposure (plus interactions, if applicable).

A short summary of the procedures executed for each mediation analysis method is provided below:

1. **Modified Lange *et al.* (also referred to as "Lange method")**
    a. Execute data generating process program (DGP; see Appendix B).
    b. Generate models for exposure A and mediators M_1 and M_2 and store estimated parameters.
    c. Expand dataset to create 4 replicates of each observation.
        i. Each replicate is characterized by a unique combination of exposure effects.
        ii. All other variables for the subject are repeated in each replicate.
    d. Compute marginal weights using estimated parameters for both mediators M_1 and M_2 in step 'b'.
    e. Define marginal structural model (MSM) in both linear and non-linear forms.
    f. Extract effects of interest from MSM model estimates.
2. **Steen *et al.* (also referred to as "Steen method")**
    a. Execute DGP.
    b. Generate models for exposure A, mediator M_1 or M_2, and outcome Y and store estimated parameters.
    c. Expand dataset to create 4 replicates of each observation.
        i. Each replicate is characterized by a unique combination of exposure effects.
        ii. All other variables for the subject are repeated in each replicate.

d. Compute marginal weights.
    i. Generate weights from prediction models using previously stored estimates from exposure *A* and one of two mediators.
e. Impute outcome *Y*.
    i. Generate imputed outcomes by recalling the stored parameters from the *Y* model in step ‘b’ and use a prediction process to generate a fitted value for *Y* (fitY).
f. Define MSM featuring fitY in both linear and non-linear forms.
g. Extract effects of interest from MSM model estimates.

Section 3. Data Generating Process (DGP)

3.1 True Data and the Super Population

The process for generating data for this simulation begins with the DAG shown in Figure 1. Finding motivation in Liu *et al.* (9), an aim was to set parameter specifications for the individual covariates in such a way that the prevalence of the exposure, mediators, and outcome would mimic the observed data in their paper. Equations 1-4 in Appendix D detail the parameter specifications for the true data generated from the super population as well as the subsequent random variable distribution.

A check on the prevalence of each generated variable is shown in Table 1. Comparing these frequencies to the published data in Liu *et al.*, it was found that the exposure, mediator, and outcome prevalence compare well. Referencing their baseline data for the poor olfaction exposure (approximately 32%), overall average frequencies for both mediators of dementia/Parkinson's Disease (≈29%) and weight loss ≥ 2% (≈19%), and a total outcome count of number of deaths (≈53%), the super population prevalence of exposure (A), mediators (M_1 and M_2), and outcome (Y) are similar with estimates of 24%, 32%, 27%, and 46%, respectively.

Table 1. Prevalence of Exposure, Mediators, & Outcome in Super Population.

A	Freq Percent	M_1	Freq Percent	M_2	Freq Percent	Y	Freq Percent
0	76.05	0	68.35	0	73.30	0	54.03
1	23.95	1	31.65	1	26.70	1	45.97
Total	1000000	**Total**	1000000	**Total**	1000000	**Total**	1000000

Abbreviations: A, exposure; M_1, mediator 1; M_2, mediator 2; Y, outcome.

3.2 Computing True Effects

In order to gain some perspective on what the direct and indirect effects from mis-specified models mean, it's vital to discuss how the true effects were derived from the super population. After the exposure, mediator, and outcome variables were generated, counterfactuals for M_1 and M_2 were computed, followed by generating counterfactual outcomes for potential Y scenarios. These counterfactual outcomes are defined as the outcomes that would have been observed, perhaps contrary to factual observation, had exposure A been set to a^* while the mediators were set to the levels they would have taken if A were set to level a (2,3,7,16). From here, direct and indirect effects were obtained from these counterfactuals and used as benchmarks in assessing the performance of each simulation scenario. Equations 5-8 detail the definitions for these effects. Of note, the expressions shown operate on a scale of risk difference (RD). To obtain a scale of risk ratio (RR), one would divide the first term by the second term as opposed to subtracting.

$$(5)\ NDE_{RD} = E\left(Y_{1,M_1(0),M_2(0,M_1(0))}\right) - E\left(Y_{0,M_1(0),M_2(0,M_1(0))}\right)$$

$$(6)\ NIE_{RD} = E\left(Y_{1,M_1(1),M_2(1,M_1(1))}\right) - E\left(Y_{1,M_1(0),M_2(1,M_1(0))}\right)$$

$$(7)\ PIE_{RD} = E\left(Y_{1,M_1(0),M_2(1,M_1(0))}\right) - E\left(Y_{1,M_1(0),M_2\left(0,M_1(0)\right)}\right)$$

$$(8)\ TE_{RD} = E\left(Y_{1,M_1(1),M_2(1,M_1(1))}\right) - E\left(Y_{0,M_1(0),M_2\left(0,M_1(0)\right)}\right)$$

As explained before, this is one decomposition of the natural effect such that the total effect is the sum of the component effects. For other decompositions, see Daniel *et al.* (8) and Steen *et al*. (3). For each simulation scenario, the DGP as described in equations 1-4 was used for each simulation to generate 2000 observations. The direct, indirect, and total effects of interest were estimated within each scenario using the two statistical approaches discussed here. To assess performance, method- and scale-specific effects from each simulation were then compared to the true effects obtained from the super population.

Section 4. Data Simulation

4.1 Simulation Scenarios

To investigate the robustness of each mediation analysis method, several scenarios featuring mis-specified parametric models were defined and implemented into the data simulation program. Table 2 provides a condensed description of each numbered scenario. Moving forth, scenarios will be referred to either by their method and number (e.g. Steen Scen. 3), or by the defining feature of the mis-specification (e.g. outcome Y mis-specified due to unmeasured confounding). In addition to a reduced description, it is also indicated which scenarios were applicable to the two analysis methods.

Table 2. Description of Deployed Simulation Scenarios.

Applicable Methods	Scenario #	Description
Lange & Steen	True	All models correctly specified according to the DGP
	1	M_1 mis-specified due to unmeasured confounding
	2	M_2 mis-specified due to unmeasured confounding
	3	M_2 mis-specified due to lack of $A*M_1$ interaction
Steen only	4	Y mis-specified due to unmeasured confounding
	5	Y mis-specified due to lack of M_1*M_2 interaction

Abbreviations: DGP, data generating process.

Lange *et al.'s* and Steen *et al.'s* methods vary at their foundations; therefore, it was not possible to simulate all scenarios using both methods. Because Lange *et al.'s* method does not require modeling the response variable Y, it was not necessary to mis-specify Y and apply the Lange method. As a result, the discussion section will primarily touch on the Steen method's isolated performance with respect to scenarios four and five.

4.2 Performance Metrics Definitions

Each simulation scenario was replicated 2000 times. The performance metrics of interest were calculated using the observations in the resulting dataset and stored in a separate file for later use in table presentation. Specific metrics of interest to this study were relative bias, RMSE, and coverage.

Relative bias is calculated by

$$\left(\frac{\hat{\bar{\gamma}}_{sim}-\gamma_{ref}}{\gamma_{ref}}\right) \times 100,$$

where $\hat{\bar{\gamma}}_{sim}$ and γ_{ref} are effect-specific. For example, the NDE RD estimates were obtained from

$$\gamma_{ref_{RD}} = E\left(Y_{1,M_1(0),M_2(0,M_1(0))}\right) - E\left(Y_{a^*,M_1,M_2}\right) \text{ and}$$

$$\hat{\bar{\gamma}}_{sim} = \frac{1}{2000}\sum_{i=1}^{2000}(\hat{\gamma}_i),$$

where $\hat{\gamma}_i$ is the estimate for simulation iteration *i*.

On the RR scale, γ_{ref} is obtained by

$$\gamma_{ref_{RR}} = \frac{E\left(Y_{a,M1,M2}\right)}{E\left(Y_{a^*,M1,M2}\right)}.$$

RMSE is given by

$$\sqrt{\frac{1}{2000}\sum_{i=1}^{2000}\left(\hat{\gamma}_i - \gamma_{ref}\right)^2}.$$

Coverage is determined by

$$\frac{1}{2000}\sum_{i=1}^{2000}\left[\gamma_{ref} \in [LB_i, UB_i]\right]$$

where $LB_i\ and\ UB_i$ are the respective lower bound and upper bound limits of the corresponding effect for simulation iteration *i*. The standard errors of estimators in each simulation were calculated using the cluster robust standard errors due to data expansion. Other approaches for calculating the standard errors and confidence intervals are available, e.g., bootstrapping. The implication is discussed later.

4.3 Two-way Interactions

For all simulation scenarios, two versions of the natural effects MSM model were estimated. In the first, only main effects were included. In the second, main effects and two-way interactions were included in the MSMs. The second specification is the preferred simulation for interpreting the results of the study, since the two-way interactions are implied by the DGP of the model. Stata code for how the models were programmed for the preferred simulation is shown in Appendix E for the “True” scenarios pertaining to each method. More commentary on the impact of omitting the two-way interactions occurs in the Discussion section of this thesis.

Section 5. Simulation Results

The results of the preferred simulation, which includes main effects and two-way interactions, are shown in Tables A1-A4, located in Appendix C. They are not included here due to the size of the tables. In Tables A1 and A3, each row corresponds to a specific causal effect on the RD scale, whether that effect was estimated from a linear or non-linear model, and specified performance metrics. In tables A2 and A4, each row corresponds to a specific causal effect on the RR scale and specified performance metrics.

The intention of providing results from linear modeling was to show that even in situations where a non-linear approach is preferred, the linear model performed comparably. With respect to relative bias and coverage, the two modeling approaches are nearly indistinguishable. Where the scenarios featuring mis-specified models failed to provide ample coverage and minimize bias, the non-linear and linear models displayed similar trends and results.

For the Lange *et al.* approach (Tables A1 and A2), when the models were correctly specified, the bias for the NDE and TE was small and the coverage was acceptable. The coverage for the NIE and PIE in this scenario was lower than expected (<90%). The most relative bias occurred when mediator M_1 was mis-modeled due to unmeasured confounding (Scen.1). As expected, this impacted the NIE resulting in relative bias greater than 20% and poor coverage (57.4%). The relative bias for PIE when mediator M_2 was mis-specified due to omitting the $A*M1$ interaction (Scen.3) also elevated greater than 20%, but the coverage managed to remain above 80%. Smaller relative bias also occurred when M_2 was mis-specified due to unmeasured confounding

(Scen.2), affecting the PIE in both the linear and nonlinear models. Notably, the total effect (TE) was relatively unaffected by any mis-modeling using the Lange method, which speaks to the resilience of the method on this effect. For this effect, relative bias remained less than 2% and coverage was greater than 93% across all scenarios. The Lange *et al.* approach performed as expected with the only arguable surprise being the resilience shown by the total effect data across all scenarios. In comparing results in tables A1 and A2, we see that the bias for RR scales was relatively smaller than the bias in the RD scales.

In terms of relative bias, Steen *et al.'s* method performed comparably—and in some scenarios, better—to the modified Lange *et al.* method, but coverage was questionable to poor throughout (Tables A3 and A4). A detailed discussion on the likely causes of this occurs later. Unexpectedly, when mediator M_1 was mis-specified, the relative bias was higher for the PIE than the NIE (Scen.1). This is also where the relative bias was highest during the entire simulation. Elevated relative bias also occurred for the NDE, PIE (in the RD scale), and TE when outcome Y was specified incorrectly (Scen.4). Interestingly, when outcome Y was mis-modeled due to omission of the M_1*M_2 interaction (Scen.5), the Steen *et al.* method was largely unaffected despite the additional requirement that Y be modeled prior to data expansion. Like Lange *et al.*, the TE was resilient to parameter mis-specification, maintaining coverages greater than 80% and relative bias less than 2% across all scenarios except Scen.4 (where Y is mis-modeled due to unmeasured confounding). Additionally, the coverage for all scenarios during the Steen *et al.* portion of the simulation on the NDE was poor, never exceeding 30%. All things considered, the Steen method also performed as expected and a further discussion on this can be found in the next section.

As stated earlier, the simulation was also executed when the interaction effects in the MSM for the outcome *Y* were omitted from the models. This led to increased relative bias and more extreme coverage mishaps in both statistical methods and across all effects. A summary of this inadequacy can be seen in Table 3, which shows the performance metrics for both Lange's and Steen's correct specification scenario. Relative bias is increased for all component effects obtained via logistic regression models as compared to linear probability models and coverage is poor for NIE and PIE. Both methods are effective at mitigating the bias in NDE, NIE and TE—but not in PIE—created from two-way interaction omission when linear probability models for M_1, M_2 and *Y* were used, although the coverage for the Lange method was always better.

Table 3. Performance in Risk Difference Scale with Interactions Missing from MSM when Both Methods Used Correct Specifications.

			Lange *et al.*	**Steen *et al.***
True Effects	**Model**	**Metric**	**True^a**	**True^a**
NDE_{RD} = .216	Linear	Bias %	-1.513 (15.772)	-1.710 (11.192)
		RMSE	0.027 (0.020)	0.019 (0.016)
		Coverage	95.6%	35.3%
	GLM	Bias %	-15.121 (14.646)	-15.428 (10.328)
		RMSE	0.038 (0.025)	0.035 (0.020)
		Coverage	83.9%	15.8%
NIE_{RD} = .088	Linear	Bias %	2.442 (13.714)	2.542 (12.491)
		RMSE	0.010 (0.008)	0.009 (0.007)
		Coverage	83.0%	60.4%
	GLM	Bias %	-25.240 (12.309)	-25.137 (10.737)
		RMSE	0.023 (0.010)	0.022 (0.009)
		Coverage	31.4%	9.2%
PIE_{RD} = .059	Linear	Bias %	12.138 (18.139)	12.096 (34.595)
		RMSE	0.010 (0.008)	0.017 (0.013)
		Coverage	81.2%	95.7%
	GLM	Bias %	-23.022 (15.453)	-22.950 (25.425)
		RMSE	0.014 (0.008)	0.016 (0.012)
		Coverage	54.1%	87.1%

Table 3 (cont'd)

TE_{RD} = .363	Linear	Bias %	1.667 (8.824)	1.568 (8.710)
		RMSE	0.026 (0.020)	0.025 (0.020)
		Coverage	93.7%	84.5%
	GLM	Bias %	-1.922 (7.798)	-2.020 (7.793)
		RMSE	0.023 (0.018)	0.022 (0.019)
		Coverage	95.0%	86.7%

Note: Values in parentheses are the standard deviations of the measure of interest from 2000 replicates.

Abbreviations: NDE, natural direct effect; NIE, natural indirect effect; PIE, partial indirect effect; TE, total effect; RD, risk difference; GLM, generalized linear model; RMSE, root mean square error.

[a] All models correctly specified according to the DGP.

Section 6. Discussion

Among the more noteworthy discoveries made during the study, properly specifying the marginal structural model (MSM) is critical. After the first execution of the simulation—where significant interactions were omitted from the MSM—the relative bias and coverage metrics of each scenario in both statistical methods were poor. As Table 3 shows, even in scenarios where each parameter is correctly modeled, the performance metrics are unexpectedly poor to questionable in quality. While the linear modeling of both methods performs uniformly better than the GLM models, GLM models are almost always the default choice in real world applications. The lesson to be learned here is that even if the algorithm is correctly programmed and sequenced, omitting interactions implied by the DGP for the MSM model will lead to increased relative bias and poor coverage in the effect estimates. To counter the problem of mis-specified MSMs, flexibility in the model is encouraged while testing for significance in the interaction effects.

In instances where confounders were unmeasured or interactions were missing from the model, relative bias was significantly increased. The only real exception to this consequence was the total effect (TE) analysis using the Lange *et al.* method. Across all scenarios for the Lange method, the relative bias remained less than |2|% while the coverage held greater than 93%. This advantage is likely because the Lange method does not require modeling the outcome variable. The outcome is fit after data expansion and weights are calculated, leading to an outcome variable that is modeled directly by a weighted approach. Moreover, the mis-specification of mediator M_2 in both statistical methods led to lower relative bias in TE when the $A*M_1$ interaction was omitted (Scen. 3) than in scenarios where confounding was unmeasured (Scen. 2). Also, in the Steen *et al.* method, TE experienced the lowest relative bias for outcome mis-modeling in the scenario where the M_1*M_2 interaction was omitted (Scen. 5), not where confounding was unmeasured (Scen. 4). These findings suggest that with respect to TE, the omission of interactions versus confounders that are unmeasured leads to lower relative bias. In fact, this trend held true in the Steen method across all effects for outcome mis-modeling and for the NDE when M_2 was mis-specified.

Both the natural indirect effect (NIE) through M_1 and partial indirect effect (PIE) through M_2 are heavily biased for both Lange and Steen when mediator M_1 is mis-modeled, but the bias for PIE is larger with Steen *et al.* Interestingly, the NIE bias behaves differently in each method when mediator M_2 is mis-specified. In the modified Lange *et al.* method, unmeasured confounding leads to higher relative bias, rather than omitting the $A*M_1$ interaction. In the Steen *et al.* method, though, the relative bias for NIE is higher when the $A*M_1$ interaction is missing, rather than when unmeasured confounding is in play. This may be due to differences in the methods, as Lange's method models both mediators, while Steen's method only models one of the mediators, which was M_2 for scenarios two and three. To help illustrate method- and effect-specific performance, Table 4 shows a basic comparison of the methods on the RR scale across the "True" scenario and Scen. 1-3.

Table 4. Comparison of Lange Vs. Steen Methods using Relative Bias in Risk Ratio Scale.

Effects	Metric	True[a]	Scen. 1[b]	Scen. 2[c]	Scen. 3[d]
NDE_{RR}	Bias %	—	—	✓	—
NIE_{RR}	Bias %	—	—	✓	✓
PIE_{RR}	Bias %	✓	✓	—	—
TE_{RR}	Bias %	—	—	✓	✓

Note: '✓' if Lange method < Steen method, '—' otherwise.

Abbreviations: NDE, natural direct effect; NIE, natural indirect effect; PIE, partial indirect effect; TE, total effect; RR, risk ratio.

[a] All models correctly specified according to the DGP.

[b] M_1 mis-specified due to unmeasured confounding.

[c] M_2 mis-specified due to unmeasured confounding.

[d] M_2 mis-specified due to lack of $A*M_1$ interaction.

Another notable observation from this simulation is the relatively poor coverage across all effects and scenarios using Steen *et al.'s* method. This is likely due to the need to bootstrap in order to obtain accurate standard errors, which was not executed for this simulation. Lange *et al.* suggests that robust standard errors can be obtained from the simulation alone, leading to conservative confidence intervals (2). We were interested in exploring this approach, so to equalize the handling of each statistical method, the same approach was utilized for Steen *et al.'s* method as well. It would be reasonable to consider the absence of bootstrapping a limitation of this study, though the information gleaned from this decision may be valuable for future work.

It is interesting that the TE using Lange's modified method is so resilient to mis-specification of relevant models. It seems this is because Lange's method does not require modeling the outcome prior to expansion and weighting. The impact of modeling the outcome preemptively is shown using Steen's method in scenario four, where unmeasured confounding leads to poor coverage and high bias across all effects. Because TE is simply the sum of the direct and indirect effects, it would seem reasonable that any mis-modeling of the outcome would bias TE using Steen's method, however this is only witnessed in the scenario where confounders were unmeasured. The performance remains relatively unbiased and coverage is fair when the $M_1 * M_2$ interaction is omitted. This suggests that accounting for the confounders involved in the exposure and outcome is critical, because one can circumvent the $A \rightarrow M_k \rightarrow Y$ pathway and still assess the total effect through the $A \rightarrow Y$ pathway, though somewhat less accurately and obviously without proper model decomposition.

With respect to 'between scenarios' relationships, it was found that relative bias for an effect could be impacted depending on the confounders chosen for model mis-specification. Referencing the DAG in Figure 1, for example, confounders C_4 and C_5 were omitted when mis-modeling mediator M_1 in scenario one for Lange's modified method. These confounders also affect mediator M_2 and outcome Y. The estimated -11% bias in the PIE for simulation scenario one reflects this secondary consequence. As such, it would be sensible to extend this thinking and expect that omitting C_1 from a model of exposure A would impact the NIE through M_1, like omitting C_2 from a model of outcome Y would impact the TE measured through the $A \rightarrow Y$ pathway.

Section 7. Conclusions

Proper specification of the marginal structural model (MSM) is critical when decomposing total effect into direct and indirect effects. To minimize bias in a modeling effort, confounders must be measured, and significant interactions included in the modeling processes. When confounders are unmeasured and interactions are omitted from a natural effects model, more than one component effect can be impacted, depending on the specific confounder(s) or interaction(s) that is(are) absent. Linear models perform comparably to non-linear modeling techniques, however, are less preferred when working with a series of Bernoulli random variables. Furthermore, bootstrapping will likely provide better coverages as the calculation of the standard errors becomes more accurate. Compared directly and within the scope of the simulation design and with respect to sequential mediation, the Steen *et al.* method appears to be more resilient to mis-specifications and generally more apt at minimizing bias than the modified Lange *et al.* method. In situations where confounders or interactions may be missing from an MSM, one may be able to mitigate any shortcomings created as a result by giving extra attention to the total effect of a causal pathway, rather than the component effects. When mis-modeling or mis-specification occurs in a model, the direct and indirect effects can be significantly and negatively impacted while the total effect seems to remain rather unperturbed to biasing and coverage issues. The exception to this would be in situations where the deployed statistical technique requires modeling of the outcome in the algorithm and the outcome variable is mis-specified.

APPENDICES

APPENDIX A: Proof of Modified Lange Method

$$E\left[Y_{a,M^1_{a^1},M^2_{a^2}}\right] = \sum_{m^1,m^2}\sum_{c} E\left[Y_{a,m^1,m^2} | M^1_{a^1} = m^1, M^2_{a^2} = m^2, C = c\right]$$

$$\times P\left[M^1_{a^1} = m^1, M^2_{a^2} = m^2 | C = c\right] P(C = c)$$

$$= \sum_{m^1,m^2}\sum_{c} E\left[Y_{a,m^1,m^2} | C = c\right] P\left[M^1_{a^1} = m^1 | C = c\right] P\left[M^2_{a^2} = m^1 | M^1_{a^1} = m^1, C = c\right] \times P(C = c)$$

$$= \sum_{m^1,m^2}\sum_{c} E\left[Y_{a,m^1,m^2} | M^1 = m^1, M^2 = m^2, A = a, C = c\right]$$

$$\times P\left[M^1_{a^1} = m^1 | A = a^1, C = c\right] P\left[M^2_{a^2} = m^1 | A = a^2, M^1_{a^1} = m^1, C = c\right] P(C = c)$$

$$= \sum_{m^1,m^2}\sum_{c} E\left[Y_{a,m^1,m^2} | M^1 = m^1, M^2 = m^2, A = a, C = c\right]$$

$$\times P[M^1 = m^1 | A = a^1, C = c] P\left[M^2 = m^1 | A = a^2, M^1_{a^1} = m^1, C = c\right] P(C = c)$$

$$= \sum_{y}\sum_{m^1,m^2}\sum_{c} yP[Y = y | M^1 = m^1, M^2 = m^2, A = a, C = c]$$

$$\times P[M^1 = m^1 | A = a^1, C = c] P[M^2 = m^1 | A = a^2, M^1 = m^1, C = c] P(C = c)$$

$$= \sum_{a}\sum_{y}\sum_{m^1,m^2}\sum_{c} yI(A = a) P[Y = y | M^1 = m^1, M^2 = m^2, A = a, C = c]$$

$$\times P[M^1 = m^1 | A = a, C = c] \frac{P[M^1 = m^1 | A = a^1, C = c]}{P[M^1 = m^1 | A = a, C = c]}$$

$$\times P[M^2 = m^2 | A = a, M^1 = m^1, C = c] \frac{P[M^2 = m^2 | A = a^2, M^1 = m^1, C = c]}{P[M^2 = m^2 | A = a, M^1 = m^1, C = c]}$$

$$\times P(C = c) \frac{1}{P(A = a | C = c)}$$

$$= \sum_{a}\sum_{y}\sum_{m^1,m^2}\sum_{c} yI(A = a) P[Y = y, M^1 = m^1, M^2 = m^2, A = a, C = c] W$$

$$= E[YI(A = a)W]$$

APPENDIX B: Stata Code for Data Generating Procedure for Super Population and Simulations

```
/*----------------------------------------------------------------------------*/
// Generating true TE, NDE, NIE1, PIE2
/*----------------------------------------------------------------------------*/
clear
set seed 1234
set obs 1000000
gen ID=_n
/*----------------------------------------------------------------------------*/
// Data Generating Process (DGP) for A, M1, M2, Y
/*----------------------------------------------------------------------------*/
gen byte C1=rbinomial(1,0.5)
gen byte C2=rbinomial(1,0.5)
gen byte C3=rbinomial(1,0.5)
gen byte C4=rbinomial(1,0.5)
gen byte C5=rbinomial(1,0.5)
local lgodds_a “-log(16)+log(4)*C1+log(4)*C2”
        gen double Pa=exp(`lgodds_a’)/(1+exp(`lgodds_a’))
        gen byte A=rbinomial(1,Pa)
local lgodds_m1 “-log(32)+log(4)*C1+log(4)*C4+log(4)*C5+log(4)*A”
        gen double PM1=exp(`lgodds_m1’)/(1+exp(`lgodds_m1’))
        gen byte M1=rbinomial(1,PM1)
local lgodds_m2 “-log(32)+log(3)*M1+log(4)*C4+log(4)*C3+log(3)*A+log(2)*A*M1”
        gen double PM2=exp(`lgodds_m2’)/(1+exp(`lgodds_m2’))
        gen byte M2=rbinomial(1,PM2)
local lgodds_y “-
log(32)+log(4)*C3+log(4)*M1+log(4)*M2+log(4)*C5+log(4)*C2+log(3)*A+log(4)*M1*M2+log(3)
*A*M1+log(4)*A*M2”
        gen double PY=exp(`lgodds_y’)/(1+exp(`lgodds_y’))
```

```
        gen byte Y=rbinomial(1,PY)
/*Check distribution of A, M1, M2, Y*/
tab1 A M1 M2 Y
/*----------------------------------------------------------------------------*/
// Generate counterfactual Mediator M1:
// M1(0)
// M1(1)
/*----------------------------------------------------------------------------*/
local lgodds_m1_0 “-log(32)+log(4)*C1+log(4)*C4+log(4)*C5+log(4)*0”
        gen double PM1_0=exp(`lgodds_m1_0’)/(1+exp(`lgodds_m1_0’))
        gen byte M1_0=rbinomial(1,PM1_0)
local lgodds_m1_1 “-log(32)+log(4)*C1+log(4)*C4+log(4)*C5+log(4)*1”
        gen double PM1_1=exp(`lgodds_m1_1’)/(1+exp(`lgodds_m1_1’))
        gen byte M1_1=rbinomial(1,PM1_1)
/*----------------------------------------------------------------------------*/
// Generate counterfactual Mediator M2:
// M2(0,M1(0))
// M2(0,M1(1))
// M2(1,M1(0))
// M2(1,M1(1))
/*----------------------------------------------------------------------------*/
local lgodds_m2_00 “-log(32)+log(3)*M1_0+log(4)*C4+log(4)*C3+log(3)*0+log(2)*0*M1_0”
        // plug in A = 0, M = M1_0
        gen double PM2_00=exp(`lgodds_m2_00’)/(1+exp(`lgodds_m2_00’))
        gen byte M2_00=rbinomial(1,PM2_00)
local lgodds_m2_01 “-log(32)+log(3)*M1_1+log(4)*C4+log(4)*C3+log(3)*0+log(2)*0*M1_1”
        // plug in A = 0, M = M1_1
        gen double PM2_01=exp(`lgodds_m2_01’)/(1+exp(`lgodds_m2_01’))
        gen byte M2_01=rbinomial(1,PM2_01)
```

```
local lgodds_m2_10 “-log(32)+log(3)*M1_0+log(4)*C4+log(4)*C3+log(3)*1+log(2)*1*M1_0”
        // plug in A = 1, M = M1_0

        gen double PM2_10=exp(`lgodds_m2_10’)/(1+exp(`lgodds_m2_10’))

        gen byte M2_10=rbinomial(1,PM2_10)

local lgodds_m2_11 “-log(32)+log(3)*M1_1+log(4)*C4+log(4)*C3+log(3)*1+log(2)*1*M1_1”
        // plug in A = 1, M = M1_1

        gen double PM2_11=exp(`lgodds_m2_11’)/(1+exp(`lgodds_m2_11’))

        gen byte M2_11=rbinomial(1,PM2_11)

/*-------------------------------------------------------------------------------*/

// Generate counterfactual Outcome Y:

// Y(0,M1(0),M2(0,M1(0)))

// Y(1,M1(0),M2(0,M1(0)))

// Y(1,M1(0),M2(1,M1(0)))

// Y(1,M1(1),M2(1,M1(1)))

/*-------------------------------------------------------------------------------*/

local lgodds_y_0000 “-log(32)+log(4)*C3+log(4)*M1_0+log(4)*M2_00+log(4)*C5+log(4)*C2+log(3)*0+log(4)*M1_0*M2_00+log(3)*0*M1_0+log(4)*0*M2_00”

        gen double PY_0000 = exp(`lgodds_y_0000’)/(1+exp(`lgodds_y_0000’))

        gen byte Y_0000 = rbinomial(1,PY_0000)

local lgodds_y_1000 “-log(32)+log(4)*C3+log(4)*M1_0+log(4)*M2_00+log(4)*C5+log(4)*C2+log(3)*1+log(4)*M1_0*M2_00+log(3)*1*M1_0+log(4)*1*M2_00”

        gen double PY_1000 = exp(`lgodds_y_1000’)/(1+exp(`lgodds_y_1000’))

        gen byte Y_1000 = rbinomial(1,PY_1000)

local lgodds_y_1010 “-log(32)+log(4)*C3+log(4)*M1_0+log(4)*M2_10+log(4)*C5+log(4)*C2+log(3)*1+log(4)*M1_0*M2_10+log(3)*1*M1_0+log(4)*1*M2_10”

        gen double PY_1010 = exp(`lgodds_y_1010’)/(1+exp(`lgodds_y_1010’))

        gen byte Y_1010 = rbinomial(1,PY_1010)
```

```
local lgodds_y_1111 “-log(32)+log(4)*C3+log(4)*M1_1+log(4)*M2_11+log(4)*C5+log(4)*C2+log(3)*1+log(4)*M1_1*M2_11+log(3)*1*M1_1+log(4)*1*M2_11”

        gen double PY_1111 = exp(`lgodds_y_1111’)/(1+exp(`lgodds_y_1111’))

        gen byte Y_1111 = rbinomial(1,PY_1111)

/*------------------------------------------------------------------------------*/

// Total effect (TE)

/*------------------------------------------------------------------------------*/

sum Y_1111

local m1111 = r(mean)

sum Y_0000

local m0000 = r(mean)

dis “. TE in risk difference scale = “ `m1111’-`m0000’

dis “. TE in risk ratio scale = “ `m1111’/`m0000’

/*------------------------------------------------------------------------------*/

// Natural direct effect (NDE)

/*------------------------------------------------------------------------------*/

sum Y_1000

local m1000 = r(mean)

dis “. NDE in risk difference scale = “ `m1000’-`m0000’

dis “. NDE in risk ratio scale = “ `m1000’/`m0000’

/*------------------------------------------------------------------------------*/

// Natural indirect effect of M1 (NIE1)

/*------------------------------------------------------------------------------*/

sum Y_1010

local m1010 = r(mean)

dis “. NIE1 in risk difference scale = “ `m1111’-`m1010’

dis “. NIE1 in risk ratio scale = “ `m1111’/`m1010’

/*------------------------------------------------------------------------------*/
```

```
// Partial indirect effect of M2 (PIE2)

/*----------------------------------------------------------------------------*/

dis “. PIE2 in risk difference scale = “ `m1010’-`m1000’

dis “. PIE2 in risk ratio scale = “ `m1010’/`m1000’

/*----------------------------------------------------------------------------*/

// Program DGP for Simulation Scenarios

/*----------------------------------------------------------------------------*/

cap program drop dgp

program define dgp

	gen ID=_n

	gen byte C1=rbinomial(1,0.5)

	gen byte C2=rbinomial(1,0.5)

	gen byte C3=rbinomial(1,0.5)

	gen byte C4=rbinomial(1,0.5)

	gen byte C5=rbinomial(1,0.5)

local lgodds_a “-log(16)+log(4)*C1+log(4)*C2”

	gen double Pa=exp(`lgodds_a’)/(1+exp(`lgodds_a’))

	gen byte A=rbinomial(1,Pa)

local lgodds_m1 “-log(32)+log(4)*C1+log(4)*C4+log(4)*C5+log(4)*A”

	gen double PM1=exp(`lgodds_m1’)/(1+exp(`lgodds_m1’))

	gen byte M1=rbinomial(1,PM1)

local lgodds_m2 “-log(32)+log(3)*M1+log(4)*C4+log(4)*C3+log(3)*A+log(2)*A*M1”

	gen double PM2=exp(`lgodds_m2’)/(1+exp(`lgodds_m2’))

	gen byte M2=rbinomial(1,PM2)

local lgodds y “-
log(32)+log(4)*C3+log(4)*M1+log(4)*M2+log(4)*C5+log(4)*C2+log(3)*A+log(4)*M1*M2+log(3)
*A*M1+log(4)*A*M2”

	gen double PY=exp(`lgodds_y’)/(1+exp(`lgodds_y’))

	gen byte Y=rbinomial(1,PY)
```

```
end
```

APPENDIX C: Tabulated Performance Metrics by Method and Scale

Table A1. Performance of Lange *et al.* in Risk Difference Scale.

True Effects	Model	Metric	True[a]	Scen. 1[b]	Scen. 2[c]	Scen. 3[d]
NDE_{RD} = .216	Linear	Bias %	3.028 (18.486)	-5.894 (17.610)	-2.048 (18.201)	-5.484 (19.490)
		RMSE	0.032 (0.024)	0.032 (0.024)	0.032 (0.024)	0.035 (0.026)
		Coverage	94.3%	95.0%	95.5%	94.2%
	GLM	Bias %	2.334 (18.227)	-6.783 (17.324)	-2.873 (17.897)	-6.109 (19.130)
		RMSE	0.032 (0.024)	0.032 (0.024)	0.031 (0.023)	0.035 (0.026)
		Coverage	94.6%	94.8%	95.2%	94.0%
NIE_{RD} = .088	Linear	Bias %	6.652 (19.470)	28.203 (19.321)	7.325 (19.590)	3.552 (19.153)
		RMSE	0.014 (0.011)	0.026 (0.016)	0.014 (0.012)	0.013 (0.011)
		Coverage	88.0%	47.3%	87.6%	89.6%
	GLM	Bias %	2.382 (19.248)	23.813 (19.016)	2.549 (19.371)	0.598 (19.119)
		RMSE	0.013 (0.011)	0.022 (0.015)	0.014 (0.011)	0.013 (0.010)
		Coverage	89.8%	57.4%	89.8%	89.5%
PIE_{RD} = .059	Linear	Bias %	-10.825 (27.147)	-10.949 (26.112)	7.745 (28.703)	20.315 (28.165)
		RMSE	0.014 (0.010)	0.013 (0.010)	0.014 (0.011)	0.016 (0.012)
		Coverage	84.0%	83.4%	83.8%	86.4%
	GLM	Bias %	-5.786 (26.986)	-5.685 (25.780)	13.372 (28.622)	23.885 (27.971)
		RMSE	0.013 (0.010)	0.013 (0.009)	0.015 (0.011)	0.017 (0.013)
		Coverage	85.3%	85.8%	80.5%	82.3%
TE_{RD} = .363	Linear	Bias %	1.661 (8.814)	1.588 (8.841)	1.825 (8.847)	0.906 (8.830)
		RMSE	0.026 (0.020)	0.026 (0.020)	0.026 (0.020)	0.026 (0.020)
		Coverage	93.7%	93.6%	93.5%	94.0%
	GLM	Bias %	1.027 (8.677)	0.846 (8.677)	1.086 (8.680)	0.395 (8.666)
		RMSE	0.025 (0.019)	0.025 (0.019)	0.025 (0.019)	0.025 (0.019)
		Coverage	94.1%	94.3%	94.0%	94.4%

Note: Values in parentheses are the standard deviations of the measure of interest from 2000 replicates.

Abbreviations: NDE, natural direct effect; NIE, natural indirect effect; PIE, partial indirect effect; TE, total effect; RD, risk difference; GLM, generalized linear model; RMSE, root mean square error.

[a] All models correctly specified according to the DGP.

[b] M_1 mis-specified due to unmeasured confounding.

[c] M_2 mis-specified due to unmeasured confounding.

[d] M_2 mis-specified due to lack of $A*M_1$ interaction.

Table A2. Performance of Lange *et al.* in Risk Ratio Scale.

True Effects	Model	Metric	True[a]	Scen. 1[b]	Scen. 2[c]	Scen. 3[d]
NDE_{RR} = 1.57	GLM	Bias %	1.230 (7.219)	-2.168 (6.820)	-0.670 (7.075)	-2.032 (7.412)
		RMSE	0.091 (0.071)	0.091 (0.067)	0.089 (0.067)	0.097 (0.072)
		Coverage	94.9%	95.0%	95.0%	93.9%
NIE_{RR} = 1.14	GLM	Bias %	0.422 (2.836)	3.444 (2.964)	0.444 (2.853)	0.189 (2.803)
		RMSE	0.025 (0.021)	0.042 (0.030)	0.025 (0.021)	0.025 (0.020)
		Coverage	91.4%	69.5%	91.5%	91.0%
PIE_{RR} = 1.10	GLM	Bias %	-0.467 (2.713)	-0.182 (2.663)	1.470 (2.957)	2.587 (3.118)
		RMSE	0.024 (0.018)	0.023 (0.018)	0.028 (0.023)	0.035 (0.028)
		Coverage	86.5%	88.0%	85.8%	87.9%
TE_{RR} = 1.97	GLM	Bias %	0.993 (5.394)	0.840 (5.378)	1.043 (5.402)	0.461 (5.336)
		RMSE	0.085 (0.066)	0.085 (0.065)	0.085 (0.066)	0.084 (0.064)
		Coverage	94.3%	94.3%	94.3%	94.5%

Note: Values in parentheses are the standard deviations of the measure of interest from 2000 replicates.

Abbreviations: NDE, natural direct effect; NIE, natural indirect effect; PIE, partial indirect effect; TE, total effect; RR, risk ratio; GLM, generalized linear model; RMSE, root mean square error.

[a] All models correctly specified according to the DGP.

[b] M_1 mis-specified due to unmeasured confounding.

[c] M_2 mis-specified due to unmeasured confounding.

[d] M_2 mis-specified due to lack of $A*M_1$ interaction.

Table A3. Performance of Steen *et al.* in Risk Difference Scale.

True Effects	Model	Metric	True[a]	Scen. 1[b]	Scen. 2[c]	Scen. 3[d]	Scen. 4[e]	Scen. 5[f]
NDE_{RD} = .216	Linear	Bias %	2.849 (12.430)	0.983 (12.387)	3.272 (12.449)	3.087 (12.407)	23.932 (13.002)	1.332 (12.381)
		RMSE	0.022 (0.017)	0.021 (0.016)	0.022 (0.017)	0.022 (0.017)	0.052 (0.027)	0.021 (0.016)
		Covg.	24.1%	21.6%	24.3%	24.7%	4.3%	22.2%
	GLM	Bias %	2.136 (12.274)	0.410 (12.247)	2.778 (12.302)	2.424 (12.248)	22.632 (12.825)	0.508 (12.215)
		RMSE	0.021 (0.017)	0.021 (0.016)	0.021 (0.017)	0.021 (0.017)	0.050 (0.026)	0.021 (0.016)
		Covg.	25.4%	22.3%	26.4%	26.5%	4.8%	23.1%
NIE_{RD} = .088	Linear	Bias %	6.713 (16.800)	24.173 (17.808)	-2.650 (25.622)	-10.094 (25.758)	10.928 (16.338)	3.283 (16.500)
		RMSE	0.013 (0.010)	0.022 (0.014)	0.018 (0.014)	0.020 (0.015)	0.014 (0.011)	0.012 (0.009)
		Covg.	71.7%	32.0%	96.0%	94.5%	62.5%	74.5%
	GLM	Bias %	2.363 (16.826)	21.694 (17.873)	-7.170 (24.932)	-14.717 (25.298)	5.113 (16.226)	0.049 (16.621)
		RMSE	0.012 (0.009)	0.021 (0.014)	0.018 (0.014)	0.021 (0.015)	0.012 (0.009)	0.012 (0.009)
		Covg.	76.1%	38.9%	95.8%	91.8%	71.8%	76.6%
PIE_{RD} = .059	Linear	Bias %	-10.967 (42.001)	-35.915 (40.386)	3.076 (26.133)	14.232 (23.575)	-20.297 (33.643)	-4.590 (42.593)
		RMSE	0.020 (0.016)	0.026 (0.019)	0.012 (0.010)	0.012 (0.010)	0.018 (0.014)	0.020 (0.016)
		Covg.	94.4%	88.1%	46.7%	44.5%	90.6%	95.2%
	GLM	Bias %	-5.819 (42.897)	-32.802 (41.252)	8.154 (26.342)	19.642 (23.828)	-12.842 (34.661)	-0.397 (43.424)
		RMSE	0.020 (0.016)	0.025 (0.018)	0.012 (0.010)	0.014 (0.011)	0.017 (0.013)	0.020 (0.016)
		Covg.	95.0%	90.0%	45.4%	39.0%	93.7%	95.0%
TE_{RD} = .363	Linear	Bias %	1.546 (8.832)	0.638 (8.752)	1.798 (8.851)	1.687 (8.833)	13.583 (8.093)	0.845 (8.746)
		RMSE	0.025 (0.021)	0.024 (0.021)	0.025 (0.021)	0.025 (0.021)	0.051 (0.026)	0.024 (0.021)
		Covg.	84.5%	85.6%	84.4%	84.7%	31.1%	85.5%
	GLM	Bias %	0.900 (8.681)	0.200 (8.611)	1.229 (8.675)	1.046 (8.681)	12.604 (7.913)	0.249 (8.603)
		RMSE	0.024 (0.021)	0.023 (0.021)	0.024 (0.021)	0.024 (0.021)	0.048 (0.025)	0.023 (0.021)
		Covg.	85.2%	85.6%	84.8%	85.0%	33.9%	85.7%

Table A3 (cont'd)

Note: Values in parentheses are the standard deviations of the measure of interest from 2000 replicates.

Abbreviations: NDE, natural direct effect; NIE, natural indirect effect; PIE, partial indirect effect; TE, total effect; RD, risk difference; GLM, generalized linear model; RMSE, root mean square error; Covg., coverage.

[a] All models correctly specified according to the DGP.

[b] M_1 mis-specified due to unmeasured confounding.

[c] M_2 mis-specified due to unmeasured confounding.

[d] M_2 mis-specified due to lack of $A*M_1$ interaction.

[e] Y mis-specified due to unmeasured confounding.

[f] Y mis-specified due to lack of M_1*M_2 interaction.

Table A4. Performance of Steen *et al.* in Risk Ratio Scale.

True Effects	Model	Metric	True[a]	Scen. 1[b]	Scen. 2[c]	Scen. 3[d]	Scen. 4[e]	Scen. 5[f]
NDE$_{RR}$ = 1.57	GLM	Bias %	1.158 (5.180)	0.313 (5.106)	1.482 (5.220)	1.301 (5.174)	10.464 (5.728)	0.361 (5.095)
		RMSE	0.066 (0.052)	0.064 (0.049)	0.067 (0.053)	0.066 (0.052)	0.166 (0.087)	0.064 (0.049)
		Coverage	34.8%	32.0%	35.3%	35.0%	5.5%	32.5%
NIE$_{RR}$ = 1.14	GLM	Bias %	0.400 (2.453)	3.136 (2.730)	-0.918 (3.171)	-1.920 (3.129)	0.235 (2.290)	0.091 (2.416)
		RMSE	0.022 (0.018)	0.038 (0.028)	0.030 (0.022)	0.034 (0.024)	0.020 (0.016)	0.022 (0.017)
		Coverage	76.9%	46.3%	94.1%	89.8%	71.3%	76.8%
PIE$_{RR}$ = 1.10	GLM	Bias %	-0.521 (3.993)	-2.949 (3.829)	0.730 (2.534)	1.779 (2.367)	-1.543 (3.080)	-0.005 (4.059)
		RMSE	0.035 (0.027)	0.043 (0.032)	0.022 (0.019)	0.025 (0.021)	0.030 (0.023)	0.035 (0.028)
		Coverage	94.5%	89.5%	44.0%	38.7%	91.3%	95.2%
TE$_{RR}$ = 1.97	GLM	Bias %	0.935 (5.360)	0.305 (5.252)	1.214 (5.389)	1.058 (5.365)	8.922 (5.400)	0.347 (5.253)
		RMSE	0.083 (0.068)	0.080 (0.066)	0.084 (0.069)	0.083 (0.068)	0.180 (0.098)	0.080 (0.066)
		Coverage	82.8%	83.0%	82.8%	82.8%	28.5%	83.2%

Note: Values in parentheses are the standard deviations of the measure of interest from 2000 replicates.

Abbreviations: NDE, natural direct effect; NIE, natural indirect effect; PIE, partial indirect effect; TE, total effect; RD, risk difference; GLM, generalized linear model; RMSE, root mean square error; Covg., coverage.

[a] All models correctly specified according to the DGP.

[b] M_1 mis-specified due to unmeasured confounding.

[c] M_2 mis-specified due to unmeasured confounding.

[d] M_2 mis-specified due to lack of $A*M_1$ interaction.

[e] Y mis-specified due to unmeasured confounding.

[f] Y mis-specified due to lack of M_1*M_2 interaction.

APPENDIX D: Equations for Data Generating Procedure

$$C_1, C_2, C_3, C_4, C_5 \overset{i.i.d.}{\sim} Bern(1, 0.5)$$

$$(1)\ P(A) = \frac{\exp(-\log(16) + \log(4)\ C1 + \log(4)\ C2)}{1 + \exp(-\log(16) + \log(4)\ C1 + \log(4)\ C2)}; A{\sim}Bern(1, P(A))$$

$$(2)\ P(M1) = \frac{exp(-log(32) + log(4)C1 + log(4)C4 + log(4)C5 + log(4)A)}{1 + exp(-log(32) + log(4)C1 + log(4)C4 + log(4)C5 + log(4)A)}; M1{\sim}Bern(1, P(M1))$$

$$(3)\ P(M2) = \frac{exp(-log(32) + log(3)M1 + log(4)C4 + log(4)C3 + log(3)A + log(2)AM1)}{1 + exp(-log(32) + log(3)M1 + log(4)C4 + log(4)C3 + log(3)A + log(2)AM1)};\ M2{\sim}Bern(1, P(M2))$$

$$\begin{aligned}&(4)\ P(Y)\\&= \frac{exp(-log(32) + log(4)C3 + log(4)M1 + log(4)M2 + log(4)C5 + log(4)C2 + log(3)A + log(4)M1M2 + log(3)AM1 + log(4)AM2)}{(1 + exp(-log(32) + log(4)C3 + log(4)M1 + log(4)M2 + log(4)C5 + log(4)C2 + log(3)A + log(4)M1M2 + log(3)AM1 + log(4)AM2))};\ Y{\sim}Bern(1, P(Y))\end{aligned}$$

APPENDIX E: Stata Code for Properly Specified Scenarios

```
/*Lange True Scenario*/
capture program drop LangeTrue
program define LangeTrue, rclass
        drop _all
        set obs 2000
/*DGP*/
        dgp
/*Create Expanded Dataset*/
        expdata_lange
/*Correctly specified models*/
        gen A_temp=A
        mlogit M1 C1 C4 C5 A_temp if originalObs==1
        predict out0 out1, p
        gen w1denom=out0 if M1==0
        replace w1denom=out1 if M1==1
        drop out0 out1
        replace A_temp=AStar1
        predict out0 out1, p
        gen w1num=out0 if M1==0
        replace w1num=out1 if M1==1
        drop out0 out1
        drop A_temp
        gen A_temp=A
        mlogit M2 C3 C4 i.M1##i.A_temp if originalObs==1
        predict out0 out1, p
        gen w2denom=out0 if M2==0
        replace w2denom=out1 if M2==1
```

```
        drop out0 out1
        replace A_temp=AStar2
        predict out0 out1, p
        gen w2num=out0 if M2==0
        replace w2num=out1 if M2==1
        drop out0 out1
/*Compute Conditional Weight*/
        gen W=(w1num/w1denom)*(w2num/w2denom)
/*Generating Marginal Weight*/
        mlogit A C1 C2 if originalObs==1
        predict out0 out1
        gen Wa=1/out0 if A==0
        replace Wa=1/out1 if A==1
        gen MW=Wa*W
/*Return Mean Weight*/
        sum MW
        return scalar mweight = r(mean)
/*Marginal Structural Model - Linear Probability*/
        rename A a0
        rename AStar1 a1
        rename AStar2 a2
        regress Y i.a0 i.a1 i.a2 i.a0#i.a1 i.a0#i.a2 i.a1#i.a2 [pweight=MW], vce(cluster ID)
        mat b = r(table)
        *NDE*
        local nde = _b[1.a0]
        local lb = b[5,2]
        local ub = b[6,2]
        return scalar nde_l_rd= `nde'
```

```
	return scalar nde_l_rd_ll = `lb'
	return scalar nde_l_rd_ul = `ub'
	*NIE1*
	lincom _b[1.a1] + _b[1.a0#1.a1] + _b[1.a1#1.a2]
	local nie = r(estimate)
	local lb = r(lb)
	local ub = r(ub)
	return scalar nie_l_rd= `nie'
	return scalar nie_l_rd_ll = `lb'
	return scalar nie_l_rd_ul = `ub'
	*PIE2*
	lincom _b[1.a2] + _b[1.a0#1.a2]
	local pie = r(estimate)
	local lb = r(lb)
	local ub = r(ub)
	return scalar pie_l_rd= `pie'
	return scalar pie_l_rd_ll = `lb'
	return scalar pie_l_rd_ul = `ub'
	*TE*
	lincom _b[1.a0] + _b[1.a1] + _b[1.a2] + _b[1.a0#1.a1] + _b[1.a0#1.a2] + _b[1.a1#1.a2]
	return scalar te_l_rd= r(estimate)
	return scalar te_l_rd_ll = r(lb)
	return scalar te_l_rd_ul = r(ub)
/*Marginal Structural Model - Generalized Linear Model*/
	glm Y i.a0 i.a1 i.a2 i.a0#i.a1 i.a0#i.a2 i.a1#i.a2 [pweight=MW], vce(cluster ID) fam(bin) link(log) eform
	*NDE*
	margins r.a0, at(a1=0 a2=0)
	mat b = r(table)
```

```
local nde = b[1,1]

local lb = b[5,1]

local ub = b[6,1]

return scalar nde_nl_rd = `nde'

return scalar nde_nl_rd_ll =`lb'

return scalar nde_nl_rd_ul = `ub'

margins, expression(exp(_b[1.a0]))

mat b = r(table)

local nde = b[1,1]

local lb = b[5,1]

local ub = b[6,1]

return scalar nde_nl_rr = `nde'

return scalar nde_nl_rr_ll = `lb'

return scalar nde_nl_rr_ul = `ub'

*NIE1*

margins r.a1, at(a0=1 a2=1)

mat b = r(table)

local nie = b[1,1]

local lb = b[5,1]

local ub = b[6,1]

return scalar nie_nl_rd = `nie'

return scalar nie_nl_rd_ll = `lb'

return scalar nie_nl_rd_ul = `ub'

margins, expression(exp(_b[1.a1] + _b[1.a0#1.a1] + _b[1.a1#1.a2]))

mat b = r(table)

local nie = b[1,1]

local lb = b[5,1]

local ub = b[6,1]
```

```
return scalar nie_nl_rr = `nie'
return scalar nie_nl_rr_ll = `lb'
return scalar nie_nl_rr_ul = `ub'
*PIE2*
margins r.a2, at(a1=0 a0=1)
mat b = r(table)
local pie = b[1,1]
local lb = b[5,1]
local ub = b[6,1]
return scalar pie_nl_rd = `pie'
return scalar pie_nl_rd_ll = `lb'
return scalar pie_nl_rd_ul = `ub'
margins, expression(exp(_b[1.a2] + _b[1.a0#1.a2]))
mat b = r(table)
local pie = b[1,1]
local lb = b[5,1]
local ub = b[6,1]
return scalar pie_nl_rr = `pie'
return scalar pie_nl_rr_ll = `lb'
return scalar pie_nl_rr_ul = `ub'
*TE*
margins, expression(exp(_b[_cons] + _b[1.a0] + _b[1.a1] + _b[1.a2] + _b[1.a0#1.a1] + _b[1.a0#1.a2] + _b[1.a1#1.a2]) - exp(_b[_cons]))
mat b = r(table)
local te = b[1,1]
local lb = b[5,1]
local ub = b[6,1]
return scalar te_nl_rd = `te'
return scalar te_nl_rd_ll = `lb'
```

```
	return scalar te_nl_rd_ul = `ub'
	margins, expression(exp(_b[1.a0] + _b[1.a1] + _b[1.a2] + _b[1.a0#1.a1] + _b[1.a0#1.a2] + _b[1.a1#1.a2]))
	mat b = r(table)
	local te = b[1,1]
	local lb = b[5,1]
	local ub = b[6,1]
	return scalar te_nl_rr = `te'
	return scalar te_nl_rr_ll = `lb'
	return scalar te_nl_rr_ul = `ub'
end
/* Steen True Scenario */
capture program drop SteenTrue
program define SteenTrue, rclass
	drop _all
	set obs 2000
/*DGP*/
	dgp
*Step 0 - Estimate model for exposure A*
	mlogit A C1 C2
	estimates store A
*Step 1 - Estimate model for mediator M1 or mediator M2*
	logit M1 C1 C4 C5 A
	estimates store M1
*Step 2 - Estimate model for outcome Y*
	logit Y C2 C3 C5 i.A i.M1 i.M2 i.A#i.M1 i.A#i.M2 i.M1#i.M2
	estimates store Y
*Step 3 M1 - Create expanded dataset using M1*
	expand 2
```

```
    bys ID: gen int a0 = (_n-1)
    expand 2
    bys ID a0: gen int a1 = (_n-1)
    gen a2 = A
*Step 4A M1 - Calculate weights on M1 model for conditional effects*
    estimates restore M1
    tempvar temp_A
    gen `temp_A' = A
    replace A = a1
    predict double num_M1
    replace num_M1 = 1-num_M1 if M1==0
    replace A = `temp_A'
    predict double den_M1
    replace den_M1 = 1-den_M1 if M1==0
    gen double W_M1 = num_M1/den_M1
*Step 4B M1 - Calculate weights for marginal effects related to M1*
    estimates restore A
    predict predout0 predout1
    gen double W_a = 1/predout0 if A==0
    replace W_a = 1/predout1 if A==1
    gen double W_marg = W_M1*W_a
    *Return mean weight*
    sum W_marg
    return scalar mweight = r(mean)
*Step 5 - Impute outcome Y using expanded dataset for M1*
    estimates restore Y
    tempvar temp_A
    gen `temp_A' = A
```

```
        replace A = a0
        predict fitY
        replace A = `temp_A'
*Marginal Structural Model - Linear Probability*
        regress fitY i.a0 i.a1 i.a2 i.a0#i.a1 i.a0#i.a2 i.a1#i.a2 [pweight=W_marg], vce(cluster ID)
        mat b = r(table)
        *NDE*
        local nde = _b[1.a0]
        local lb = b[5,2]
        local ub = b[6,2]
        return scalar nde_l_rd= `nde'
        return scalar nde_l_rd_ll = `lb'
        return scalar nde_l_rd_ul = `ub'
        *NIE1*
        lincom _b[1.a1] + _b[1.a0#1.a1] + _b[1.a1#1.a2]
        local nie = r(estimate)
        local lb = r(lb)
        local ub = r(ub)
        return scalar nie_l_rd= `nie'
        return scalar nie_l_rd_ll = `lb'
        return scalar nie_l_rd_ul = `ub'
        *PIE2*
        lincom _b[1.a2] + _b[1.a0#1.a2]
        local pie = r(estimate)
        local lb = r(lb)
        local ub = r(ub)
        return scalar pie_l_rd= `pie'
        return scalar pie_l_rd_ll = `lb'
```

```
	return scalar pie_l_rd_ul = `ub'
	*TE*
	lincom _b[1.a0] + _b[1.a1] + _b[1.a2] + _b[1.a0#1.a1] + _b[1.a0#1.a2] + _b[1.a1#1.a2]
	return scalar te_l_rd= r(estimate)
	return scalar te_l_rd_ll = r(lb)
	return scalar te_l_rd_ul = r(ub)
*Marginal Structural Model - Generalized Linear Model*
	glm fitY i.a0 i.a1 i.a2 i.a0#i.a1 i.a0#i.a2 i.a1#i.a2 [pweight=W_marg], vce(cluster ID) family(bin) link(log) eform
	*NDE*
	margins r.a0, at(a1=0 a2=0)
	mat b = r(table)
	local nde = b[1,1]
	local lb = b[5,1]
	local ub = b[6,1]
	return scalar nde_nl_rd = `nde'
	return scalar nde_nl_rd_ll =`lb'
	return scalar nde_nl_rd_ul = `ub'
	margins, expression(exp(_b[1.a0]))
	mat b = r(table)
	local nde = b[1,1]
	local lb = b[5,1]
	local ub = b[6,1]
	return scalar nde_nl_rr = `nde'
	return scalar nde_nl_rr_ll = `lb'
	return scalar nde_nl_rr_ul = `ub'
	*NIE1*
	margins r.a1, at(a0=1 a2=1)
	mat b = r(table)
```

```
local nie = b[1,1]
local lb = b[5,1]
local ub = b[6,1]
return scalar nie_nl_rd = `nie'
return scalar nie_nl_rd_ll = `lb'
return scalar nie_nl_rd_ul = `ub'
margins, expression(exp(_b[1.a1] + _b[1.a0#1.a1] + _b[1.a1#1.a2]))
mat b = r(table)
local nie = b[1,1]
local lb = b[5,1]
local ub = b[6,1]
return scalar nie_nl_rr = `nie'
return scalar nie_nl_rr_ll = `lb'
return scalar nie_nl_rr_ul = `ub'
*PIE2*
margins r.a2, at(a1=0 a0=1)
mat b = r(table)
local pie = b[1,1]
local lb = b[5,1]
local ub = b[6,1]
return scalar pie_nl_rd = `pie'
return scalar pie_nl_rd_ll = `lb'
return scalar pie_nl_rd_ul = `ub'
margins, expression(exp(_b[1.a2] + _b[1.a0#1.a2]))
mat b = r(table)
local pie = b[1,1]
local lb = b[5,1]
local ub = b[6,1]
```

```
	return scalar pie_nl_rr = `pie'
	return scalar pie_nl_rr_ll = `lb'
	return scalar pie_nl_rr_ul = `ub'
	*TE*
	margins, expression(exp(_b[_cons] + _b[1.a0] + _b[1.a1] + _b[1.a2] + _b[1.a0#1.a1] + 
_b[1.a0#1.a2] + _b[1.a1#1.a2]) - exp(_b[_cons]))
	mat b = r(table)
	local te = b[1,1]
	local lb = b[5,1]
	local ub = b[6,1]
	return scalar te_nl_rd = `te'
	return scalar te_nl_rd_ll = `lb'
	return scalar te_nl_rd_ul = `ub'
	margins, expression(exp(_b[1.a0] + _b[1.a1] + _b[1.a2] + _b[1.a0#1.a1] + _b[1.a0#1.a2] 
+ _b[1.a1#1.a2]))
	mat b = r(table)
	local te = b[1,1]
	local lb = b[5,1]
	local ub = b[6,1]
	return scalar te_nl_rr = `te'
	return scalar te_nl_rr_ll = `lb'
	return scalar te_nl_rr_ul = `ub'
end
```

REFERENCES

REFERENCES

1. Richiardi L, Bellocco R, Zugna D. Mediation analysis in epidemiology: methods, interpretation and bias. *International Journal of Epidemiology*. 2013;42(5):1511–1519.

2. Lange T, Rasmussen M, Thygesen LC. Assessing Natural Direct and Indirect Effects Through Multiple Pathways. *American Journal of Epidemiology*. 2014;179(4):513–518.

3. Steen J, Loeys T, Moerkerke B, et al. Flexible Mediation Analysis With Multiple Mediators. *American Journal of Epidemiology*. 2017;186(2):184–193.

4. Baron RM, Kenny DA. The Moderator-Mediator Variable Distinction in Social Psychological Research: Conceptual, Strategic, and Statistical Considerations. :10.

5. Vanderweele TJ, Vansteelandt S. Conceptual issues concerning mediation, interventions and composition. *Statistics and Its Interface*. 2009;2(4):457–468.

6. MacKinnon DP, Fairchild AJ, Fritz MS. Mediation Analysis. *Annual Review of Psychology*. 2007;58(1):593–614.

7. Lange T, Vansteelandt S, Bekaert M. A Simple Unified Approach for Estimating Natural Direct and Indirect Effects. *American Journal of Epidemiology*. 2012;176(3):190–195.

8. Daniel RM, De Stavola BL, Cousens SN, et al. Causal mediation analysis with multiple mediators: Causal Mediation Analysis with Multiple Mediators. *Biometrics*. 2015;71(1):1–14.

9. Liu B, Luo Z, Pinto JM, et al. Relationship between poor olfaction and mortality among community-dwelling older adults: a cohort study. 2019;

10. Lange T, Hansen JV. Direct and Indirect Effects in a Survival Context: *Epidemiology*. 2011;22(4):575–581.

11. Tchetgen Tchetgen EJ, VanderWeele TJ. Identification of Natural Direct Effects When a Confounder of the Mediator Is Directly Affected by Exposure: *Epidemiology*. 2014;25(2):282–291.

12. Pearl J. Causality: Models, Reasoning and Inference. 2nd ed. New York, NY: Cambridge University Press; 2009.

13. Robins JM, Greenland S. Identifiability and Exchangeability for Direct and Indirect Effects. *Epidemiology*. 1992;3(2):143–155.

14. Pearl J. Direct and Indirect Effects. San Francisco, CA: University of California; 2001.

15. Bellavia A, Valeri L. Decomposition of the Total Effect in the Presence of Multiple Mediators and Interactions. *American Journal of Epidemiology*. 2018;187(6):1311–1318.

16. Hafeman DM, Schwartz S. Opening the Black Box: a motivation for the assessment of mediation. *International Journal of Epidemiology*. 2009;38(3):838–845.