

Article

# Testing and Interpreting Latent Variable Interactions Using the `semTools` Package

Alexander M. Schoemann <sup>1,\*</sup>  and Terrence D. Jorgensen <sup>2</sup> <sup>1</sup> Department of Psychology, East Carolina University, Rawl Building, Greenville, NC 27858, USA<sup>2</sup> Research Institute of Child Development and Education, University of Amsterdam, 1012 WX Amsterdam, The Netherlands; T.D.Jorgensen@uva.nl

\* Correspondence: schoemanna@ecu.edu

**Abstract:** Examining interactions among predictors is an important part of a developing research program. Estimating interactions using latent variables provides additional power to detect effects over testing interactions in regression. However, when predictors are modeled as latent variables, estimating and testing interactions requires additional steps beyond the models used for regression. We review methods of estimating and testing latent variable interactions with a focus on product indicator methods. Product indicator methods of examining latent interactions provide an accurate method to estimate and test latent interactions and can be implemented in any latent variable modeling software package. Significant latent interactions require additional steps (plotting and probing) to interpret interaction effects. We demonstrate how these methods can be easily implemented using functions in the `semTools` package with models fit using the `lavaan` package in R, and we illustrate how these methods work using an applied example concerning teacher stress and testing.



**Citation:** Schoemann, A.M.; Jorgensen, T.D. Testing and Interpreting Latent Variable Interactions Using the `semTools` Package. *Psych* **2021**, *3*, 322–335. <https://doi.org/10.3390/psych3030024>

Academic Editor: Alexander Robitzsch

Received: 25 May 2021  
Accepted: 10 July 2021  
Published: 30 July 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** structural equation modeling; latent variable; interactions; moderation

## 1. Introduction

Phenomena are rarely influenced by only one variable; rather, it is often the case that the effect of one variable on another depends on the level of third variable or even two other variables. Interactions, or moderation, occurs when the strength or sign of a relationship between two variables,  $x$  and  $y$ , depends on the value of a third variable,  $z$ . Modeling moderation is critical for advancing research programs by moving beyond simple questions such as “is  $x$  related to  $y$ ?” to address more nuanced issues, such as “when is  $x$  related to  $y$ ?” or “for whom is  $x$  related to  $y$ ?”.

There is a long tradition of evaluating moderation with regression models [1–5]. In a regression model, moderation is represented with a statistical interaction by calculating the product of the two predictors of interest and including the product as an additional predictor in the model. A structural equation model (SEM) can include regressions among latent variables (e.g., common factors or growth factors), which are unobserved. Without observing individual subjects’ values (factor scores), it is not possible to multiply latent factor scores to calculate a product term to be included in the model, and thus other methods must be implemented [6–10].

Latent interaction models provide an important benefit over regression models in testing interactions: the ability to model measurement error in observed variables. By explicitly modeling measurement error in observed variables latent interaction models provide increased power to detect interactions effects over regression models [11]. This paper discusses product–indicator methods for estimating interaction effects among latent predictors and how to evaluate moderation hypotheses using the statistical software R [12].

### 1.1. Interaction among Observed Variables

Numerous resources are available about testing and interpreting interactions in regression models, including several books [1–4] and an online primer available from Kris Preacher's website <http://quantpsy.org/medn.htm>, accessed on 9 July 2021. Here, we provide a summary of the key steps. In the simplest case, an outcome variable  $y$  is predicted by two continuous variables,  $x$  and  $z$ , in a regression model with an interaction between  $x$  and  $z$ :

$$y_i = b_0 + b_1x_i + b_2z_i + b_3x_iz_i + e_i. \quad (1)$$

In this model,  $b_3$  is an estimate of the interaction effect in the population ( $\beta_3$ ), and the null hypothesis  $H_0 : \beta_3 = 0$  represents a hypothesis of no interaction between  $x$  and  $z$ . The  $H_0$  can be tested by dividing  $b_3$  by its  $SE$  (a Wald-type statistic), which could be a  $t$  statistic using ordinary least squares (OLS) estimation or a  $z$  statistic using maximum likelihood estimation (MLE). If the  $H_0$  is rejected at a given  $\alpha$  level, one is left with the alternative hypothesis that moderation occurs.

When  $z$  moderates the effect of  $x$  on  $y$  in Equation (1), it would be misleading to report and interpret a single effect of  $x$  because its effect would vary across levels of  $z$ . The estimated effect  $b_1$  is the conditional effect of  $x$  when  $z = 0$ , also called a simple slope ([2] chapter 7). The interaction effect ( $b_3$ ) is the degree to which a simple slope changes with each 1-unit increase in the moderator; therefore, other simple slopes can be calculated by adding  $b_3$  to  $b_1$ , weighted by a value of  $z$ :

$$SS = b_1 + b_3 * z. \quad (2)$$

For example, the simple slope when  $z = 2$  is  $b_1 + 2 \times b_3$ , and the simple slope when  $z = -1.5$  is  $b_1 - 1.5 \times b_3$ . If substantively interesting levels of  $z$  are not available a priori, then representative values can be chosen from the data, such as quartiles or the mean  $\pm 1SD$ .

How the simple effects vary across the moderator can be visualized in a scatterplot with  $x$  and  $y$  on their respective axes, and drawing separate regression lines for how expected values of  $y$  vary across  $x$  at a few chosen levels of  $z$ . In order to obtain tests of simple slopes, their  $SE$ s are required, which can be obtained by refitting the regression model after centering the moderator at each desired level of  $z$ . For example, when  $z$  is mean centered,  $b_1$  will be the simple slope of  $x$  among people with the (sample) average level of  $z$ .

This process is called *probing the interaction*, and can be simplified by using the delta method [13] to derive  $SE$  estimates under different centering as a function of the  $SE$  estimates from a single fitted model. This process is implemented for regression models in the R package *interactions* [14], along with interaction plots.

Equation (1) involves a two-way interaction, which is an upper limit in a model with only two variables that can be multiplied against each other. Quadratic terms (e.g.,  $x^2$ ) are also possible, which can be interpreted as a predictor moderating its own effect ([5] chapter 7). A model with a third predictor ( $w$ ) might still include only a single two-way interaction:

$$y_i = b_0 + b_1x_i + b_2z_i + b_3x_iz_i + b_4w_i + e_i, \quad (3)$$

where  $b_4$  is not a conditional effect of  $w$  but rather a marginal effect (i.e., the same for each subject  $i$ ), given the other effects in the model. The role of a predictor like  $w$  is, therefore, not a moderator of the focal predictor  $x$  but a covariate or statistical control variable.

A three-way interaction involves multiplying three predictors and including that three-way product as a predictor:

$$y_i = b_0 + b_1x_i + b_2z_i + b_3x_iz_i + b_4w_i + b_5x_iz_iw_i + b_6z_iz_iw_i + b_7x_iz_iz_iw_i + e_i, \quad (4)$$

where  $b_7$  represents the three-way interaction effect. Models with higher-order terms must follow the *hierarchical principle* by also including all lower-order component terms. In Equations (1) and (3), the product  $xz$  is only interpreted as a two-way interaction because both  $x$  and  $z$  have simple effects in the model. In Equation (4), the product  $xzw$  is only interpreted as a three-way interaction because all first-order terms ( $x$ ,  $z$ , and  $w$ ) and second-order terms ( $xz$ ,  $xw$ , and  $zw$ ) are included.

A three-way interaction can be interpreted similar to a two-way interaction, by choosing a focal predictor whose effect is of primary interest, and evaluating how that variable's effect on  $y$  is moderated by changes in both  $z$  and  $w$  ([3] chapter 3). A three-way interaction can also be interpreted as the moderation of a two-way interaction effect by a third predictor (e.g., how the nature of an  $x \times z$  interaction changes with each 1-unit increase in  $w$ ).

One additional concern with interaction models may be that interaction effects represent spurious findings driven by other non-linear effects [15,16]. To address this issue, researchers may also fit models including both product terms and quadratic effects of the focal predictor and moderator. However, when fitting a model with latent interactions, these more complex models may result in biased estimates and convergence problems [17].

### 1.2. Interaction among Latent Variables

Estimating interactions between latent variables provides additional challenges beyond multiple regression. Latent variables (e.g., common factors or growth factors) are unobserved; therefore, individual subjects' values (factor scores) are unobserved. It is not possible to multiply latent factor scores to calculate a product term to be included in the model. A latent interaction could also involve the interaction between a latent variable and an observed variable, which can be accomplished by treating the observed predictor as a single-indicator factor.

It is also possible to use the techniques discussed here [18,19]; however, when an observed moderator is categorical, we recommend using a multigroup SEM to evaluate interactions [20]. The inability to easily and accurately multiply latent factor scores has motivated many different approaches to estimating and testing latent interactions.

Some approaches have been developed specifically to estimate latent interactions, such as latent moderated structures (LMS) [8], which does not require the product terms among latent variables to be explicitly calculated. Instead, ML estimates are obtained by maximizing the log-likelihood of the joint distribution of indicators for the focal predictor and outcome, conditional on indicators of the moderator—essentially, a mixture distribution across the moderator indicators [21].

LMS assumes that the conditional distribution is multivariate normal, although it has been adapted for ordinal indicators [22] as well as censored and count variables ([23] chapter 17). LMS is computationally intensive, requiring numerical integration, and has only been implemented in two dedicated SEM software packages [23,24]. New developments in modeling factor scores are an additional method for estimating interactions among latent variables [25,26], especially when sample sizes are smaller. However, these approaches lack user-friendly software implementations and require further investigation to determine best practice methods.

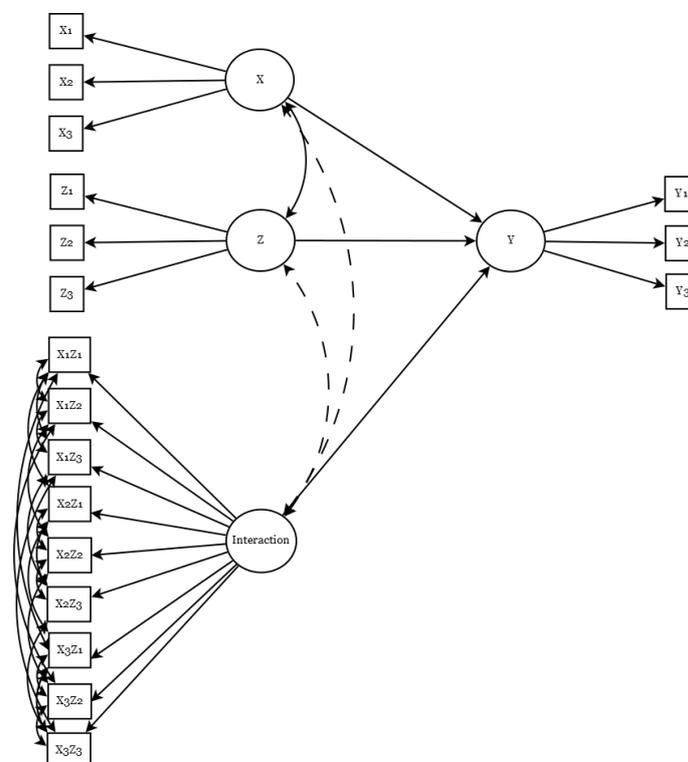
Markov chain Monte Carlo (MCMC) estimation, on the other hand, has made it feasible to incorporate latent product terms directly into the model. In a Bayesian framework, factor scores can be sampled from their posterior distribution at each iteration of the Markov chain, along with all other unknown parameters [27]. Factor score estimates from different latent variables can, then, be multiplied just as they would be if they were observed [28,29], except that the factor scores vary across samples from the posterior, appropriately reflecting their (im)precision given the observed data.

Beginning with version 8.2, *Mplus* provides dedicated syntax—via XWITH ([23] chapter 17)—to simplify this when using MCMC. In contrast, the Bayesian SEM package

blavaan [30] does not provide such functionality; instead, general Bayesian modeling software syntax would have to be written to include interaction terms.

This article focuses on product–indicator approaches [6,7,9,10,31], in which the variance of the latent product is extracted from products of the factors’ indicators, and thereby there is a measurement model for the latent interaction term(s) as well as for the lower-order terms (see Figure 1). We provide details about implementing product–indicator methods in the following section, but here we list some advantages of using product indicators, foremost of which is that they can be applied in any SEM software package using standard estimation procedures.

In contrast, LMS does not yield the standard test statistic or fit indices used to evaluate data–model correspondence, and fit indices have only recently been proposed for MCMC estimation of SEMs [32] and incorporated into *blavaan* and *Mplus* [33]. Product indicators can also accommodate heteroskedasticity in situations where LMS assumes homoskedasticity [34].



**Figure 1.** Path diagrams of unconstrained latent interaction models for both residual and mean centering. Dotted lines represent covariances estimated in mean centering models but fixed to 0 in residual centering models.

There are also some disadvantages, such as product indicators are assumed to be continuous by virtue of multiplying their values, whereas LMS and MCMC are flexible enough to incorporate ordinal indicators. Product indicators for ordinal variables have been found tenable when combined with a parceling strategy [35]. Another disadvantage of product indicators is that the number of observed variables (and thus the size of the observed and model-implied covariance matrices) increases, as well as the number of measurement parameters for the factor representing the latent interaction (which are absent when using LMS or MCMC).

### 1.3. Product-Indicator Approaches

In product–indicator approaches, each indicator of the latent focal predictor (say,  $x$ ) is multiplied with an indicator of the latent moderator ( $z$ ). The products are then used as indicators of a new latent variable representing the interaction between the two latent

predictors. This latent interaction factor can then be used to predict outcome variable(s), just as shown in Equation (1) with the observed variable(s), providing an estimate (and test) of the latent interaction effect.

Early implementations of the product–indicator approach were based on computing products of indicators and imposing complex constraints when estimating the model [6,7]. The constrained approach [6] originally proposed calculating products between all possible pairs of the first-order indicators to measure the latent interaction factor, and measurement parameters (factor loadings and residual variances) for the product indicators were constrained to be functions of the first-order indicators' parameters.

To simplify the approach, a single indicator of the latent interaction factor was later proposed [7], calculated from the first (reference) indicators of the two latent predictors; however, complex constraints were still imposed. Even interactions between only two latent variables could require over 30 constraints, making the constrained approaches difficult to implement and prone to programming errors in practice.

These two constrained approaches were later found not to perform appreciably better than unconstrained estimation [9], which is much simpler because it does not require making product–indicator loadings (or residual variances) a function of the respective first-order indicator loadings (or residual variances). The same study also revealed that a single product indicator was insufficient to capture the latent-interaction variance, but that all pairs were unnecessary; instead, a matched-pairs strategy would utilize each indicator's information only once (assuming both factors have the same number of indicators) [9].

In addition to proposing these two simplifications, Marsh et al. [9] recommended mean-centering first-order indicators before calculating product indicators [9]. In the context of regression with observed variables, mean-centering predictors has been recommended [2] so that the product term is uncorrelated with the lower-order terms, when the predictors are normally distributed. Although the estimated interaction effect is unaffected by centering, the estimated first-order effects are simple slopes holding the moderator constant at its mean [36], rather than constant at zero (on the original scale). Centering indicators (e.g.,  $x_1$  and  $z_1$ ) also ensures that they will be uncorrelated with their corresponding product indicators (i.e.,  $x_1 \times z_1$ ).

Although mean-centering first-order predictors obviates the need to estimate their intercepts, the SEM must still be estimated with a mean structure because product indicators can have nonzero means, especially when indicators are not normally distributed [9]. Residual centering [10] was proposed to simplify a SEM with product indicators by rendering the mean structure unnecessary. Residual centering also originated in regression models with observed variables, and involves regressing a product term on all lower-order indicators (including lower-order product indicators in models with higher-order interactions). Working from Equation (1):

$$\begin{aligned}x_i z_i &= a_0 + a_1 x_i + a_2 z_i + r_i \\r_i &= x_i z_i - (a_0 + a_1 x_i + a_2 z_i),\end{aligned}\tag{5}$$

where the residual  $r_i$  (uncorrelated with lower-order terms) is used in place of the usual product term. Again, the estimated interaction effect is unaffected, but the lower-order effects in this case are marginal effects rather than simple or conditional slopes [37]. Residual centering product indicators also involves regressing the product indicators on the first-order indicators from which they were calculated and, then, using each product indicator's residual as an indicator of the latent interaction.

Double mean centering (DMC) was proposed to combine the advantage of residual centering with the simplicity of mean centering [31]. DMC begins with mean centering; however, after calculating product indicators, the product indicators are additionally mean centered. Compared with residual centering, DMC product indicators always measure the latent product term, whereas residualized product indicators introduce bias whenever indicators have nonzero skew [31] (e.g., when normally distributed).

When indicators are symmetric (e.g., when normally distributed), the latent interaction factor will be uncorrelated with the first-order latent factors when using DMC or residual centering, and thus those covariances can be fixed to zero. However, perfectly symmetric data are rare in practice, and with the availability of nonnormality corrections to *SEs* and test statistics [38], it may be safer to freely estimate the factor covariances with the latent interaction. Doing so can also account for unmodeled heteroskedasticity of factor scores [34].

Finally, when calculating products between all pairs of indicators, residual covariances should be estimated between product indicators that share a first-order indicator (e.g.,  $x_1 \times z_1$  with  $x_2 \times z_1$ ). Otherwise, shared item-specific variance would be conflated with shared latent-interaction variance, potentially biasing its estimated effect. Equality constraints could also be applied, as shown in Table 1.

However, residual covariances are unnecessary when using the matched-pairs strategy because each indicator is only used to calculate one product indicator. Although the simulation study in [9] showed similar performance between matched and all pairs, the estimated interaction effect was found to vary substantially across different matched-pairs strategies in real data [39], calling into question the value of its simplicity.

**Table 1.** The matrix of residual covariances for the latent interaction models. Note, covariances with the same label are constrained to equality.

	x1z1	x1z2	x1z3	x2z1	x2z2	x2z3	x3z1	x3z2	x3z3
x1z1									
x1z2	t1								
x1z3	t1	t1							
x2z1	t4	0	0						
x2z2	0	t5	0	t2					
x2z3	0	0	t6	t2	t2				
x3z1	t4	0	0	t4	0	0			
x3z2	0	t5	0	0	t5	0	t3		
x3z3	0	0	t6	0	0	t6	t3	t3	

An additional consideration when estimating latent interactions is examining the model fit. Currently there are not well developed measures of model fit appropriate for latent interaction variables using LMS. When using a product–indicator method, the model fit will be reported by software for the latent interaction model, but traditional measures of model fit are not appropriate for product–indicator models. In a product–indicator model, there are several variables included within the model that are functions of other variables, and the product indicators are functions of their constituent indicators.

Traditional methods of assessing model fit do not take this dependence among observed variables into account and, thus, provide incorrect estimates of model fit. To assess model fit for latent interaction model we recommend first fitting a model without the latent interaction variable, or any product–indicators, and assessing model fit on this “main effects” model. The latent interaction model does not include any additional variables, only functions of the variables in the “main effects” model, and fit for this “main effects” model can be viewed as a lower bound for fit of the latent interaction model.

## 2. Materials and Methods

In this section, we describe how to employ product–indicator methods in R, using a hypothetical model similar to the model in Figure 1. This model has a focal predictor  $x$  with three indicators ( $x_1, x_2, x_3$ ), a moderator  $z$  with three indicators ( $z_1, z_2, z_3$ ), and an outcome  $y$  with three indicators ( $y_1, y_2, y_3$ ). First, we describe how to use the `indProd()` function in the `semTools` [40] package to automatically create product indicators to estimate a latent interaction effect in `lavaan` [41].

Using the fitted model returned by `lavaan`, the `semTools` package can also be used to probe and plot the interaction. After describing these software utilities, they are applied to a real data set to demonstrate how to interpret the results. The data and syntax can be retrieved from our Open Science Framework (OSF) project <https://osf.io/npq24>, accessed on 9 July 2021.

### 2.1. Estimating Latent Interactions in *Lavaan*

Before fitting a model with a latent interaction variable, product indicators must be created using either a DMC or residual centering strategy. This process is automated using the `indProd()` function in the `semTools` package. The `indProd()` function takes a data frame as input and returns a new data frame with product indicators added as new variables for either two- or three-way interactions, the product indicators can be constructed using mean centering, DMC, or residual centering. The first argument to the `indProd()` function is `data` argument, which is a data frame that includes the indicators of the focal predictor and moderator, along with other variables to be used in the model.

In our example, the data frame would include the variables `x1`, `x2`, `x3`, `z1`, `z2`, `z3`, `y1`, `y2`, and `y3`. The next arguments are `var1`, `var2`, and `var3`. The `var1` argument should be a vector of the names of the indicators of the focal predictor, the `var2` argument should be a vector of the names of the indicators of the moderator, and, in the case of a three-way interaction, the `var3` argument should be a vector of the names of the indicators of the second moderator.

The `match` argument is a logical argument; when `match` is `TRUE`, a matched pair strategy is used, constructing product indicators based on the order variables are specified in the `var1`, and `var2` arguments. When `match` is `FALSE`, the all pairs strategy is used. The next three arguments (`meanC`, `residualC`, and `doubleMC`) dictate how product indicators are centered. If `meanC` = `TRUE`, mean centering is used, if `residualC` = `TRUE`, residual centering is used, and if `doubleMC` = `TRUE`, DMC is used.

Only one of `meanC`, `residualC`, `doubleMC` should be specified as `TRUE` each time the `indProd()` function is used. We do not recommend using mean centering `meanC` = `TRUE` as DMC and residual centering perform equivalently and require fewer constraints in the model. The final argument is `namesProd`, and this optional argument allows users to specify the names of the new product terms as a character vector with the same length as the number of product terms.

If this argument is not specified, the default names for product terms are the name of the indicator for the focal predictor and the name of the indicator for the moderator separated by a period. In our example, the default name for the product indicator of `x1` and `z1` would be `x1.z1`. In our example, assuming the data are in a data frame called `dat`, computing product using DMC with an all pairs approach the code would be:

```
dat2 <- indProd(dat, var1 = c("x1", "x2", "x3"),
               var2 = c("z1", "z2", "z3"),
               match = FALSE, meanC = FALSE,
               residualC = FALSE, doubleMC = TRUE)
```

The data frame `dat2` will include all the variables in `dat` with nine new variables representing the product terms for the latent interaction variable. To test the interaction between `x` and `z`, a model similar to Figure 1 would be fit in `lavaan`. The model syntax would be:

```
mod <- "
## Factor loadings
x =~ x1 + x2 + x3
z =~ z1 + z2 + z3
y =~ y1 + y2 + y3
xz =~ x1.z1 + x1.z2 + x1.z3 +
      x2.z1 + x2.z2 + x2.z3 +
      x3.z1 + x3.z2 + x3.z3
## Regression
y ~ x + z + xz
## Residual covariances
## constrained to equality between the same item
x1.z1 ~~ t1*x1.z2 + t1*x1.z3
```

```

x1.z2 ~~ t1*x1.z3
x2.z1 ~~ t2*x2.z2 + t2*x2.z3
x2.z2 ~~ t2*x3.z3
x3.z1 ~~ t3*x3.z2 + t3*x3.z3
x3.z2 ~~ t3*x3.z3
x1.z1 ~~ t4*x2.z1 + t4*x3.z1
x2.z1 ~~ t4*x3.z1
x1.z2 ~~ t5*x2.z2 + t5*x3.z3
x2.z2 ~~ t5*x3.z2
x1.z3 ~~ t6*x2.z3 + t6*x3.z3
x2.z3 ~~ t6*x3.z3
"
fit <- sem(mod, data = dat2, std.lv = TRUE,
          meanstructure = TRUE)

```

As the all pairs approach was used, residual covariances are included among all items following the pattern indicated in Table 1. In the `sem` function, the `std.lv` argument is used to set the scale by fixing the factor variances to one, and a mean structure is estimated using the `meanstructure` argument. These arguments are not required but using them makes plotting and probing interactions easier.

## 2.2. Probing and Plotting Latent Interactions

In SEM with latent interactions, the first-order factor scores cannot be centered for the same reason they cannot be used to calculate a product term to estimate the interaction effect. However, latent variables have no inherent location or scale (mean or variance), and therefore they are set using arbitrary identification rules (e.g., by fixing them to 0 and 1, respectively, so that factor scores are assumed to be  $z$  scores).

In theory, the model could be refit using different fixed values to identify the mean of the moderating factor; however, this would require a mean structure to be estimated, which was a complication meant to be avoided by using DMC and residual centering. Instead, simple slopes can be calculated as shown above (by adding a weighted  $b_3$  to  $b_1$ ), and the delta method can be used to derive the appropriate  $SE$ .

This method cannot be used with residual-centered product indicators because the intercept and slopes for lower-order terms are what they would have been in a model without an interaction term; however, their  $SE$ s differ. This is why [37] argued against the use of residual centering in the context of OLS regression. When a latent interaction has residual-centered product indicators, probing the interaction requires knowing the sufficient statistics (means and covariance matrix) among noncentered latent predictors, as well as what the point and  $SE$  estimates of their effects on the outcome in a model when they are not centered. This can be accomplished by fitting such an additional model and calculating the appropriate transformations, but the required values can also be derived analytically from the fitted model [42].

This process is automated in the R package `semTools` [40], using one of four dedicated functions that is designed to probe latent interactions estimated with `lavaan`. The function `probe2WayMC()` assumes product indicators for a two-way latent interaction were formed using mean-centering or DMC. The function `probe3WayMC()` is analogously designed for probing a three-way latent interaction. The functions `probe2WayRC()` and `probe3WayRC()` are analogous functions for latent interactions formed using residual centering.

All functions accept a fitted `lavaan` model, and users are required to provide names for the factors that are the outcome (the `nameY=` argument) and predictors, including the interaction latent variable, which should be listed last (the `nameX=` argument), specifically designating one predictor as the moderator (the `modVar=` argument). The `valProbe=` argument is used to indicate the values of the moderator at which to probe the focal predictor's effect. In multigroup models, the `group=` argument is used to select in which group to probe the interaction.

The functions provide estimates, standard errors ( $SE$ s), and statistical significance tests of simple slopes at the values specified in the `valProbe=` argument. Optionally, simple intercepts can be requested. The output of the probing functions can be plotted using `plotProbe()` function. Users are required to specify the output from probing the interaction

(from the `probe2WayMC()`, `probe3WayMC()`, `probe2WayRC()`, `probe3WayRC()` functions) and the range of values to plot on the  $x$  axis (the `xlim=` argument). Plots can be customized using additional arguments to `plotProbe()` and additional graphical parameters for base graphics in R.

### 2.3. Real-Data Application

Data from this example are a subset of the data reported in [43]. For more extensive details on measures and methods see [43]. Von der Embse and colleagues were interested in how teacher stress influenced counterproductive teaching practices in the context of high stakes testing, and how this relationship is moderated by teachers' perceived value of high stakes testing. After listwise-deleting 485 participants (6.6%), the sample included 6796 teachers from the Southeastern United States; for demographic information see [43]. Teacher stress, the focal predictor, was measured using five items from the Educator Test Stress Inventory [44], which assesses sources of teacher stress related to testing (example item: "I feel pressure from administrators to raise student test scores").

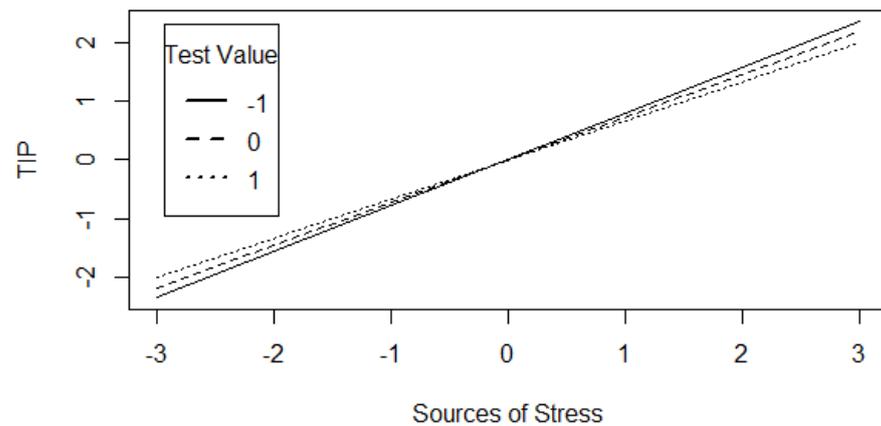
The perceived test value, the moderator, was measured using four items developed by von der Embse and colleagues (example item "Student test performance represents a valid assessment of school effectiveness"). Teacher instructional practices (TIP), the outcome, was measured using three items developed by von der Embse and colleagues (example item "I spend most of my time teaching to a standardized test."). All items were measured using a 5-point Likert-type scale. Here, we provide an example using DMC with an all pairs approach, example syntax using residual centering is available with Supplementary Materials. All models used a fixed factor method of scale setting, using a maximum likelihood estimator.

We first fit the "main effects" model, without an interaction latent variable to assess model fit and the effect of test value and sources of stress on TIP. We, then, fit a second model, which included the interaction latent variable to test the interaction of the test value and sources of stress. The interaction was plotted and probed using a simple slopes approach with simple slopes estimated at the 1  $SD$  below the mean of test value ( $-1$ ), the mean of test value ( $0$ ), and 1  $SD$  above the mean of test value ( $1$ ). All analyses used `lavaan` version 0.6–8 and `semTools` version 0.5–4.

### 3. Results

The main effects model fit reasonably well,  $\chi^2(51) = 1262.00$ ,  $RMSEA = 0.058$ , 90% CI = 0.055–0.062, CFI = 0.957, TLI = 0.944, and SRMR = 0.039, with the test value negatively related to TIP,  $b = -0.12$ ,  $SE = 0.02$ , and  $p < 0.001$  and sources of stress positively related to TIP,  $b = 0.72$ ,  $SE = 0.03$ , and  $p < 0.001$ . Teachers who valued standardized testing more reported less instruction focused on testing, and teachers who felt more stress about standardized testing reported more instruction focused on testing.

There was an interaction between the test value and sources of stress,  $b = -0.07$ ,  $SE = 0.02$ ,  $p = 0.001$ . Probing of the interaction showed that across values of the test value, the simple slope of sources of stress was positive, but this relationship was weaker as the test value increased, Table 2. This relationship can be visualized with a plot of simple slopes produced by the `plotProbe()` function, see Figure 2 for an example.



**Figure 2.** Simple slopes for sources of stress predicting teacher instructional practices at different values of the test value.

**Table 2.** Simple slopes for sources of stress predicting teacher instructional practices.

Test Value	Slope	SE	<i>p</i>
-1	0.78	0.03	<0.001
0	0.72	0.03	<0.001
1	0.66	0.03	<0.001

#### 4. Discussion

Testing for interactions using latent variable models provides additional power to detect effects when compared to regression models [11]. Testing the same interaction in the real-data application using regression models with scale scores as the variables, resulted in the conclusion that there was no interaction between the test value and sources of stress,  $b = -0.01$ ,  $SE = 0.01$ , and  $p = 0.056$  when predicting TIP. Product indicator methods provide an easy and accurate method of testing interactions among latent variables.

The process of estimating and interpreting latent variable interactions using a product indicator method is easily implemented using the `semTools` and `lavaan` packages. Product indicators were created using the `indProd` function in `semTools`, the model was fit with `lavaan`, and interactions were probed using the `probe2WayMC()`, `probe3WayMC()`, `probe2WayRC()`, and `probe3WayRC()` functions and plotted using the `plotProbe()` function.

##### 4.1. Extensions

We focused on a simple model with a single focal predictor, a single moderator, and a single outcome variable. However, the techniques we discussed can be extended to more complex models using additional moderators or additional outcome variables. One extension is the inclusion of an additional moderator variable, resulting in a three-way interaction. In a three-way interaction, the interaction between two predictors differs across the levels of a third variable. In regression models, three-way interactions are tested using the product of all three variables as a predictor, along with all conditional two-way products and conditional main effects as well (c.f. [2]).

Three-way interactions are plotted and probed by investigating the simple slope of one predictor at conditional values of both moderating variables [4,45]. When estimating three-way interactions among latent variables using a product–indicator method, product indicators must be formed for all two-way interactions among latent variables, as well as the three-way interaction among latent variables.

In a preliminary simulation study, residual centering performed better than DMC [46]; however, additional research is needed on the best practices for methods estimating three-way interactions among latent variables. The `indProd()` function includes functionality for creating product terms for three-way interactions using either residual centering or DMC.

The `probe3WayRC()` and `probe3WayMC()` functions provide simple slopes for three-way interactions.

Interactions may easily be included as part of a larger model, with additional predictors and outcomes in the model. One type of popular model that includes interactions in a larger model is moderated mediation models. In the simplest form of these models, the relationship between a predictor ( $x$ ) and an outcome ( $y$ ) is partially explained by a third variable ( $m$ ). Regression paths from  $x$  to  $y$ ,  $x$  to  $m$ , and  $m$  to  $y$  are estimated. An indirect effect is computed to estimate the effect of  $x$  predicting  $y$  through  $m$ , which is the product of the regression paths from  $x$  to  $m$  and  $m$  to  $y$ .

In a moderated mediation model, the paths from  $x$  to  $m$  or  $m$  to  $y$  may be moderated [4]. The indirect effect may now vary across values of the moderator variable, and simple indirect effects are estimated for different levels of the moderator. These models can be extended to latent variable interactions using product indicator methods. Simple slopes computed using the `probe2WayRC()` or `probe2WayMC()` functions may be multiplied with other slopes to compute simple indirect effects.

The simple indirect effects can be tested using bootstrapping, via the `bootstrapLavaan()` function or with Monte Carlo confidence intervals. We included an example of one of these models in the Supplementary Materials. One additional consideration for some moderated mediation models, or other models where moderators are endogenous, is to use appropriate values of the moderator when computing simple slopes [47].

The current example used ML estimation for all models. Given the inherently non-normal distributions of product indicators, robust corrections for ML are recommended when fitting models with product indicators. This estimation method can be easily implemented in `lavaan` by adding `estimator = 'mlr'` to the `sem` command. However, when probing interactions, the `probe2WayMC()`, `probe3WayMC()`, `probe2WayRC()`, and `probe3WayRC()` functions require ML, not robust ML, as the standard errors for simple slopes are currently only derived under ML. Future research should investigate methods for computing robust standard errors for simple slopes.

We focused on a simple slopes (sometimes called “pick-a-point”) approach to probing interactions. Interactions may also be probed using a Johnson–Neyman technique [45]. In the Johnson–Neyman technique, a range of values of the moderator are identified, which include values where the simple slope is significantly different from zero. This has the advantage of considering a much wider range of values for the moderator when compared to the simple slopes approach.

Currently, the Johnson–Neyman technique is not implemented in the `semTools` probing functions. With latent interactions, the Johnson–Neyman technique is of limited value. The scale of latent variables is arbitrary, and determining the specific values of the latent moderator where the simple slope is different from zero is not as useful as identifying how simple slopes change across values of the moderator, which can be accomplished using a simple slopes approach.

#### 4.2. Conclusions

Examining interactions among predictors is an important part of a developing research program. When predictors are modeled as latent variables, estimating and testing interactions requires additional steps beyond the models used for regression. Product indicator methods of examining latent interactions provide an accurate method to estimate and test latent interactions. Significant latent interactions require additional steps (plotting and probing) to interpret interaction effects. These methods can be easily implemented using functions in the `semTools` package with models fit using the `lavaan` package in R.

**Supplementary Materials:** The following are available online at <https://osf.io/npq24>.

**Author Contributions:** Conceptualization, A.M.S. and T.D.J.; methodology, A.M.S. and T.D.J.; software, T.D.J.; data curation and formal analysis, A.M.S.; writing—original draft preparation, A.M.S. and T.D.J.; writing—review and editing, A.M.S. and T.D.J.; visualization, A.M.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Dutch Research Council (NWO), project number 016.Veni.195.457, awarded to Terrence D. Jorgensen.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data and syntax for the example are available at Supplementary Materials.

**Acknowledgments:** We would like to acknowledge Sunthud Pornprasernmanit for his work as the original maintainer of the `semTools` package and his work on writing and editing original code for examining latent interactions.

**Conflicts of Interest:** The authors declare no conflict of interest. The funders had no role in the design of the study, nor in the writing of the manuscript.

### Abbreviations

The following abbreviations are used in this manuscript:

DMC	Double mean centering
LMS	Latent moderated structural equations
ML(E)	Maximum likelihood (estimation)
SE	Standard error
SEM	Structural equation model(ing)

### References

1. Aiken, L.S.; West, S.G.; Reno, R.R. *Multiple Regression: Testing and Interpreting Interactions*; Sage: Thousand Oaks, CA, USA, 1991.
2. Cohen, J.; Cohen, P.; West, S.G.; Aiken, L.S. *Applied Multiple Regression/Correlation Analysis for the Behavioral Sciences*; Routledge: Mahwah, NJ, USA, 2003. [[CrossRef](#)]
3. Jaccard, J.; Turrisi, R. *Interaction Effects in Multiple Regression*; Sage: Mahwah, NJ, USA, 2003; Number 27.
4. Hayes, A.F. *Introduction to Mediation, Moderation, and Conditional Process Analysis: A Regression-Based Approach*; Guilford: New York, NY, USA, 2017.
5. Judd, C.M.; McClelland, G.H.; Ryan, C.S. *Data Analysis: A Model Comparison Approach to Regression, ANOVA, and Beyond*, 3rd ed.; Routledge: Mahwah, NJ, USA, 2017.
6. Kenny, D.A.; Judd, C.M. Estimating the nonlinear and interactive effects of latent variables. *Psychol. Bull.* **1984**, *96*, 201–210. [[CrossRef](#)]
7. Jöreskog, K.G.; Yang, F. Nonlinear structural equation models: The Kenny–Judd model with interaction effects. In *Advanced Structural Equation Modeling: Issues and Techniques*; Marcoulides, G.A., Schumacker, R.E., Eds.; Psychology Press: New York, NY, USA, 1996; pp. 57–88. [[CrossRef](#)]
8. Klein, A.; Moosbrugger, H. Maximum likelihood estimation of latent interaction effects with the LMS method. *Psychometrika* **2000**, *65*, 457–474. [[CrossRef](#)]
9. Marsh, H.W.; Wen, Z.; Hau, K.T. Structural equation models of latent interactions: Evaluation of alternative estimation strategies and indicator construction. *Psychol. Methods* **2004**, *9*, 275–300. [[CrossRef](#)] [[PubMed](#)]
10. Little, T.D.; Bovaird, J.A.; Widaman, K.F. On the merits of orthogonalizing powered and product terms: Implications for modeling interactions among latent variables. *Struct. Equ. Model.* **2006**, *13*, 497–519. [[CrossRef](#)]
11. Lodder, P.; Denollet, J.; Emons, W.H.; Nefs, G.; Pouwer, F.; Speight, J.; Wicherts, J.M. Modeling interactions between latent variables in research on Type D personality: A Monte Carlo simulation and clinical study of depression and anxiety. *Multivar. Behav. Res.* **2019**, *54*, 637–665. [[CrossRef](#)] [[PubMed](#)]
12. R Core Team. *R: A language and Environment for Statistical Computing*; Version 3.6.1; R Foundation for Statistical Computing: Vienna, Austria, 2019.
13. Oehlert, G.W. A note on the delta method. *Am. Stat.* **1992**, *46*, 27–29. [[CrossRef](#)]
14. Long, J.A. *Interactions: Comprehensive, User-Friendly Toolkit for Probing Interactions*; R package version 1.1.0; R Foundation for Statistical Computing: Vienna, Austria, 2019.
15. Ganzach, Y. Misleading interaction and curvilinear terms. *Psychol. Methods* **1997**, *2*, 235. [[CrossRef](#)]

16. Lubinski, D.; Humphreys, L.G. Assessing spurious “moderator effects”: Illustrated substantively with the hypothesized (“synergistic”) relation between spatial and mathematical ability. *Psychol. Bull.* **1990**, *107*, 385. [[CrossRef](#)]
17. Harring, J.R.; Weiss, B.A.; Li, M. Assessing spurious interaction effects in structural equation modeling: A cautionary note. *Educ. Psychol. Meas.* **2015**, *75*, 721–738. [[CrossRef](#)] [[PubMed](#)]
18. Kolbe, L.; Jorgensen, T.D. Using product indicators in restricted factor analysis models to detect nonuniform measurement bias. In *Quantitative Psychology, Proceedings of the 82nd Annual Meeting of the Psychometric Society, New York, NY, USA, 10–13 July 2017*; Wiberg, M., Culppepper, S.A., Janssen, R., González, J., Molenaar, D., Eds.; Springer: Zürich, Switzerland, 2018; pp. 235–245. [[CrossRef](#)]
19. Woods, C.M.; Grimm, K.J. Testing for nonuniform differential item functioning with multiple indicator multiple cause models. *Appl. Psychol. Meas.* **2011**, *35*, 339–361. [[CrossRef](#)]
20. Little, T.D.; Card, N.A.; Slegers, D.W.; Ledford, E.C. Representing contextual effects in multiple-group MACS models. In *Modeling Contextual Effects in Longitudinal Studies*; Little, T.D., Bovaird, J.A., Card, N.A., Eds.; Routledge: New York, NY, USA, 2007; pp. 121–147.
21. Cham, H.; Reshetnyak, E.; Rosenfeld, B.; Breitbart, W. Full information maximum likelihood estimation for latent variable interactions with incomplete indicators. *Multivar. Behav. Res.* **2017**, *52*, 12–30. [[CrossRef](#)]
22. Aytürk, E.; Cham, H.; Jennings, P.A.; Brown, J.L. Exploring the performance of latent moderated structural equations approach for ordered-categorical items. *Struct. Equ. Model.* **2020**, *28*, 410–422. [[CrossRef](#)]
23. Muthén, L.K.; Muthén, B.O. *Mplus User’s Guide*, 8th ed.; Mplus: Los Angeles, CA, USA, 1998–2017.
24. Umbach, N.; Naumann, K.; Brandt, H.; Kelava, A. Fitting nonlinear structural equation models in R with package `n1sem`. *J. Stat. Softw.* **2017**, *77*, 1–20. [[CrossRef](#)]
25. Ng, J.C.; Chan, W. Latent moderation analysis: a factor score approach. *Struct. Equ. Model. Multidiscip. J.* **2020**, *27*, 629–648. [[CrossRef](#)]
26. Cox, K.; Kelcey, B. Croon’s bias-corrected estimation of latent interactions. *Struct. Equ. Model.* **2021**. [[CrossRef](#)]
27. Scheines, R.; Hoijtink, H.; Boomsma, A. Bayesian estimation and testing of structural equation models. *Psychometrika* **1999**, *64*, 37–52. [[CrossRef](#)]
28. Lee, S.Y.; Song, X.Y.; Poon, W.Y. Comparison of approaches in estimating interaction and quadratic effects of latent variables. *Multivar. Behav. Res.* **2004**, *39*, 37–67. [[CrossRef](#)]
29. Lee, S.Y.; Song, X.Y.; Tang, N.S. Bayesian methods for analyzing structural equation models with covariates, interaction, and quadratic latent variables. *Struct. Equ. Model.* **2007**, *14*, 404–434. [[CrossRef](#)]
30. Merkle, E.C.; Rosseel, Y. `blavaan`: Bayesian structural equation models via parameter expansion. *J. Stat. Softw.* **2018**, *85*, 1–30. [[CrossRef](#)]
31. Lin, G.C.; Wen, Z.; Marsh, H.W.; Lin, H.S. Structural equation models of latent interactions: Clarification of orthogonalizing and double-mean-centering strategies. *Struct. Equ. Model.* **2010**, *17*, 374–391. [[CrossRef](#)]
32. Garnier-Villarre, M.; Jorgensen, T.D. Adapting fit indices for Bayesian structural equation modeling: Comparison to maximum likelihood. *Psychol. Methods* **2020**, *25*, 46–70. [[CrossRef](#)] [[PubMed](#)]
33. Asparouhov, T.; Muthén, B. Advances in Bayesian model fit evaluation for structural equation models. *Struct. Equ. Model.* **2020**, *28*, 1–14. [[CrossRef](#)]
34. Kolbe, L.; Jorgensen, T.D.; Molenaar, D. The impact of unmodeled heteroskedasticity on assessing measurement invariance in single-group models. *Struct. Equ. Model.* **2020**, *28*, 82–98. [[CrossRef](#)]
35. Aytürk, E.; Cham, H.; Jennings, P.A.; Brown, J.L. Latent variable interactions with ordered-categorical indicators: Comparisons of unconstrained product indicator and latent moderated structural equations approaches. *Educ. Psychol. Meas.* **2020**, *80*, 262–292. [[CrossRef](#)] [[PubMed](#)]
36. Echambadi, R.; Hess, J.D. Mean-centering does not alleviate collinearity problems in moderated multiple regression models. *Mark. Sci.* **2007**, *26*, 438–445. [[CrossRef](#)]
37. Echambadi, R.; Arroniz, I.; Reinartz, W.; Lee, J. Empirical generalizations from brand extension research: How sure are we? *Int. J. Res. Mark.* **2006**, *23*, 253–261. [[CrossRef](#)]
38. Savalei, V. Understanding robust corrections in structural equation modeling. *Struct. Equ. Model.* **2014**, *21*, 149–160. [[CrossRef](#)]
39. Foldnes, N.; Hagtvet, K.A. The choice of product indicators in latent variable interaction models: Post hoc analyses. *Psychol. Methods* **2014**, *19*, 444–457. [[CrossRef](#)]
40. Jorgensen, T.D.; Pornprasertmanit, S.; Schoemann, A.M.; Rosseel, Y. *semTools: Useful Tools for Structural Equation Modeling*; R package version 0.5-4; R Foundation for Statistical Computing: Vienna, Austria, 2021.
41. Rosseel, Y. `lavaan`: An R package for structural equation modeling and more. *J. Stat. Softw.* **2012**, *48*, 1–36. [[CrossRef](#)]
42. Pornprasertmanit, S.; Schoemann, A.M.; Geldhof, G.J.; Little, T.D. Probing latent interactions estimated with a residual centering approach. *PsyArXiv* **2021**. [[CrossRef](#)]
43. Von der Embse, N.P.; Schoemann, A.M.; Kilgus, S.P.; Wicoff, M.; Bowler, M. The influence of test-based accountability policies on teacher stress and instructional practices: A moderated mediation model. *Educ. Psychol.* **2017**, *37*, 312–331. [[CrossRef](#)]
44. von der Embse, N.P.; Kilgus, S.P.; Solomon, H.J.; Bowler, M.; Curtiss, C. Initial development and factor structure of the Educator Test Stress Inventory. *J. Psychoeduc. Assess.* **2015**, *33*, 223–237. [[CrossRef](#)]

- 
45. Preacher, K.J.; Curran, P.J.; Bauer, D.J. Computational tools for probing interactions in multiple linear regression, multilevel modeling, and latent curve analysis. *J. Educ. Behav. Stat.* **2006**, *31*, 437–448. [[CrossRef](#)]
  46. Geldhof, G.J.; Pornprasertmanit, S.; Schoemann, A.M.; Little, T.D. Orthogonalizing through residual centering: Extended applications and caveats. *Educ. Psychol. Meas.* **2013**, *73*, 27–46. [[CrossRef](#)]
  47. Frisby, M.; Diemer, M.A. Everything in moderation: A proposed improvement to variance calculation for visualizing latent endogenous moderation. *Struct. Equ. Model.* **2021**. [[CrossRef](#)]