Local Fit Evaluation of Structural Equation Models Using Graphical Criteria

Felix Thoemmes
Cornell University

Yves Rosseel
Ghent University

Johannes Textor
Radboud University Medical Center

Evaluation of model fit is critically important for every structural equation model and sophisticated methods have been developed for this task. Among them are the χ^2 goodness-of-fit test, decomposition of the χ^2, derived measures like the popular RMSEA or CFI, or inspection of residuals or modification indices. Many of these methods provide a *global* approach to model fit evaluation: A single index is computed that quantifies the fit of the entire SEM to the data. In contrast, graphical criteria like *d*-separation or trek-separation allow to derive implications that can be used for *local* fit evaluation, an approach that is hardly ever applied. We provide an overview of local fit evaluation from the viewpoint of SEM practitioners. In the presence of model misfit, local fit evaluation can potentially help in pinpointing where the problem with the model lies. For models that do fit the data, local tests can identify the parts of the model that are corroborated by the data. Local tests can also be conducted before a model is fitted at all, and they can be used even for models that are globally under-identified. We discuss appropriate statistical local tests, and provide applied examples. We also present novel software in R that automates this type of local fit evaluation.
Keywords: Structural equation modeling, fit evaluation

Introduction

Evaluation of model fit[1] is an integral part of any research project that involves structural equation models (SEMs). Researchers who formulate SEMs are interested in whether the proposed model has adequate fit to the actually observed data and they often spend a large amount of time on this testing process. There are a variety of fit measures that are routinely reported, including the global χ^2 statistic of the model, decompositions of the χ^2, and various derived fit indices. In addition, researchers often examine standardized residuals between the observed and model-implied covariance matrix, and may also consult modification indices, and expected parameter change values. The χ^2 statistics are often reported with a significance test, whereas the derived fit measures are typically evaluated based on cut-off values, and are more interpreted like effect sizes.

Researchers almost always report the global χ^2 statistic and its associated *p*-value (Jöreskog, 1969). In addition, the χ^2 is often decomposed in portions that are attributable to the measurement model, or the structural part of the model. This decomposition is sometimes referred to as a two-step procedure in which first the fit of the measurement model is evaluated, and only after that, the fit of the full model that includes both measurement and structural portions (Anderson & Gerbing, 1988). Going even further in this decomposition, James, Mulaik, and Brett (1982) suggested that portions of the model that posit the existence or the absence of an effect should be tested separately. Such a decomposition, along with derivation of fit indices, is given in Lance, Beck, Fan, and Carter (2016).

The global χ^2 is not universally endorsed. An often levied criticism is that the test yields rejection of the model if sample sizes increase, even in the presence of small misspecification. Additional fit measures are often reported instead. A popular one is the root mean square error of approximation (RMSEA) (Steiger, 1990), which is derived from the model χ^2, the degrees of freedom, and the sample size. Another one is the comparative fit index (CFI) (Bentler, 1990), which evaluates the relative distance between a null model (in which all variables are assumed to be independent) and the actual model. Both the RMSEA and the CFI rely on cut-off values that are based on approximate rules as to what constitutes adequate fit. The RMSEA is usually also supplemented with a confidence interval, and a test of close fit

 This article is in press. The published version will be available at http://doi.org/10.1037/met0000147. The R code accompanying this article is available at https://github.com/jtextor/localTesting and in the Appendix.

(Steiger, 2004), which is a significance test of the observed RMSEA against a minimum threshold, usually .05.

A form of local fit assessment are so-called modification indices or expected parameter change values. Modification indices are single-degree of freedom χ^2 tests that show what would happen to the overall global χ^2 if an additional arrow would be added to the model. Expected parameter change values quantify the magnitude of potentially added paths. This data-driven approach to model modification has occasionally been criticized as being too a-theoretical, capitalizing on chance, and leading to models that often cannot be replicated with new samples (MacCallum, 1986; MacCallum, Roznowski, & Necowitz, 1992)[2].

The reliance on χ^2-based measures is so prevalent that alternatives are hardly ever considered. However, it is possible (and we argue fruitful) to perform local testing beyond the modification index and expected parameter change. In our experience, a majority of applied researchers using SEM are unaware that such tests even exist and local tests are currently not featured in any of the leading SEM programs. The goal of the present paper is the following: we will present two graphical criteria, *d*-separation and trek-separation, that yield two local tests, conditional independence and tetrad tests. We will explain how both graphical criteria can be applied to enumerate local tests, and how the derived statistical tests can be performed on data. In particular, we will show that such tests can be performed either by classic significance testing, or by interval estimation of the associated effect sizes. We then will present a series of examples that explain how local fit testing can lead to insights about model misspecification. Some of these local tests are already implemented in existing software (Hipp, Bauer, & Bollen, 2005; Scheines, Spirtes, Glymour, Meek, & Richardson, 1998; Textor, Hardt, & Knüppel, 2011; Bauldry & Bollen, 2016), but we will also present a novel software package, "dagitty" (Textor, van der Zander, Gilthorpe, Liśkiewicz, & Ellison, 2017), written in R, that automates all of these tasks, and can be used in conjunction with already existing SEM software, notably lavaan (Rosseel, 2012). Importantly, our paper does not aim to exhaustively compare the differences between existing fit measures and the local tests we propose. Instead, we simply provide a tutorial on what kind of local tests exist and how they can be applied. We occasionally draw comparisons to other types of model testing strategies, but we do not claim that local tests will be superior to all other types of testing in all situations.

We should also point out that local tests have actually a long history in SEM research. In a classic textbook by Saris and Stronkhorst (1984), the authors briefly discussed some of the methods that we cover below. They ultimately dismissed local fit evaluation as obsolete, and concluded that local fit evaluation has the primary disadvantage of potentially yielding contradicting results. Some tests could suggest support for the model, other tests could indicate rejection of the model. This was seen as problematic, as one would have to make a decision about the model as a whole. A reasonable counter-argument is that this is an advantage of local fit evaluation over global testing. Local tests should be able to tell which parts of a model are supported by the data, and which ones are not, and we should seek out this information. A further concern by Saris and Stronkhorst (1984) was efficiency. This has been partly addressed by the rapid advance in computing power over the past 3 decades, and importantly, efficient graphical criteria are now available that allow us to rapidly derive local tests from the graphical model structure.

Our work also draws on several earlier publications (Pearl, 1988, 2000; Shipley, 2000; Bollen & Ting, 1993, 2000) that discussed various approaches to local fit evaluation. Our own contribution is to introduce readers to local fit evaluation, by providing them a comprehensive review of existing tests based on graphical criteria, by explaining how our novel software can automate the process of local fit evaluation, and by introducing some novel local fit evaluation ideas, including the development of local equivalents for global fit indices like RMSEA or CFI.

Types of local tests

We will describe two types of local tests that are based on graphical criteria. These local tests can be derived before any data are observed, as they rely purely on the structure of the graphical model. The statistical tests that are associated with these graphical criteria can be performed once data are available.

Before describing the two criteria and tests, we first define the following terms: a graphical model consists of *variables* and *paths*. A path can be a one- or two-headed (bidirected) arrow between two variables. A one-headed arrow denotes a direct causal relationship between two variables. A double-headed arrow denotes an influence between two variables that is caused by an unobserved latent variable. We refer to models without latent variables as *path models*. We further define a *route* as any sequence of paths that can be obtained by moving through the model along paths (where moving against the direction of the arrow is allowed). Paths can occur multiple times in a given route.[3] With these definitions in hand, we can now turn to the two graphical criteria.

[2] New approaches that try to mitigate some of the problems with modification indices and give better guidelines on the use of modification indices have been proposed (Saris, Satorra, & Van der Veld, 2009), but have not been tested exhaustively and are not adopted widely.

[3] This also implies that routes can go in circles and can be of infinite length. However, it suffices to examine only those routes where every path is traversed at most once per direction.

d-separation

The *d*-separation criterion was first developed by Pearl (1995). Introductions to *d*-separation in the social sciences are provided by Hayduk and Glaser (2000), and in the context of missing data by Thoemmes and Rose (2014) and Thoemmes and Mohan (2015). We will now review the *d*-separation criterion using the concepts of routes, as defined previously.

The *d*-separation criterion informs the researcher what kind of conditional independencies are encoded in a graphical model. Importantly, these conditional independencies will hold under all parameterizations of the model. That means that if *d*-separation implies a particular conditional independence, then this conditional independence must hold regardless of the functional form a particular arrow will take on. For example, certain arrows between two variables in a graphical model could signify linear relationships, or complex non-linear relationships, and in both instances a conditional independence would be implied if the *d*-separation criterion holds.

For didactic reasons we will first explain *d*-separation using four special cases, namely, four trivially small models with three variables and two paths. These four models form the building blocks for the more general definition. All of these models use the three variables X, M, and Y and contain a path between X and M as well as another path between M and Y. In the graphical models literature, these models are typically named *chain* ($X \rightarrow M \rightarrow Y$)[4], *inverse chain* ($X \leftarrow M \leftarrow Y$), *fork* ($X \leftarrow M \rightarrow Y$), and *collider* ($X \rightarrow M \leftarrow Y$), as shown in Table 1. Each of these models can be expressed as a small set of regression equations. The collider corresponds to a single equation $M \sim X + Y$ whereas the other three models each correspond to two different equations. Each of these models implies a certain (conditional) independence statement.

Assume that we have four datasets that were generated by the models in Table 1. In the collider case, X and Y must be independent variables, otherwise their error terms would be correlated. Therefore, the collider model has the testable implication $\mathrm{Cov}(X, Y) = 0$. The collider model is the *only* model out of the four models in which this unconditional independence holds. In the other three cases, X and Y are in general unconditionally dependent (e.g., correlated), which is not a testable implication since the dependence can be arbitrarily weak. This illustrates the first type of testable implication that we can derive from a graphical model, namely, vanishing covariances.

The second type of implication we can derive is a vanishing *conditional* (or *partial*) covariance. Consider the fork model, which corresponds to the following two equations:

$$X = \beta_1 M + \epsilon_1 \quad (1)$$

$$Y = \beta_2 M + \epsilon_2 \quad (2)$$

Based on these equations, $\mathrm{Cov}(X, Y)$ can be written as $\mathrm{Cov}(\beta_1 M + \epsilon_1, \beta_2 M + \epsilon_2)$. However, if we now hold M constant, then this expression reduces to

$$\mathrm{Cov}(X, Y) = \mathrm{Cov}(\beta_1 M + \epsilon_1, \beta_2 M + \epsilon_2) = \mathrm{Cov}(\epsilon_1, \epsilon_2) = 0 \, .$$

It follows that $\mathrm{Cov}(X, Y \mid M) = 0$. A similar argument could be made for the chain and the inverse chain. In the collider case, however, conditioning on M renders X and Y dependent if both associated regression coefficients are nonzero, and we get no testable implication on the conditional covariance $\mathrm{Cov}(X, Y \mid M)$. In other words, the implied unconditional and conditional independencies are reversed by conditioning on M (see Table 1).

So far we have only included paths with a single arrowhead (directed paths), but triplets with bi-directed arrowheads would work the exact same way. For example, the model obtained by taking the chain model and replacing the path $X \rightarrow Y$ with a bi-directed arrow, yielding $X \leftrightarrow M \rightarrow Y$, would still be a chain, because the variable in the middle (from now referred to as the "midpoint") has one arrowhead going in and one arrowhead going out. Replacing the second path with a bi-directed arrow, yielding $X \leftrightarrow M \leftrightarrow Y$, would result in a collider model, because the midpoint has two arrowheads pointing to it. Alternatively, one could consider that every path with bi-directed arrowheads is simply an expression of an unobserved variable with two paths with directed arrows, e.g., $X \leftrightarrow Y$, would become $X \leftarrow L \rightarrow Y$. The same criterion as before is then applied to the expanded model that includes the latent variable L. Results with respect to the observed variables from the model with bi-directed paths and the model with latent variables would be identical.

We can generalize these principles to arbitrarily large models, using the notation defined previously. First, if there exists no route at all between two variables X and Y, then the graphical model implies that X and Y are independent. However, this can only be the case if the model consists of at least two independent parts that are not connected by any paths. Such models rarely occur because they would only be used if the data to be modeled actually consisted of two independent subsets. In the previous examples we have considered all possible routes of length 2 (i.e., routes that consist of two paths) between X and Y. We have seen that X and Y are guaranteed to be uncorrelated only if the route between them contains a collider. This translates to arbitrary models as follows. We call a given route between two variables X and Y *open* if it does not contain any colliders. Likewise, if

[4]This model is very familiar to most psychologists as the *full mediation model*.

Table 1
Names given to the four "building block" models along with the regression equations they represent and the vanishing covariances they imply.

			Independence implication	
Model	Name	Equations	Unconditional	Conditional
$X \to M \to Y$	chain	$Y \sim M\ ;\ M \sim X$	none	$X \perp Y \mid M$
$X \leftarrow M \leftarrow Y$	inverse chain	$X \sim M\ ;\ M \sim Y$	none	$X \perp Y \mid M$
$X \leftarrow M \to Y$	fork	$X \sim M\ ;\ Y \sim M$	none	$X \perp Y \mid M$
$X \to M \leftarrow Y$	collider	$M \sim X + Y$	$X \perp Y$	none

it does contain a collider, we may call it *closed*. The implication $\mathrm{Cov}(X, Y) = 0$ holds if and only if there exists no open route between X and Y. Note that there can be many routes between X and Y, but we need to require that all of these routes are closed. Using this simple rule, we can identify the variable pairs that are guaranteed to be uncorrelated simply by tracing the paths in a model.

This previous rule only identifies unconditional independence implications and did not consider that we may condition on a set of variables $\mathbf{Z}$. In our previous examples, we saw that conditioning on the midpoint potentially *induces* a correlation for colliders and *removes* a correlation for chains, inverse chains or forks. We can generalize this as follows. A route between X and Y is *open* with respect to $\mathbf{Z}$ if for *every* triplet of variables in this route that form the collider model, the midpoint of this collider triplet is in $\mathbf{Z}$, and for *every* triplet of variables that forms a chain, fork, or inverted fork, the midpoint of this triplet is *not* in $\mathbf{Z}$. Equivalently, we may also say that a route between X and Y is *closed* with respect to $\mathbf{Z}$ if for *at least one* triplet of variables in this route that form the collider model, the midpoint of this collider triplet is not in $\mathbf{Z}$, or alternatively, for *at least one* triplet of variables that forms a chain, fork, or inverted fork, the midpoint of this triplet is in $\mathbf{Z}$. For two variables X and Y and a given set $\mathbf{Z}$ of other variables, the implication $\mathrm{Cov}(X, Y \mid \mathbf{Z}) = 0$ must hold if and only if there exists no route between X and Y that is open with respect to $\mathbf{Z}$.

Now, for any given variable pair (X, Y) in the model, we can have one of the following cases.

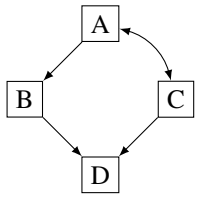
1. X and Y are connected by a path. Then no unconditional independence statement is implied, and it is impossible to find a set $\mathbf{Z}$ that leads to a conditional independence, because the direct path between X and Y always remains an open route. We then say that X and Y are d-connected.

2. X and Y are not connected by a path, and there exists no open route between them. This implies $\mathrm{Cov}(X, Y) = 0$. We then say that X and Y are d-separated.

3. X and Y are not connected by a path, but there exist open routes between them. However, there exists a set $\mathbf{Z}$ such that all routes between X and Y are closed by $\mathbf{Z}$. This implies $\mathrm{Cov}(X, Y \mid \mathbf{Z}) = 0$. We then say that X and Y are d-connected, and are d-separated given $\mathbf{Z}$. Note that there may be more than one set of variables that closes all routes.

4. X and Y are not connected by a path, but there exist open routes between them that cannot be closed by any set $\mathbf{Z}$. This yields no implication, and can only occur in models with bi-directed arrows and / or cycles. We then say that X and Y are d-connected and cannot be d-separated.

Figure 1. A simple SEM along with its testable implications.



$\mathrm{Cov}(B, C \mid A) = 0$
$\mathrm{Cov}(A, D \mid B, C) = 0$

To illustrate these cases, examine the simple model presented in Figure 1. Most of the variables in this model are directly connected with each other, thus no (conditional) independence can emerge. The two interesting variable pairs which are not connected by a direct path are (A, D) and (B, C). There are two relevant open routes between A and D, namely $A \leftrightarrow B \to D$ and $A \to C \to D$. Both are chains. If we condition on the midpoint of both chains, namely B and C, then these routes are closed while no other routes are opened. Therefore, our first implication is $\mathrm{Cov}(A, D \mid B, C) = 0$. The second implication of this model is based on the variable pair (B, C). There are again two routes between B and C, namely $B \to D \leftarrow C$, and $B \leftarrow A \leftrightarrow C$. The first route is not open, because the midpoint D is a collider. Only the second route is open, but can be closed when conditioning on the midpoint A. Conditioning on A does not open any new routes. Therefore our second testable implication is $\mathrm{Cov}(B, C \mid A) = 0$. If one would in fact also condition on D, the observed covariance between B and C would no longer be guaranteed to vanish.

In summary, d-separation is a graphical criterion that allows us to enumerate the unconditional and conditional van-

ishing covariances between variables in a SEM. All of these d-separation constraints imply certain (conditional) independencies, and thus each of these constraints provides a local test of the model. Before we describe how these independencies are tested on actual data, we now present the second graphical criterion for local fit evaluation.

Trek separation

The zero conditional covariances implied by d-separation are only directly testable if the set $\mathbf{Z}$ on which we need to condition consists entirely of observed variables. In typical SEMs where the only observed variables are the manifest indicators of latent variables, we will therefore not get any conditional covariance implications at all. However, in models with latent variables there are additional constraints that cannot be identified using d-separation. A different graphical criterion called "trek separation"[5], or t-separation, can be used to apply local fit evaluation to linear models in such cases as well. However, one important limitation of t-separation is that it does not apply to models that contain cycles.

This second type of local tests was first considered by Spearman (1904) in his analysis of vanishing tetrad constraints. Vanishing tetrad constraints apply to foursomes of variables from the model, say X, Y, Z, and W. The tetrad τ_{XYZW} is defined as a difference between two products involving four covariances:

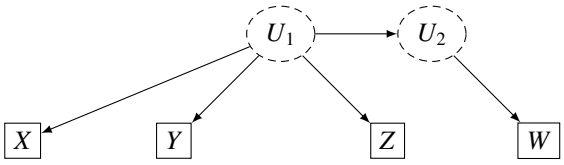
$$\tau_{XYZW} = \mathrm{Cov}(X, Y)\mathrm{Cov}(Z, W) - \mathrm{Cov}(X, Z)\mathrm{Cov}(Y, W) \ .$$

A tetrad τ_{XYZW} is said to *vanish* if $\tau_{XYZW} = 0$ holds in the population. Though in general, the set of vanishing tetrads of a particular SEM depends on the model parameters, there are a number of tetrads that must always vanish under any set of models that share the same graphical structure. Therefore, just like the d-separation constraints considered above these *vanishing tetrad constraints* can be derived purely from the model structure without any reference to data.

The t-separation criterion is not commonly applied in the field, and most researchers instead resort to the recommendation by Bollen and Ting (2000) to determine vanishing tetrads empirically from the implied covariance matrix of a model instance with randomly chosen parameters (Johnson & Bodner, 2014). There is no inherent advantage of using this simulation-based approach as opposed to a graphical criterion, and we believe that the wide-spread use of the simulation approach is partly due to the somewhat intricate form of the original t-separation argument. We provide here an alternative, yet equivalent, definition that we believe is more accessible.

First, we define a *directed* route as a route that consists only of forward-pointing directed arrows, e.g. $X \to Y \to Z$ but not $X \to Y \leftarrow X$. For two sets of variables $\mathbf{I}, \mathbf{J}$, we define $C(\mathbf{I}, \mathbf{J})$ as the set of those variables from which there is a directed route to at least one variable in $\mathbf{I}$ and a directed route to at least one variable in $\mathbf{J}$. In a graphical model, $C(\mathbf{I}, \mathbf{J})$ is referred to as the set of "common ancestors" of $\mathbf{I}$ and $\mathbf{J}$. The tetrad representation theorem (Spirtes et al., 2000) says that tetrad $\tau_{I_1 J_1 J_2 I_2}$ vanishes if and only if one of the following two conditions is met:

Figure 2. An example latent variable model with vanishing tetrads.



1. There exists a variable M_I that lies on every directed route from $C(\mathbf{I}, \mathbf{J})$ to $\mathbf{I} = \{I_1, I_2\}$ (the "outer pair" of the tetrad variables).

2. There exists a variable M_J that lies on every directed route from $C(\mathbf{I}, \mathbf{J})$ to $\mathbf{J} = \{J_1, J_2\}$ (the "inner pair" of the tetrad variables).

Note that these two conditions are not mutually exclusive – a single variable M could fulfill them both, i.e., $M = M_I = M_J$. The variable M_I or M_J in the previous conditions is sometimes called "bottleneck" or "choke point".

To see how this rule can be applied, we provide a small example in Figure 2. In this example X, Y, Z are indicators of a latent variable U_1 and W is the *single* indicator[6] of a latent variable U_2. We are interested in whether there is a vanishing tetrad constraint. Any tetrad will vanish if one of the conditions above holds, namely that there is a variable M_I or M_J that serves as a bottleneck for all effects of sets of common ancestors. In this example, there are only four observed variables. For any two pairs $\mathbf{I}$ and $\mathbf{J}$ we can form from these observed variables, the set $C(\mathbf{I}, \mathbf{J})$ consists of a single variable, U_1, because U_2 is only an ancestor of W, but not of any other variable. Now let us consider the tetrad τ_{XYZW}, where $\mathbf{I} = \{X, W\}$ and $\mathbf{J} = \{Y, Z\}$. The directed routes of $C(\mathbf{I}, \mathbf{J}) = \{U_1\}$ on X, W are $U_1 \to X$, and $U_1 \to U_2 \to W$. U_1 lies on every directed route, and is thus a bottleneck, satisfying the first tetrad condition noted above, and thus the tetrad τ_{XYZW} will vanish. Of course this implies that tetrads that are formed on the same difference of covariances (but with reversed order of variables) will also vanish, e.g, the tetrad

[5]The term *trek* refers to certain pairs of directed routes (Spirtes, Glymour, & Scheines, 2000). However, we will explain t-separation in a different, equivalent manner which does not need this term.

[6]We assume that the reliability of the measurement of W as an indicator of the latent U_2 is known, and thus a constraint on the variance of the latent U_2 can be used to allow for a single indicator item and still have a globally identified model.

τ_{YXWZ}, or τ_{XZYW}. In fact, all three unique tetrads that can be formed in this model will vanish.

Our general definition above implies *all* vanishing tetrads that can be read off a graphical model. In fact, this definition subsumes the types of vanishing tetrads from a previous typology for measurement models by Kenny (1979). In this typology, X, Y, Z, W are assumed to be indicators each connected to a single latent variable. Kenny (1979) lists the following conditions as implying a vanishing tetrad τ_{XYZW}.

1. Homogeneity within constructs: X, Y, Z, W are indicators of the same latent U. In this case, U itself can act as both M_I and M_J in the above criterion.

2. Homogeneity between constructs: X, W are indicators of U_1 and Y, Z are indicators of U_2. U_1 and U_2 are correlated. In this case, U_1 can act as M_I and U_2 can act as M_J[7].

3. Consistency of epistemic correlations: X, W, Y are indicators of $U1$ and Z is an indicator of U_2. Again, U_1 and U_2 are correlated. In this case, U_1 can act as M_I in our definition (but note that U_2 cannot act as M_J).

Thus, one can verify that t-separation implies all tetrads from the typology by Kenny (1979). The typology is very useful, because it allows to classify the potentially large number of tetrads that are yielded by a single model. However, the typology is only well defined for measurement models in which indicators load only on one single construct. As soon as this is violated, the distinction between the different types of tetrads becomes blurry, because tetrads may then belong to more than one category. In contrast, the t-separation criterion allows one to identify *all* vanishing tetrads implied by an arbitrarily complex model, including those that have indicators load on more than one latent variable.

We conclude this explanation with a short mathematical argument why our graphical criterion causes the tetrads to vanish. This argument is a simplified version of the tetrad representation theorem (Spirtes et al., 2000). The tetrad τ_{XYZW} is in fact the determinant of the 2×2 matrix[8]

$$\begin{pmatrix} \mathrm{Cov}(X,Y) & \mathrm{Cov}(X,Z) \\ \mathrm{Cov}(W,Y) & \mathrm{Cov}(W,Z) \end{pmatrix},$$

which is zero if and only if there exist a λ such that

$$\mathrm{Cov}(X,Y) = \lambda\,\mathrm{Cov}(W,Y) \text{ and } \mathrm{Cov}(X,Z) = \lambda\,\mathrm{Cov}(W,Z)\,. \tag{3}$$

Assume that there exists a single variable M_I that serves as a bottleneck and therefore transmits all effects from the $C(X, Z, Y, W)$ to the outer pair of variables $\mathbf{I} = \{X, W\}$. Then we can decompose

$$\mathrm{Cov}(X,Y) = \beta_{X|M_I}\mathrm{Cov}(M_I,Y)\,;\mathrm{Cov}(X,Z) = \beta_{X|M_I}\mathrm{Cov}(M_I,Z)$$

and

$$\mathrm{Cov}(W,Y) = \beta_{W|M_I}\mathrm{Cov}(M_I,Y)\,;\mathrm{Cov}(W,Z) = \beta_{W|M_I}\mathrm{Cov}(M_I,W)\,,$$

which leads to Equation 3 using $\lambda = \beta_{X|M_I}/\beta_{W|M_I}$.

Effect sizes and statistical tests for local fit evaluation

We now discuss how local tests would actually be performed once data are available. In the Supporting Information, we provide R code that performs the tests on simulated data without using any external packages.

Effect size and tests for d-separation constraints

Every d-separation statement of the form "X and Y are d-separated by $\mathbf{Z}$" leads to a statistically testable constraint on the probability distributions that are compatible with the assumed model. In the simplest case, when the set $\mathbf{Z}$ is empty, d-separation implications take the form of unconditional independence. The most basic way to test unconditional independence between two variables with a statistical test is to compute the correlation coefficient and apply a significance test. If the two variables are statistically independent, their correlation coefficient is expected to be zero and tests of it should (under repeated sampling) only yield significant results with a frequency equal to the Type I error rate of the test[9]. Things become more complicated when the set $\mathbf{Z}$ is not empty, which implies a *conditional* independence between X and Y. A general strategy for testing conditional independence is regressing both X and Y on $\mathbf{Z}$, and then testing for independence between the residuals of these regressions. If X and Y are indeed conditionally independent given $\mathbf{Z}$, then these residuals should be statistically independent as well. Performing this analysis using linear regression leads to the partial correlation coefficient $r_{XY.\mathbf{Z}}$, where the period behind the two variables in the subscript denotes the variables that are being partialled out. This partial correlation coefficient is a natural effect size measure for d-separation constraints, since correlation coefficients are very familiar to applied researchers.

[7] In this case, the common ancestor of X, W, Y, Z is the implicit latent variable represented by the bi-directed arrow $U_1 \leftrightarrow U_2$. It is helpful to replace bi-directed paths by explicit latent variables, as explained above, before evaluating the graphical criterion for vanishing tetrads.

[8] Larger submatrices can be considered. Those yield pentad, and even higher order constraints. Sullivant, Talaska, and & Draisma (2010) give a detailed account of such constraints and show that both t-separation and d-separation can be derived from their general definitions.

[9] Such tests come with all advantages and disadvantages of significance testing (Nickerson, 2000; Wagenmakers, 2007). Some practitioners might prefer to perform Bayesian statistics, e.g., in the form of a Bayes Factor, or a posterior distribution with a Bayesian credible interval (Kruschke, 2010).

An important caveat is that conditional independence *only* implies zero partial correlation if the relationships between X and $\mathbf{Z}$ and between Y and $\mathbf{Z}$ are indeed linear. A nonzero correlation between regression residuals does not immediately mean that the tested variables are truly conditionally dependent. Instead, the regression may also have failed to capture the form of dependence between X or Y and $\mathbf{Z}$, and therefore have generated incorrect residuals. Fortunately, the basic approach of examining residuals generalizes to many kinds of regression, which enables semi-parametric conditional independence testing (Shipley, 2002). This means that instead of using residuals from linear regression models, we could estimate flexible semi-parametric models that approximate the true relationships more closely than linear trends, and rely on residuals from these models instead. We shall illustrate this later in a worked example.

Significance testing in general and in the case of testing conditional independence using correlation coefficients is not without shortcomings. One possible concern is that rejection of the null hypothesis does not inform us about the amount of misspecification, especially in large samples. In addition to the p-value, we may of course also inspect the confidence interval, which puts the focus more on the magnitude of the effect, and the uncertainty of the estimate. By examining the midpoint of the interval (the point estimate) researchers can directly examine the magnitude of the correlation coefficient. Correlation coefficients are one possible measure of the size of the effect, and being on a metric that is readily interpretable by most researchers should facilitate judgment about the importance of a rejected significance test. As an example, in a large sample of several thousand participants, a correlation coefficient of .01 may indeed yield a significant result, but researchers may feel any attempt to "repair" such a small violation may result in overfitting of the model to the dataset at hand. A difficulty in this approach is the reliance on cut-off values, as there is some uncertainty about what cut-off should be chosen, and if a cut-off can be universally be used in all circumstances.

Besides judging the absolute value of the correlation coefficient, we may also conduct tests of close fit, where observed correlation coefficients are *not* tested against zero, but some other value that is chosen to be of sufficiently small magnitude. For example, one may test whether the observed correlation coefficient is significantly more extreme than ±.05 or some other reasonably small value. The resulting p-value of this test then indicates whether an observed correlation coefficient deviates significantly from a minimally acceptable amount of misfit. The same approach is widely used, for example, to construct a test of close fit based on the RMSEA fit index.

Significance tests of correlation coefficients against non-zero values are not routinely implemented in standard software. However, there are several ways to obtain such tests. The way that we propose to conduct these tests is to first use Fisher's Z transformation of both the observed correlation coefficient and the minimum tolerable correlation that one wants to test against. Correlation coefficients that have been transformed using Fisher's Z transformation have an approximately normally distributed sampling distribution with a standard error of $\frac{1}{\sqrt{N-3}}$[10]. To test for deviations that could go in either direction, we square the Fisher's Z-transformed value, and then perform a one-sided significance test using a χ^2 distribution with non-centrality parameter that is identical to the squared value of a Fisher's Z-transformation of the non-zero value that one wants to test against.

We provide R code for a demonstration of this type of test for the simple model in Figure 1, but for now we defer any further numerical examples to later sections. Here it suffices to say that the d-separation constraints and the implied (conditional) independencies can be statistically tested (either against zero or a meaningfully small value), and their effect size can be observed.

Effect size and tests for t-separation constraints

The parametric statistical tests of vanishing tetrads follow naturally from the definition of the tetrad as a difference between two products of covariances. There are several statistical tests that can be applied. Wishart (1928) proposed a test statistic formed by dividing the value of the tetrad by a standard error derived from the covariance matrix of tetrads. Other test statistics, e.g., by Kenny (1974) compute canonical correlations. The test statistic of Wishart's test converges to a normal distribution in large samples. In small samples, where this convergence is questionable, bootstrapping of the standard error is highly recommended.

In addition to the significance tests, we may again consider an effect size and its confidence intervals. Tetrads computed from a correlation matrix are bounded by -1 and 1. As such they are on a familiar effect size metric and their magnitude can be easily assessed. Like in the case of the d-separation constraint, we may want to perform a test of close fit against some non-zero but minimally acceptable value of misfit for the tetrad. This test of close fit is very similar to the one presented above. Here we first standardize both the observed tetrad and the minimally acceptable value by dividing both by the standard error of the tetrad, and upon squaring use a non-central χ^2 distribution to derive p-values.

We provide R code to compute tetrads for the simple model in Figure 2. We will again defer actual numerical examples to later sections. For now, it again suffices to say that t-separation constraints and the implied vanishing tetrads can

[10] When computing correlation coefficients from residuals of regression models, the degrees of freedom are further diminished by the size of the conditioning set.

be easily parametrically tested (either against zero or a meaningfully small value), and their effect size can be observed.

Software

Whether or not a method becomes widely used and adopted by applied researchers depends in no small part on software. To our knowledge, none of the currently available software programs that estimate SEMs offers even an option for local fit evaluation. This includes the open-source software lavaan (Rosseel, 2012), which we have used in our example code, but also all of the currently available commercial SEM software. The web-based DAGitty software (Textor et al., 2011) provides an interface to draw a graph and returns a list of implied conditional independencies, but does not itself compute the tests. The accompanying R package 'dagitty' by Textor et al. (2017) fills this existing gap and provides functions in R that perform all relevant tasks. In particular, the software can read a graphical model using both DAGitty notation, or lavaan notation. The software can find every possible *d*-separation, and *t*-separation constraint, and can display those before any data has been collected. Once data are collected, the program can perform all conditional independence, and vanishing tetrad tests. Both normal theory standard errors, or bootstrapped standard errors are supported. The program reports significance tests, along with confidence intervals, the magnitude of the correlation or tetrad as a simple effect size measure, and an optional test of close fit. For *d*-separation constraints only, the software can perform both a parametric and a semi-parametric conditional independence test based on local polynomial regression (LOESS).

Illustrative examples

We will now present some intentionally simplified examples to showcase the basic behavior of local tests and compare it to other, more traditional forms of fit evaluation. Note that these examples are for illustrative purposes only. We do not attempt a full comparison, for which large simulation studies, and not simple examples, would be needed.

The presented local tests rely purely on the assumed graphical structure, and therefore they can be derived before data has been collected. This means that the researcher has a chance to think about what assumptions he or she is making, and whether these assumptions seem plausible, given the current theoretical knowledge. For example, a researcher may realize that his or her assumptions imply that two variables in the model must be independent (*d*-separated), given another set of variables in the model. A-priori this may or may not be plausible, and the local test encourages this kind of critical thinking.

Local tests can be performed immediately after data collection, even before the model itself has been fitted (as the local tests do not require global identification). This is in contrast to the χ^2 test, and any procedure or fit index based on it, which requires a fitted model. Any local test that fails to be refuted during this stage of testing provides some evidential support for the model, and every local test that is rejected weakens support for the model, inviting the researcher to think about what part of the model is likely incorrect. The fact that the tests can be conducted without having to worry that a model may have trouble converging to a maximum likelihood solution is a potential advantage of local tests.

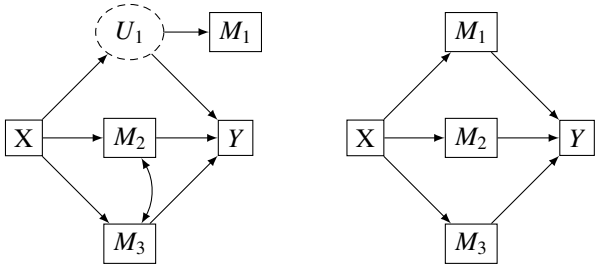
Local tests can also provide information about which part of a large SEM violates the observed data. This is not to say that local tests will always be able to pinpoint the exact misspecification, but unlike a purely global test they can at least sometimes succeed in doing so. In this regard, local tests are similar, but not identical to modification indices, or standardized residuals. In cases in which it is known that the misspecification is due to a missing arrow, both modification indices and local tests can inform us which specific arrow needs to be added. But in other cases, in which misspecification is, e.g., due to a missing variable in a model, or several mis-oriented arrows, modification indices could be misleading. Local tests, on the other hand, do not immediately suggest certain arrows to be added, but inform the researcher which implications of his or her assumptions are violated. They therefore encourage researchers to think critically about these assumptions, why they could be violated, and how this violation could be remedied. This may result in the inclusion of another arrow, but it can also result in different changes to the model, e.g., inclusion of a latent variable. Finally, a difference between local tests and modification indices is that the latter always require a fitted model, whereas the local tests can be computed before model fitting.

Through a small set of worked examples, we now demonstrate the behavior of both types of local tests discussed in this paper. We provide R code for all examples as Supporting Information. This code includes the data-generating models, all standard output of the SEM software *lavaan* (Rosseel, 2012), and the local tests provided by the R package *dagitty* (Textor et al., 2017).

Identifying misfit location in path models

To demonstrate how local tests operate, we first use simple path models consisting only of manifest variables that are all assumed to be perfectly measured. Such models yield tests based on *d*-separation constraints. Assume that the true data-generating model looks like Figure 3 (a). In this example model, variable X has an indirect effect on Y that is mediated by the observed variables M_2 and M_3 (whose error terms are correlated with each other), and the unobserved variable U_1. We only observe a proxy of U_1, namely M_1, which in this model is not a mediator, but simply caused by the unobserved U_1. An applied researcher, however, proposes the model in Figure 3 (b) that is identical to the true model, except that

Figure 3. Worked example of *d*-separation tests.



(a) true model (b) assumed model

error terms of M_2 and M_3 are uncorrelated with each other, and that U_1 is not in the model, and in its place is the variable M_1. This model – contrary to the truth – assumes that the effect of X on Y is fully mediated by M_1, M_2, and M_3. It also incorrectly assumes the absence of any common cause of M_2 and M_3 except X. The model of the researcher has certain implications, encoded in the following *d*-separation constraints. First, X is *d*-separated from Y given M_1, M_2, and M_3. Second, every pair of the variables M_1, M_2, and M_3 is *d*-separated from each other given X. Together there are thus a total of four *d*-separation constraints.

We can now compare these constraints to the ones in the true model. The first constraint of the assumed model is that X and Y are *d*-separated from each other, given the three variables M_1, M_2, and M_3. This does not hold in the true model (because the route that traverses U_1 is still open). Likewise, the constraint that M_2 and M_3 are *d*-separated given X is incorrect, because in the true model these two variables are connected by a bi-directed arrow. On the other hand, the two remaining constraints that M_1 is *d*-separated from both M_2 and M_3 given X is correct in both the assumed and the true model. Hence, we would expect that two of the local tests will be violated, and that the remaining two tests would not be violated.

Based on the true model, we simulated a single dataset with 1000 datapoints. We used standardized variables, and set all path coefficients to .4. We fitted the incorrect model, and observed simple measures of global fit and tests of local fit. The global χ^2 test of the model soundly rejected it, $\chi^2(4) = 311.2, p < .001$. Other global fit indices also suggested misfit, CFI = .821, RMSEA = .277, and SRMR = .129. There are a variety of other fit indices that could have been computed, but for demonstration purposes we only report this subset. Based on the global tests alone, we cannot determine whether the assumption of full mediation, or the assumption of residually uncorrelated mediators (both expressed in the *d*-separation constraints) is more likely violated.

The local tests yielded the following results. The constraint between variables M_1 and M_2 was not violated, $r_{M_1M_2.X} = .02, p = .55$ thus indicating that this particular restriction of the model is in agreement with the data. The small effect size ($r = .02$), and the non-significant test of close fit ($p = .85$ against a minimally acceptable value of .05) further strengthen this belief. Likewise, the constraint between variables M_1 and M_3 was not violated, $r_{M_1M_3.X} = .02, p = .64$, again with a very small effect size and a test of close fit with a high p-value of .88. On the other hand, the constraint between variables M_2 and M_3 showed a strong violation, $r_{M_2M_3.X} = .46, p < .001$, indicating to the researcher that the model may be misspecified in a way that this conditional independence is incorrect. The relatively large effect of $r = .46$, and a test of close fit with a p-value much smaller than .001 corroborate this view. Lastly, the constraint between variables X and Y also showed strong violation, $r_{XY.M_1M_2M_3} = .33, p < .001$ (test of close fit, $p < .001$), and points the researcher to another part of the model that exhibits misfit, namely that the three variables M_1, M_2, and M_3 do not fully mediate the effect between X and Y.

Importantly, these local tests do not directly suggest any particular modification (e.g., adding a directed arrow), but point to the specific implications of conditional independence that are violated. This is a general feature of local tests that are based on *d*-separation. Instead of suggesting direct fixes to a model, they confront the researcher with the implications of the structural assumptions that were made in the model, and whether or not they are refuted by data. However, knowing the graphical rules that these tests are based on does provide some immediate clues as to which modifications could, and which could not, remedy the misfit. Specifically, if a *d*-separation implication is falsified, then possible reasons include that (a) a direct path between the separated variables is missing; (b) an element of the conditioning set is not perfectly measured; (c) an element of the conditioning set is truly a collider on some path between the separated variables. We can immediately see, for instance, that possible changes that could repair the failed test $r_{XY.M_1M_2M_3} = 0$ include (a) adding a path from X to Y; (b) introducing a measurement model for one of the M's, (c) reversing an arrow from Y to an M. We also realize that actions that will *not* repair the implication include (a) adding any arrow between the M's, (b) reversing an arrow from X to one of the M's, or (c) introducing a measurement model for X and Y.

At this point, it is informative to compare and contrast the results of the local tests with those of the more commonly used modification indices and the inspection of the standardized residuals. Using the exact same data and model as above, we may also request modification indices. This model yields a total of 21 modifications indices. The three largest indices all have a value of 226.3. They suggest either the addition of a bi-directed arrow between M_2 and M_3, or the addition of a directed arrow either from M_2 to M_3 or vice versa. What all of these three modification indices share is that when their suggested change is implemented, the re-

sulting model no longer violates the local test that suggested conditional independence between M_2 and M_3. In fact, if one were to compute a p-value for this modification index (which is a one degree of freedom χ^2 test), it would be numerically very similar to the corresponding local test. In this situation, the local test and the modification index are virtually identical.[11] This will always be the case in which a single, unique local test can be relaxed through the addition of a path or bi-directed arrow. Likewise the matrix of standardized residuals showed that the largest residual was between M_2 and M_3.

That this fortunate behavior of the modification index is not guaranteed can be seen by the second local test. After modifying the model by adding an arrow between M_2 and M_3, 18 additional modification indices can be identified. Four of these indices have the same value of 52.9. One of these modification indices suggests connecting X and Y with a reciprocal directed arrow. The matrix of standardized residuals also suggest that the largest residual is contained in the covariance of X and Y, with all other residuals being quite small. The remaining three modification indices suggest connecting Y with one of the three mediators M_1, M_2, or M_3 with a bi-directed arrow. What all of these modification indices have in common is that when implemented the resulting model would not violate the other local test anymore. However, what these tests also have in common is that they all suggest an incorrect modification of the model. In other words, none of the resulting models (even though better-fitting) aligns with the true model.

This small example demonstrated that global tests can inform the applied researcher that a model does not fit the data. Local tests on the other hand inform the applied researchers which of the model's implications in the form of conditional independencies is violated. This is not to say that local tests can always identify a correct model, but they do identify local sources of misfit.

Identifying misfit location in latent variable models

A second example involves latent variables, and hence the use of tetrad tests. Consider the data-generating model in Figure 4, in which a latent variables L_X causes another latent variable L_Y. Both latent variables have three indicators each. However, some of the manifest indicators are correlated with each other. In particular variable X_1 is correlated with X_3, indicating that the latent construct L_X does not fully capture all relationships between the indicators. Also, X_1 is correlated with Y_1, indicating the potential presence of some shared methods variance.

Now suppose that a researcher fits a model that is identical to the true model but does not include the correlations among indicator variables. This model has additional constraints on various tetrads that are not present in the true model. We generated a single dataset with $1,000$ datapoints from the true model, using again completely standardized variables. All path coefficients from latent variables to the indicators and the coefficient between the latents were set to .7. The correlations between individual items X_1 and X_3, and X_1 and Y_1 were set to .25. Fitting this model yielded a large χ^2 statistic ($\chi^2(8) = 263, p < .001$). Likewise CFI (.892) and RMSEA (.179) suggested rejection. The SRMR suggested adequate fit (.049). In fact, even fitting an unrestricted model (the first step of the four-step testing procedure suggested by Mulaik and Millsap (2000) in which every item is allowed to load on any of the two factors, essentially an exploratory factor analysis) yielded bad fit ($\chi^2(4) = 150, p < .001$). In addition to these global tests, we can also easily compute the implied vanishing tetrads of this model. This model in particular yields a total of 27 vanishing tetrads.

Unlike in the case of d-separation constraints, tetrad tests are a bit more complicated to evaluate. One reason is that the tetrad tests tend to be much more numerous, and secondly, they do not map to conditional independencies between observed variables. The large number of tetrad tests means that adjustment for multiple testing is usually recommended. When we adjusted the p-values of the 27 tetrad tests in our example model with the Bonferroni-Holm method, we obtained 12 significantly violated tetrads (at $\alpha = 0.05$). For example, two of the most strongly violated tetrads were:

- $\tau_{X_1,X_2,Y_1,X_3} = -.13$, $p = 2.4 \times 10^{-17}$
- $\tau_{X_1,Y_1,Y_2,X_3} = .10$, $p = 7.6 \times 10^{-15}$.

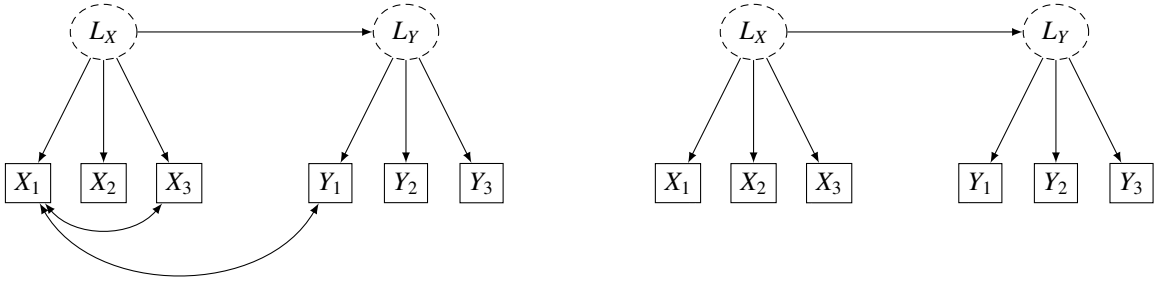
Even though their absolute effect size was somewhat modest, tests of close fit (against an arbitrary value of .05), also indicated that the observed tetrad was significantly larger than this threshold, with p-values far below .001, even after adjustment for Type I error inflation.

Both of these violated tetrad constraints postulate that the set $\mathbf{I} = \{X_1, X_3\}$ can be t-separated from another set $\mathbf{J}$ that includes Y_1 (in the first tetrad $\mathbf{J} = \{X_2, Y_1\}$; in the second tetrad, $\mathbf{J} = \{Y_1, Y_2\}$). So this would indeed suggest to add the missing covariance between X_1 and Y_1. That same conclusion would also be reached when examining any set of most strongly violated tetrads. However, instead of adding residual covariances, another course of action is also suggested by the local tests: All 13 significantly violated tetrads involve the indicator X_1. Therefore, it would also appear reasonable to drop this indicator from the model altogether. Indeed, the result would also be a well-fitting model in this case.

In contrast to these tetrad tests, we may examine standardized residuals, and the 21 possible modification indices. The

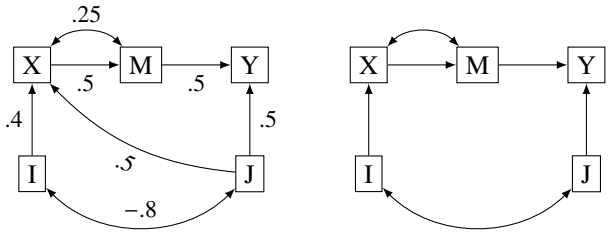
[11] In the local test, the test statistic is simply the ratio between the estimate and its standard error. If we square this quantity, we get a test statistic that is very close, but not identical to the corresponding modification index. The small difference is merely due to the fact that the modification index is in fact a score test, while in the regression model, we use a Wald test.

Figure 4. Worked example of tetrad tests.



(a) true model (b) assumed model

Figure 5. Worked example of a non-converging path model.



(a) true model (b) assumed model

largest standardized residuals were concentrated on the covariances between X_1 and Y_1, X_1 and Y_3, and Y_1 and X_3. The modification index with the largest expected difference in the χ^2 statistic is the covariance between X_1 and Y_1, 232.91. It may be plausible to drop a variable such as X_1 from the model based on this set of modification indices, but it may also be plausible to add a bi-directed arrow. Re-examining the resulting modification indices yields two suggested modifications, either adding a covariance between X_1 and X_3, or a directed arrow from L_Y to X_2, both with an expected change in the χ^2 statistic of 29.31. Therefore, one of the modification indices correctly identifies the second missing covariance, whereas the other one does not.

Local tests with non-converging models

In the next example, we want to demonstrate how local tests can be used when a global model does not converge (and thus modification indices cannot be computed). Consider the model in Figure 5 (a).

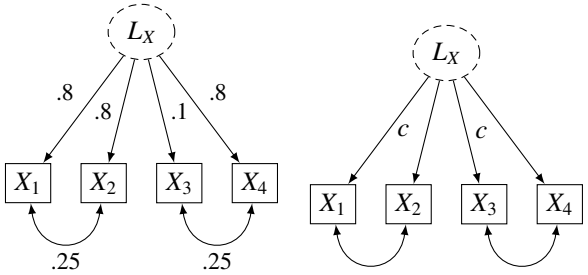
For this model we chose specific path coefficients that lead to convergence problems: The variable I in this model effectively acts as an instrumental variable to render the part involving X and M identified. However, the effect of J on X counters the effect of I on X because I and J are negatively correlated. The assumed model in Figure 5 (b) does not include the effect of J on X. Therefore, the variable I will appear to be very weakly correlated with X and the model is therefore locally not identified.

Based on the true model, we simulated a single dataset with $1,000$ datapoints, and set standardized path coefficients to the values shown in Figure 5 (a). We tried to fit the incorrect model, but as expected it would not converge. Without convergence it was also impossible to observe any global fit indices, let alone any of the modification indices.

The local tests on the other hand can be performed without any problems. The assumed model implies five conditional independencies. One of these was very strongly violated. This implication states that J and X are independent given I ($r_{JX.I} = .19, p < 10^{-22}$), thus casting doubt on this particular independence. A test of close fit (against the threshold .05) also resulted in a very small p-value, 1.49^{-17}. As we saw earlier, every violation of a local test can be remedied by adding an additional arrow (although without guarantees that this is the correct fix). A researcher faced with these local tests should question the violated assumptions, and think about ways in which they could have been violated. However, in this case, adding the direct arrow $J \rightarrow X$ is in fact the correct course of action, and the model that includes this additional arrow – i.e., the true model – converges without problems, and shows no violations of the two remaining local tests.

We now give a second example of a non-converging model, this time involving the use of latent variables and tetrad tests. At the same time, we use this example to illustrate the interplay between local tests and constraints on the model parameters. Consider the model in Figure 6 (a). A single latent variable U affects all observed variables X_1 to X_4. Both X_1 and X_2, and likewise X_3 and X_4 share an additional covariance. We choose the path coefficients such as shown in Figure 6 (a). Note that one of the items has a very weak loading. In order to be able to estimate this particular model from data some constraints need to be imposed, since otherwise the model is not identified. One possible constraint is shown in the assumed model in Figure 6 (b), in which the factor loading of X_1 and the loading of X_3 are forced to be identical, indicated by the shared letter c in the Figure.

Figure 6. Worked example of tetrad tests for a model with parameter constraints.



(a) true model

(b) assumed model; label *c* denotes equality-constrained path coefficients.

Based on the true model, we simulated a single dataset with 100 datapoints, a smaller sample size deliberately chosen to force non-convergence. The standardized path coefficients were set to the values shown in Figure 6 (a). Then we attempted to fit the incorrect model (that included the equality constraint, because a model without constraints cannot be estimated). This model did not converge.

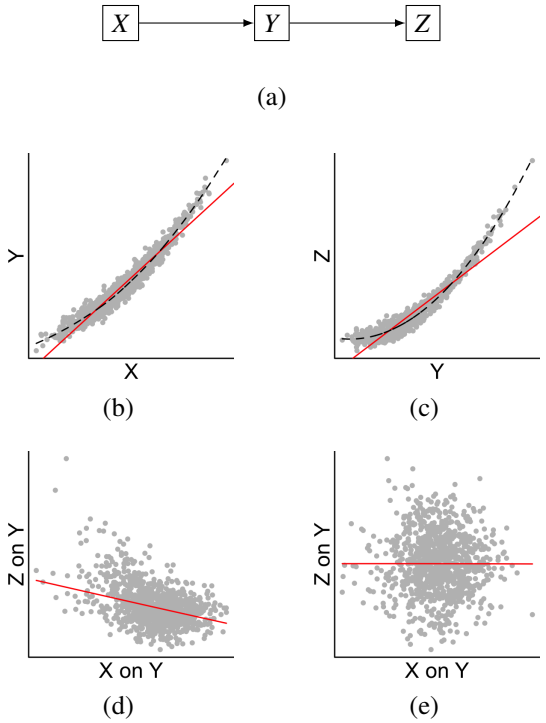
In a next step, we computed the local tetrad tests. In this example, there is only one single tetrad test, $\tau_{X_1,X_3,X_4,X_2} = .04$, $p = .11$, indicating no violation. The small magnitude of the tetrad and a test of close fit (against a value of .05) that yields a p-value of .49 confirm this further. The local test only tests the implications of the structure of the model, and thus the equality constraint (that is necessary to estimate the model) does not influence the local test. The fact that the tetrad test does not show a violation bolsters faith in the actual structure of the model, and suggests that the equality constraint is the likely culprit responsible for the non-convergence. Changing the equality constraint to the other two loadings, e.g., X_1 and X_4 yields a converged model with decent, although not perfect, fit, $\chi^2(1) = 2.826, p = .093$, CFI = .991, RMSEA = .135, and SRMR = .019. In this particular example, modification indices would not have helped to detect the problem even if the model had converged, as modification indices are based on the entire model, including any imposed constraints. Local test only test the structure without regard to the imposed equality constraints, which correctly informs the researcher about the source of the problem.

Non-parametric testing of *d*-separation constraints

Lastly, we give a small example to illustrate how local fit evaluation can be used to disentangle structural and distributional aspects when assessing model fit. Note that this is only possible for *d*-separation but not tetrad tests, since the latter only work with linear SEMs.

To give the simplest possible example, we generated data from the three-variable mediation model shown in Figure 7 (a). Here, we imposed quadratic rather than linear dependencies between variables, as can be seen in Figure 7 (b) and (c), though we used additive Gaussian noise like in a linear SEM. We simulated a dataset with 1,000 datapoints in this manner. The mediation model fits poorly to this dataset, $\chi^2(1) = 222.28$, $p < .001$. RMSEA also indicates poor fit (.47), though CFI and SRMR do not, .957, .016, respectively. The standard local test, which is based on linear partial correlation, also indicates a violation ($r_{XZ.Y} = -.32, p < 0.001$). This is because linear regression fails to capture the true shape of the functional relations between X, Y and Z, which then lets the residuals appear correlated, as can be seen in Figure 7 (d). However, when we instead use local polynomial regression to estimate the functional relations, then the residuals are no longer correlated ($r_{XZ.Y} = -.02$, 95% CI: -.09 to .04; Figure 7 (e)).

Figure 7. Worked example of local fit evaluation with non-normal data. (a) The underlying model structure. (b,c) Linear regressions (red lines) of Y on X (b) and Z on Y (c) fail to capture the quadratic dependencies between those variables generated by our simulation. (d) Residuals of the regressions shown in (b) and (c) are negatively correlated. (e) When using locally polynomial instead of linear regressions, residuals are not correlated.



In the face of such results – a failing parametric test whose semi-parametric version passes – a researcher might conclude that the source of misfit is not the model structure, but

instead the distribution of the data. In contrast, modification indices for our model suggest to either add a direct path or covariance between Z and Y, or a direct path from X to Z. All these modifications would lead to a saturated model with zero degrees of freedom, which of course can fit every covariance matrix. However, such modifications would obscure the true source of misfit, and lead to an incorrect model structure.

In summary, we tried to show through a series of simple examples how local tests can be used, and what kind of information they yield. We also contrasted them with the more commonly used modification indices. What we observe is that local tests first and foremost test the structure and the implied independencies of a model. In some instances the local tests will align with the modification indices, in other instances they will be quite different. Local tests do not generally suggest additions of particular arrows in the model, but directly test the assumptions of the model. This information can then be used to revise a model by thinking about ways that a violated assumption could emerge. As we have demonstrated, local tests can be used even in cases in which global identification of a chosen model is not possible.

Practical implications

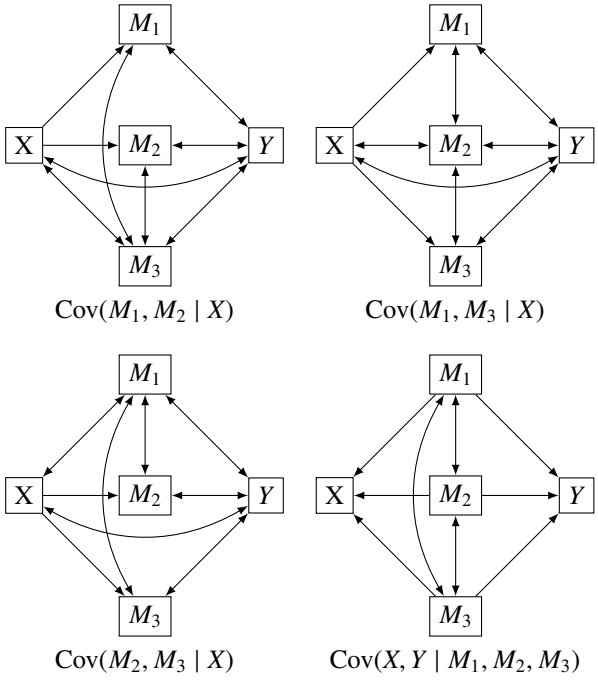
Our examples so far served the purpose of illustrating the behaviour of local tests, and were not meant to represent realistic SEM analyses. In particular, path models without any latent variables are rare (certainly in psychology). As discussed above, researchers who test full latent variable SEMs will most often not rely on a single global test to evaluate their model, but will more often use some variation of the two-step procedure (Anderson & Gerbing, 1988) to evaluate the measurement and structural portions of their models separately. In this final section, we will explore how the local fit evaluation ideas presented in the previous sections can be incorporated into such a two-step test of a full latent variable SEM, which could be considered a more realistic representation of practice than the examples given above.

d-separation testing of latent variable models

Our examples so far suggest that tetrad testing is the only option available for latent variable SEMs. This approach has, however, some disadvantages: first, failing tetrad constraints are more difficult to interpret than failing conditional independencies, and second, complex SEMs can imply hundreds to thousands of vanishing tetrad constraints. Here, we describe a strategy that can be used to apply d-separation-based local fit evaluation to latent variable SEMs.

The key idea of our strategy is that, for each d-separation implication I of a path model M, it is possible to create another path model M_I that has I as its one and only implication. Sometimes this can be done by adding paths to a model, but in general this will not be possible. As a counterexample, consider the path model $X \to Y \to Z \leftarrow W$. This model implies that $r_{YW} = 0$ and $r_{XW} = 0$. To remove the second implication, we would need to add a path between X and W. However, every path that we add would change the first implication from $r_{YW} = 0$ to $r_{YW.X} = 0$.

Figure 8. Generating single-implication path models. These four path models each imply a single one of the four vanishing partial covariances implied by the model in Figure 3. The unique partial covariance that is constrained to 0 is shown below each model.



Instead, the following algorithm always works: Let $r_{YW.\mathbf{Z}} = 0$ be the desired implication. Then (1) create a saturated model by linking all variables with bi-directed paths, (2) remove the path $X \leftrightarrow Y$, and (3) for each $Z \in \mathbf{Z}$, change $X \leftrightarrow Z \leftrightarrow Y$ to $X \leftarrow Z \rightarrow Y$. It is easy to verify that the resulting model (1) implies $r_{XY.\mathbf{Z}} = 0$; (2) does not imply $r_{XY.\mathbf{Z}'} = 0$ for any proper subset $\mathbf{Z}' \subset \mathbf{Z}$ or superset $\mathbf{Z}' \supset \mathbf{Z}$; and (3) no variables except X and Y can be d-separated. If we apply this strategy to our mediation model from Figure 3, we get the four models shown in Figure 8.

We can use such single-implication models to evaluate each d-separation constraint I entailed by the structural part of a latent variable SEM M separately by replacing the structural model with a single-implication model for I. As we shall explain below, this naturally leads to local versions of several known fit indices for latent variable SEMs.

Extending a two-step SEM analysis by local fit evaluation

In many latent variable SEMs, the measurement model contributes the vast majority of degrees of freedom. Since

the global χ^2 is an additive statistic, good fit of the measurement model can obscure poor fit of the structural model. In the two-step procedure (Anderson & Gerbing, 1988), one establishes the validity of the measurement model separately before moving on to test the path model by first testing a version of the model in which the structural part is saturated. Having established the measurement model, we can then assess the fit of the structural model using the χ^2 difference statistic

$$\chi^2_{\mathrm{P}} = \chi^2_{\mathrm{M}} - \chi^2_{\mathrm{SS}} \, ,$$

where M denotes the tested model and SS a structurally saturated model. The degrees of freedom $\mathrm{df_P} = \mathrm{df_M} - \mathrm{df_{SS}}$ are determined solely by the structural model. Examining this difference statistic or a fit index based on it, such as the RMSEA-P (where P stands for paths, because it only focuses on the structural paths) defined by

$$\text{RMSEA-P} = \sqrt{\frac{(\chi^2_{\mathrm{P}} - \mathrm{df_P})}{\mathrm{df_P}(N-1)}} \, ,$$

can often reveal misfit of the structural model despite good fit of the overall model (McDonald & Ho, 2002; O'Boyle Jr & Williams, 2011) – in fact, this is a coarse-grained kind of local fit evaluation. When the source of misfit is found to be the structural model, researchers will often move on to examining standardized residuals and/or modification indices to further isolate the source of misfit. Based on the local tests described above, we can suggest the following alternative: fit each single-implication model M_I devised from M, and compare this to a reference model (either the structural null, or a saturated model). For example, analogously to the RMSEA-P, we can define a "local RMSEA" for each specific implication I as

$$\text{RMSEA}_I = \sqrt{\frac{\chi^2_I - 1}{N-1}} \, ,$$

where χ^2_I denotes the test statistic obtained when replacing the structural model by the one-df model M_I. Along the same lines, we can define local versions of various other fit indices. Lance et al. (2016) recently proposed a taxonomy in which they categorize fit indices as C9-based or C10-based, where C9 and C10 are the 9th and 10th criteria for causal inference from nonexperimental data by James et al. (1982): C9 refers to hypotheses tested by comparing a model to an alternative model with fewer paths, whereas C10 refers to comparing a model to an alternative with more paths (O'Boyle Jr & Williams, 2011). Lance et al. (2016) then suggest indices of the structure

$$\frac{F_{\mathrm{SN}} - F_{\mathrm{M}}}{F_{\mathrm{SN}} - F_{\mathrm{SS}}}$$

for C9 and

$$\frac{F_{\mathrm{M}} - F_{\mathrm{SS}}}{F_{\mathrm{SN}} - F_{\mathrm{SS}}}$$

for C10, where SN stands for the structural null model and F denotes any fit measure that is monotone with respect to nestedness and increases with poorer fit. Such indices are always bounded between 0 and 1. We obtain a local version of any such index simply by replacing F_{M} with F_{M_I}.

To illustrate these ideas, we again used our partial mediation model from Figure 3. We now treat this model as a full latent variable SEM, in which each variable is measured by three indicators, and set the loading of each indicator to 0.8. The path coefficients in the structural part of the model were all kept at 0.4, and we generated a sample of 500 datapoints. We then fitted each of the models in Figure 3, and compared the fit to the structurally saturated model and the original model. For each single-implication model fit, we compute the RMSEA_I as well as C9 and C10 indices using $F = \chi^2/\mathrm{df}$. The results of this analysis are shown in Table 2. First, we observe that while the overall model appears to fit well according to its RMSEA, the structural part actually fits poorly as can be seen from the RMSEA-P of 0.17. Inspecting each implication points to the same two violated constraints that we identified in our path model analysis. All fit indices agree that the lacking conditional covariance between M_2 and M_3 is a more severe problem than the omission of the relevant variable U_1. Interestingly, using the cutoff value of .99 for C9 or the cutoff value of .01 for C10, as Lance et al. (2016) suggested, properly separates the wrong from the correct conditional independence implications in this example.

In summary, we have given suggestions how local fit evaluation could be incorporated into state-of-the-art SEM analyses. The key idea behind our suggestions is the fact that implications can be tested individually by constructing specific single-implication models, and comparing their fits to alternative models. This leads to natural local equivalents of a wide variety of fit indices, including but not limited to all indices in the recent taxonomy by Lance et al. (2016).

Discussion

In this paper we explained the underlying logic of local fit evaluation, and showed how these tests can be derived using simple graphical criteria. There are three aspects of local tests that are potentially useful: We can enumerate them directly after designing our model, and directly judge whether current theoretical knowledge supports the implied constraint (though this is likely only feasible for d-separation constraints). We can perform the tests before fitting a model to data, which helps to test models that do not converge, and we can apply them after a model fit indicated a significant misfit to potentially pinpoint where the exact problem is. Manifest and latent variables yield different types of local tests, but both share the fact that they can be enumerated before data has been collected, and they can be parametrically tested before or after a model has been fitted.

Failure to converge can occur in practice for reasons in-

Table 2
Local fit evaluation of a latent variable SEM using single-implication path models as shown in Figure 8. Note that C9 and C10 are fit indices that only apply to the path model or parts of it, and are therefore not defined for the whole model.

	χ^2	df	p	RMSEA	C9(χ^2/df)	C10(χ^2/df)
Total	132	84	7.11^{-04}	.034	-	-
Structural	62.6	4	8.44^{-13}	.171	.896	.104
$r_{M_1M_2.X} = 0$	0.37	1	.543	.000	.999	.001
$r_{M_1M_3.X} = 0$	0.04	1	.838	.000	1.000	.000
$r_{M_2M_3.X} = 0$	53	1	3.39^{-13}	.323	.912	.088
$r_{XY.M_1M_2M_3} = 0$	12.2	1	4.71^{-04}	.150	.980	.020

cluding identification problems (global or local), numerical issues with optimization algorithms, small sample sizes, and specification errors. Faced with such problems, researchers may be tempted to achieve convergence by adjusting the model because the SEM toolbox currently does not offer many other options. Local tests can help in such situations to find problems with the structural part of the initial model. The reason that local tests work in such cases is that they are based on very basic methodology: they are simple statistical tests of population parameters that can always be computed and, unlike maximum likelihood methods such as the χ^2 test, do not require any numerical optimization algorithms. Thus, a major advantage of local tests is that they are unaffected by convergence issues.

Most researchers agree that model misfit, as indicated by a significant χ^2 or other fit measures, is a serious problem that should be investigated carefully. Faced with a non-fitting model, researchers could report various fit indices to argue that the problem is of minor importance, inspect whether measurement portions or structural portions of the model are responsible for the misfit, or in an extreme case simply discard the model outright. Neither option is completely satisfying. Even if adjunct fit indices indicate reasonable fit, this still does not help to understand which parts of the model caused the misfit. On the other hand, a model can still be useful or largely correct despite a minor misfit, and outright rejection of the whole model does not always appear warranted. For models that fail the χ^2 test, but produce reasonable fit indices, one should nevertheless investigate and report the results of local tests to help the reader understand what the exact reasons are for the misfit, or in other words, which of the predictions of the model are least consistent with the data. Paired with examination of the absolute magnitude of the tests and tests of close fit, the severity of the violations can be assessed and communicated.

Modification indices are related to local tests: Every locally testable implication of a model is derived from missing paths in the model, and thus a failing test can always be repaired by adding a path to the model (although this may influence the other tests and create new failed tests). Thus, a local test (of a converging model) indicates significant misfit if and only if one (or several) modification indices show significant improvement of fit. Why should we then apply local tests instead of familiar modification indices to converging models? As our examples show, modification indices can be misleading if the model fails to fit for a reason other than a missing or wrong path (such as non-normality or lack of representation of measurement error), and may prevent the researcher from thinking about such other reasons at all. Since it is always possible to improve model fit by adding paths, the researcher may end up with an incorrect model if the reason for misfit was not a missing path in the first place. Thus, conceptually, local tests differ from modification indices in that they force the researcher to think about possible reasons for misfit. In this aspect, local tests are more similar to examination of standardized residuals between model-implied and sample covariances, another means of diagnosing model misfit.

The number of conditional independence constraints in a path model is moderate – it equals the number of missing paths, or in other words, the degrees of freedom (for identified models). However, the number of tetrad constraints for medium-sized or large latent variable models is huge, and can appear daunting. An issue that arises in this context is the multiple testing problem. As is well known, conducting multiple significance tests carries with it an increased risk of Type I errors. With a large number of local tests, false positives can become more frequent, and some Type I error adjustment appears to be warranted. Stringent adjustment however also influences the statistical power of tests. In addition, the statistical power of the local tests may vary widely within a single model. A test of a violation of large magnitude may have very high power, while the same model may have a violation of smaller magnitude, and subsequently a test that is under-powered. It may be helpful to consider a-priori what magnitude of a violation can be reliably detected with sufficient power, given an assumed sample size, and significance level. A complicating matter in the context of local tests is that many of the tests are not independent, and therefore p-value adjustment techniques should be used that do not rely on the tests being independent. Independent subsets of local tests can be derived under the hypothesis that the postulated model is correct, which has been done for both conditional independencies (Shipley, 2000) and for tetrads

(Bollen & Ting, 1993). However, this approach defies the purpose of local fit evaluation as presented here, because we wish to test the individual implications of the model separately rather than the model as a whole. In that regard, the approach presented here is closer in spirit to the tetrad-based Bayesian posterior predictive checks discussed by Johnson and Bodner (2014), who also consider all tetrads rather than an independent subset. To help researchers navigate large numbers of tetrad tests, we suggest to separately investigate the three levels of Kenny's tetrad typology (Kenny, 1979), which is also supported by the 'dagitty' package. This may allow for easier interpretation of the results.

In summary, we argue that local tests are a valuable *addition* to the applied researcher's toolbox. They are not meant to replace global tests, and in fact, this paper does not argue, nor provides evidence in favor of, abandoning global tests. Such an argument, if it were even sensible, would require large scale simulation studies, and analytic derivations. We consider local fit evaluation as a supplement that can foster a more thorough account of model fit. We do believe that local fit evaluation can provide helpful diagnostic information, especially when models fail to converge or fail to fit. To facilitate local fit evaluation in practice, we have implemented the methods discussed here in the R package 'dagitty', which is available for R on CRAN.

References

Anderson, J. C., & Gerbing, D. W. (1988). Structural equation modeling in practice: A review and recommended two-step approach. *Psychological Bulletin*, *103*(3), 411–423.

Bauldry, S., & Bollen, K. A. (2016). tetrad: A set of Stata commands for confirmatory tetrad analysis. *Structural Equation Modeling: A Multidisciplinary Journal*, *0*, 1–10.

Bentler, P. (1990). Comparative fit indexes in structural models. *Psychological Bulletin*, *107*(2), 238–246.

Bollen, K. A., & Ting, K.-f. (1993). Confirmatory tetrad analysis. *Sociological Methodology*, *23*, 147–175.

Bollen, K. A., & Ting, K.-f. (2000). A tetrad test for causal indicators. *Psychological Methods*, *5*, 3–22.

Hayduk, L., & Glaser, D. (2000). Jiving the four-step, waltzing around factor analysis, and other serious fun. *Structural Equation Modeling*, *7*, 1–35.

Hipp, J. R., Bauer, D. J., & Bollen, K. A. (2005). Conducting tetrad tests of model fit and contrasts of tetrad-nested models: a new SAS macro. *Structural Equation Modeling*, *12*, 76–93.

James, L. R., Mulaik, S. A., & Brett, J. M. (1982). *Causal analysis: Assumptions, models, and data*. Sage, Beverly Hills.

Johnson, T., & Bodner, T. (2014). Posterior predictive checks of tetrad subsets for covariance structures of measurement models. *Psychological Methods*, *18*, 494–513.

Jöreskog, K. G. (1969). A general approach to confirmatory maximum likelihood factor analysis. *Psychometrika*, *34*(2), 183–202.

Kenny, D. A. (1974). A test for a vanishing tetrad: The second canonical correlation equals zero. *Social Science Research*, *3*, 83–87.

Kenny, D. A. (1979). *Correlation and causality*. Wiley, New York.

Kruschke, J. (2010). *Doing Bayesian data analysis: A tutorial introduction with R*. Academic Press.

Lance, C. E., Beck, S. S., Fan, Y., & Carter, N. T. (2016). A taxonomy of path-related goodness-of-fit indices and recommended criterion values. *Psychological Methods*, *21*, 388–404.

MacCallum, R. (1986). Specification searches in covariance structure modeling. *Psychological Bulletin*, *100*, 107–120.

MacCallum, R., Roznowski, M., & Necowitz, L. (1992). Model modifications in covariance structure analysis: the problem of capitalization on chance. *Psychological Bulletin*, *111*, 490–504.

McDonald, R. P., & Ho, M. H. (2002). Principles and practice in reporting structural equation analyses. *Psychological Methods*, *7*, 64–82.

Mulaik, S., & Millsap, R. (2000). Doing the four-step right. *Structural Equation Modeling*, *7*, 36–73.

Nickerson, R. S. (2000). Null hypothesis significance testing: a review of an old and continuing controversy. *Psychological methods*, *5*, 241-301.

O'Boyle Jr, E. H., & Williams, L. J. (2011). Decomposing model fit: Measurement vs. theory in organizational research using latent variables. *Journal of Applied Psychology*, *96*, 1–12.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible reasoning*. Morgan Kaufmann Publishers, Los Altos.

Pearl, J. (1995). Causal diagrams for empirical research. *Biometrika*, *82*, 669-688.

Pearl, J. (2000). *Causality: models, reasoning, and inference*. Cambridge Univ Press: New York.

Rosseel, Y. (2012). lavaan: An R package for structural equation modeling. *Journal of Statistical Software*, *48*, 1–36.

Saris, W., Satorra, A., & Van der Veld, W. (2009). Testing structural equation models or detection of misspecifications? *Structural Equation Modeling*, *16*, 561–582.

Saris, W., & Stronkhorst, L. (1984). *Causal modelling in nonexperimental research: An introduction to the lisrel approach*. Sociometric Research Foundation.

Scheines, R., Spirtes, P., Glymour, C., Meek, C., & Richardson, T. (1998). The tetrad project: Constraint based aids to causal model specification. *Multivariate Behavioral Research*, *33*, 65-117.

Shipley, B. (2000). A new inferential test for path models based on directed acyclic graphs. *Structural Equation Modeling*, *7*, 206-218.

Shipley, B. (2002). *Cause and correlation in biology: a user's guide to path analysis, structural equations and causal inference*. Cambridge University Press.

Spearman, C. (1904). General intelligence, objectively determined and measured. *The American Journal of Psychology*, *15*, 201–292.

Spirtes, P., Glymour, C. N., & Scheines, R. (2000). *Causation, prediction, and search*. MIT press.

Steiger, J. (1990). Structural model evaluation and modification: An interval estimation approach. *Multivariate Behavioral Research*, *25*, 173–180.

Steiger, J. (2004). Beyond the F test: Effect size confidence intervals and tests of close fit in the analysis of variance and contrast analysis. *Psychological Methods*, *9*, 164-182.

Sullivant, S., Talaska, K., & & Draisma, J. (2010). Trek separation for Gaussian graphical models. *The Annals of Statistics*, *38*, 1665-1685.

Textor, J., Hardt, J., & Knüppel, S. (2011). DAGitty: A graphical tool for analyzing causal diagrams. *Epidemiology*, *22*, 745.

Textor, J., van der Zander, B., Gilthorpe, M., Liśkiewicz, M., & Ellison, G. T. (2017). Robust causal inference using directed acyclic graphs: The R package 'dagitty'. *International Journal of Epidemiology*. (advance access) doi: 10.1093/ije/dyw341

Thoemmes, F., & Mohan, K. (2015). Graphical representation of missing data problems. *Structural Equation Modeling: A Multidisciplinary Journal*, *22*, 631-642.

Thoemmes, F., & Rose, N. (2014). A cautious note on auxiliary variables that can increase bias in missing data problems. *Multivariate Behavioral Research*, *49*, 443–459.

Wagenmakers, E.-J. (2007). A practical solution to the pervasive problems of p values. *Psychonomic Bulletin & Review*, *14*, 779–804.

Wishart, J. (1928). Sampling errors in the theory of two factors. *British Journal of Psychology. General Section*, *19*, 180–187.

Appendix

Example R code to perform local fit evaluation

This code contains the source code for all examples discussed in the paper. Except for the first two source codes, all examples use the R packages 'dagitty' and ''lavaan'. At least version 0.2-2 of the package 'dagitty' is necessary (this is the current version at the time of writing).

Testing d-separation parametrically

```
# This example shows how to simulate data from a standardized structural
# equation model and test a d-separation constraint, without using any R
# packages

set.seed(1234)

# Sample size
N <- 1000

# Path coefficient for all paths
p <- 0.3

# The bi-directed arrow between a and b is replaced by a structure a <- L -> b
# for the simulation
L <- rnorm(N, 0, 1)

# We now generate data according to the model structure, setting the residual
# variance such that the expected variance of each variable is 1
A <- sqrt(p) * L + rnorm(N, 0, sqrt(1 - sqrt(p)^2))
B <- p * A + rnorm(N, 0, sqrt(1 - p^2))
C <- sqrt(p) * L + rnorm(N, 0, sqrt(1 - sqrt(p)^2))
D <- p * B + p * C + rnorm(N, 0, sqrt(1 - p^2 - p^2 - 2 * p^4))

# Now we test the implication Cov( B, C | A )=0. First, we regress both B and
# C on A, and compute the residuals
rB.A <- lm(B ~ A)$resid
rC.A <- lm(C ~ A)$resid

# Now, we test if there is any correlation between those residuals
cor.test(rB.A, rC.A)

#Then we perform a test of not-close fit, testing observed value against .05
z <- atanh(cor(rB.A, rC.A))
sigmafz <-  1/sqrt(length(A) - 3 -2)
pval <- pchisq((z/sigmafz)^2,1, ncp =  (atanh(.05)/sigmafz)^2,lower.tail = FALSE)

# Next we test the implication Cov( A, D | B, C )=0 in the same manner
rA.BC <- lm(A ~ B + C)$resid
rD.BC <- lm(D ~ B + C)$resid
cor.test(rA.BC, rD.BC)

#Then we perform a test of not-close fit, testing observed value against .05
z2 <- atanh(cor(rA.BC, rD.BC))
sigmafz2 <-  1/sqrt(length(A) - 3 -2)
pval2 <- pchisq((z2/sigmafz)^2,1, ncp =  (atanh(.05)/sigmafz)^2,lower.tail = FALSE)
```

Testing vanishing tetrads

```
# This example shows how to test tetrad constraints implied by a structural
# equation model using both a parametric test by Wishart and by bootstrapping
# standard errors. The code only shows how to test the first implied tetrad.
# We only use the standard R package 'boot' for bootstrapping

library(boot)
set.seed(1234)

# Sample size
N <- 1000

# Standardized coefficient for all paths
p <- 0.3

# Generate data according to two-factor latent variable model, where X,Y,Z are
# indicators of the first factor, and W is the single indicator of the second
# factor
U1 <- rnorm(N)
U2 <- p * U1 + rnorm(N, 0, sqrt(1 - p^2))
X <- p * U1 + rnorm(N, 0, sqrt(1 - p^2))
Y <- p * U1 + rnorm(N, 0, sqrt(1 - p^2))
Z <- p * U1 + rnorm(N, 0, sqrt(1 - p^2))
W <- p * U2 + rnorm(N, 0, sqrt(1 - p^2))

# Compute sample covariance matrix
d <- data.frame(X, Y, Z, W)
S <- cov(d)

# Print value of tetrad WYZX. Using the paper's notation, I={W,X} and J={Y,Z}
tetrad <- det(S[c("W", "X"), c("Y", "Z")])

# Determine standard error of tetrad WYZX using Wishart's formula
d.IJ <- det(S[c("W", "Y", "Z", "X"), c("W", "Y", "Z", "X")])
d.I <- det(S[c("W", "X"), c("W", "X")])
d.J <- det(S[c("Y", "Z"), c("Y", "Z")])

tetrad.se <- sqrt((d.I * d.J * (N + 1)/(N - 1) - d.IJ)/(N - 2))
(tetrad.pval <- 2*pnorm(abs(tetrad/tetrad.se), lower.tail = FALSE))

#test of close fit, here against arbitrary value of .05
tetrad.z <- tetrad / tetrad.se
tol.z <- .05 / tetrad.se
tetrad.pval.close <- pchisq( tetrad.z^2, 1, ncp=tol.z^2, lower.tail=FALSE )

# Bootstrap standard error of tetrad WYZX
boot.out <- boot(d, function(d, i) det(cov(d[i, ])[c("W", "X"), c("Y", "Z")]),
    R = 1000)
print(boot.ci(boot.out, type = "norm"))

#Remaining tetrads are computed via dagitty
ex2model <- dagitty("dag␣{
```

```
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣U1␣[latent]␣;␣U2␣[latent]
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣{␣X␣Y␣Z␣}␣<-␣U1␣->␣U2␣->␣{W}
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣}")

#Wishart's test
(localTests(ex2model, data.frame(X,Y,Z,W), "tetrads"))
#Bootstrapped
(localTests(ex2model, data.frame(X,Y,Z,W), "tetrads",R=1000))
#close fit
(localTests(ex2model, data.frame(X,Y,Z,W), "tetrads",tol=.05))
```

Partial mediation model

This code relates to Figure 3.

```
# Load required packages.
library(lavaan)
library(dagitty)
# At least version 0.2.2 of the dagitty package is required for this and all
# following examples.
if(packageVersion("dagitty") < "0.2.2") {
    stop("Please␣update␣the␣dagitty␣package␣to␣run␣this␣example!")
}
set.seed(1234)
N <- 1000

# Below we define a graphical model in dagitty. Models are defined using
# arrow operators like ->, <-, or <->. Variables can be grouped using curly
# braces to shorten notation
g.true <- dagitty("dag␣{
U1␣[latent]
X␣->␣{␣U1␣M2␣<->␣M3␣}␣->␣Y
U1␣->␣M1
}")

# A simple graph can be generated automatically after X and Y coordinates
# for each variable have been supplied
coordinates(g.true) <- list(x=c(X=0,U1=1,M1=2,M2=2,M3=2,Y=3),
                            y=c(X=0,U1=-1,M1=-1,M2=0,M3=1,Y=0))
plot(g.true)

# Generate data according to the true model. All standardized path
# coefficients are set to 0.9
d <- simulateSEM(g.true, 0.4, N = N)

# Enumerate all d-separation constraints implied by the true model
print(impliedConditionalIndependencies(g.true))

# Define a second graphical model that is slightly misspecified
g.assumed <- dagitty("dag␣{␣X␣->␣{␣M1␣M2␣M3␣}␣->␣Y␣}")

#Again a simple graph
coordinates(g.assumed) <- list(x=c(X=0,M1=2,M2=2,M3=2,Y=3),
                            y=c(X=0,M1=-1,M2=0,M3=1,Y=0))
plot(g.assumed)
```

```
# Enumerate all d-separation constraints implied by the assumed model
print(impliedConditionalIndependencies(g.assumed))

# Perform statistical tests of conditional independence and report results
print(localTests(g.assumed, d, "cis",tol=0))

#Also perform tests of close fit using .05 as tolerage
print(localTests(g.assumed, d, "cis",tol=.05))


# Convert the model to lavaan syntax
mymodel <- toString(g.assumed, "lavaan")

# Estimate the model in lavaan and request fit statistics and standardized
# residuals
fit <- sem(mymodel, d)
print(summary(fit, fit = TRUE, modindices = TRUE))
resid(fit,type="standardized")


# Add a covariance between M2 and M3, as suggested by the modification index
g.assumed.2 <- dagitty("dag␣{␣X␣->␣{␣M1␣M2␣<->␣M3␣}␣->␣Y␣}")
mymodel.2 <- toString(g.assumed.2, "lavaan")
fit.2 <- sem(mymodel.2, d)
print(summary(fit.2, fit = TRUE, modindices = TRUE))
resid(fit.2,type="standardized")
```

Model with latent variables

This code relates to Figure 4.

```
library(lavaan)
library(dagitty)
set.seed(1234)
N <- 1000

# Define the model in dagitty syntax. Variable and arrow attributes can be
# set in square brackets. We use this below to define which variables are
# latent and we also define some path coefficients that we will use to
# generate data
g.true <- dagitty("dag␣{
LX␣[latent]
LY␣[latent]
LX␣->␣LY
LX␣->␣{␣X1␣X2␣X3␣}
LY␣->␣{␣Y1␣Y2␣Y3␣}
X1␣<->␣X3␣[beta=.25]
X1␣<->␣Y1␣[beta=.25]
}")

#A simple graph of the model
coordinates(g.true) <- list(x=c(LX=1,LY=5,X1=0,X2=1,X3=2,Y1=4,Y2=5,Y3=6),
```

```
                                y=c(LX=-.5,LY=-.5,X1=0,X2=0,X3=0,Y1=0,Y2=0,Y3=0))
plot(g.true)

# Generate data, using .7 for all path coefficients not set in the syntax
d <- simulateSEM(g.true, 0.7, N = N)

# Lists all tetrad implications
vanishingTetrads(g.true)

# We assume a model that has more constraints (fewer arrows)
g.assumed <- dagitty("dag␣{
LX␣[latent]␣;␣LY␣[latent]
{␣X1␣X2␣X3␣}␣<-␣LX␣->␣LY␣->␣{Y1␣Y2␣Y3}
}")

#A simple graph of the model
coordinates(g.assumed) <- list(x=c(LX=1,LY=5,X1=0,X2=1,X3=2,Y1=4,Y2=5,Y3=6),
                               y=c(LX=-.5,LY=-.5,X1=0,X2=0,X3=0,Y1=0,Y2=0,Y3=0))
plot(g.assumed)


# List all vanishing tetrads implied by the assumed model
print(vanishingTetrads(g.assumed))

# Convert the assumed model to lavaan syntax
m.assumed <- toString(g.assumed, "lavaan")

# Unrestricted model (EFA)
factanal(~X1+X2+X3+Y1+Y2+Y3,data=d,factors=2)

# Fit assumed model and request fit indices
fit <- sem(m.assumed, d, std.lv = TRUE)
print(summary(fit, fit = TRUE, mod = TRUE))
resid(fit,"standardized")

# Execute all tetrad tests and return p-values and confidence intervals
print(localTests(g.assumed, d, "tetrads"))

# Test of close fit for tetrads
print(localTests(g.assumed, d, "tetrads",tol=.05))

# Fit model in which largest modificiation index was added
# (i.e., the covariance between X1 and Y1)
m.assumed2 <- paste(m.assumed,"\n","X1~~Y1")
fit2 <- sem(m.assumed2, d, std.lv = TRUE)
print(summary(fit2, fit = TRUE, mod = TRUE))
resid(fit2,"standardized")
```

Non-converging path model

This code relates to Figure 5.

```
library(lavaan)
library(dagitty)
```

```
set.seed(123)
N <- 1000

# True model
g.true <- dagitty("dag{
J␣->␣I␣[beta=-.8]
I␣->␣X␣[beta=.4]
X␣<->␣M␣[beta=.25]
J␣->␣X␣->␣M␣->␣Y
J␣->␣Y
}")
d <- simulateSEM(g.true, 0.5, N = N)

#A simple graph of the model
coordinates(g.true) <- list(x=c(I=1,J=0,X=2,M=3,Y=2),
                            y=c(I=1,J=0,X=0,M=0,Y=-1))
plot(g.true)


# Assumed model; lacks the arrow I -> X
g.assumed <- dagitty("dag{␣J␣->␣Y␣;␣J␣->␣I␣->␣X␣->␣M␣->␣Y␣;␣X␣<->␣M␣}")
m.assumed <- toString(g.assumed, "lavaan")

#A simple graph of the model
coordinates(g.assumed) <- list(x=c(I=1,J=0,X=2,M=3,Y=2),
                            y=c(I=1,J=0,X=0,M=0,Y=-1))
plot(g.assumed)



# The model fails to converge. Hence we cannot list modification indices
fit <- sem(m.assumed, d)
print(summary(fit))

# The most significant local test suggests adding an arrow I -> X to the
# model (an arrow X -> I would yield a cyclic model)
print(localTests(g.assumed, d,tol=0))
print(localTests(g.assumed, d,tol=.05))

# Adding this arrow gives the true model, which passes local tests ...
print(localTests(g.true, d))

# ... and also also converges and fits globally
m.true <- toString(g.true, "lavaan")
fit.true <- sem(m.true, d)
print(summary(fit.true))
```

Non-converging single-factor model with parameter constraint

This code relates to Figure 6.

```
library(lavaan)
library(dagitty)
set.seed(123)
```

```
N <- 100

# We define a single-factor model with residual correlations between the
# first two and the last two indicators
g.true <- dagitty("dag{
L␣[latent]
L␣->␣{␣X1␣<->␣X2␣[beta=.25]␣}
L␣->␣{␣X3␣<->␣X4␣[beta=.25]␣}
L␣->␣X3␣[beta=.1]
}")

#A simple graph of the model
coordinates(g.true) <- list(x=c(L=0,X1=-1,X2=-.5,X3=.5,X4=1),
                            y=c(L=0,X1=1,X2=1,X3=1,X4=1))
plot(g.true)

# Simulate data with all unspecified loadings set to .8
d <- simulateSEM(g.true, 0.8, N = N)

# We estimate the model by imposing an equality constraint between the
# loadings of X1 and X3. Without such a constraint, the model would not be
# identified
m.assumed <- sem("u=~l*X1+X2+l*X3+X4
X1~~X2
X3~~X4", d)

# The model does not converge. That could be due to the model structure or
# the equality constraint
print(summary(m.assumed, fit = TRUE))

# We can use local tests to test the structure in isolation. This shows that
# the structure is OK
print(localTests(g.true, d, type = "tetrads"))
print(localTests(g.true, d, type = "tetrads",tol=.05))

# So the problem must be the equality constraint. We try a different one:
m.assumed.2 <- sem("u=~l*X1+X2+X3+l*X4
X1~~X2
X3~~X4", d)

# This model does fit
print(summary(m.assumed.2, fit = TRUE))
```

Model with non-normal data

This code relates to Figure 7.

```
library(lavaan)
library(dagitty)
set.seed(123)
N <- 1000

# We generate non-linear data using an 'additive noise model'. The noise is
# still Gaussian, but the variables are non-linearly dependent
```

```
X <- 4 + rnorm(N)
Y <- X^2 + 0.2 * sd(X^2) * rnorm(N)
Z <- Y^2 + 0.2 * sd(Y^2) * rnorm(N)

# Scale data to variance 1 for numerical reasons
d <- as.data.frame(scale(cbind(X, Y, Z)))

# A standard full mediation model does not fit to this data. Modification
# indices suggest adding a direct effect X -> Z
m <- sem("Z~Y\nY~X", d)
print(summary(m, fit = TRUE, mod = TRUE))


# The (linear) local test also fails
print(localTests("dag{X->Y->Z}", type = "cis", data = d))

# However, the semi-parametric local test indicates that conditional
# independence does hold (the confidence interval includes 0). Thus, the
# misfit is due to data distribution rather than model structure
print(localTests("dag{X->Y->Z}", type = "cis.loess", data = d, R = 500))

# Visualize linear regressions & residual correlations
par(mfrow = c(2, 3))
lmX.Y <- lm(X ~ Y, d)
lmZ.Y <- lm(Z ~ Y, d)
with(d, plot(Y, X))
abline(lmX.Y, col = 2)
with(d, plot(Y, Z))
abline(lmZ.Y, col = 2)
scatter.smooth(lmX.Y$resid, lmZ.Y$resid, span = 5, lpars = list(col = 2))

# Visualize non-linear smoothing using loess
lsX.Y <- loess(X ~ Y, d)
lsZ.Y <- loess(Z ~ Y, d)
with(d, scatter.smooth(Y, X, lpars = list(col = 2)))
with(d, scatter.smooth(Y, Z, lpars = list(col = 2)))
scatter.smooth(lsX.Y$resid, lsZ.Y$resid, span = 5, lpars = list(col = 2))
```

Local versions of fit indices

This code relates to Figure 2.

```
# This script performs local test evaluation of latent variable models by
# testing single-implication path models.
#
# The script was added in the revised version of the accompanying article
# and it depends on the newest version of dagitty, available from github.

library(dagitty)
library(lavaan)
set.seed(1234)

if(packageVersion("dagitty") < "0.2.3") {
    stop("Please␣update␣the␣dagitty␣package␣to␣run␣this␣example␣by␣running
```

```
␣␣␣␣␣␣devtools::install_github('jtextor/dagitty/r')")
}

# This function takes a SEM g with latent variables, and returns another
# SEM in which the structural model is saturated.
saturateStructure <- function(g){
  g <- as.dagitty(g)
  gm <- measurementPart(g)
  gs <- completeDAG(latents(g))
  c(gs,gm)
}

# This function takes a conditional independence implication x (in dagitty
# format) and a vector of variable names v, and returns a graph in which the
# only implied d-separation implication is x.
singleImplicationGraph <- function( x, v ){
  upper.part <- x$Z
  lower.part <- setdiff(v,c(x$X,x$Y,x$Z))
  r.edges <- ""
  for( i in upper.part ){
    r.edges <- paste(r.edges,i,"->",x$X,i,"->",x$Y)
    for( j in lower.part ){
      r.edges <- paste(r.edges,i,"->",j)
    }
  }
  for( i in lower.part ){
    r.edges <- paste(r.edges,i,"<->",x$X,i,"<->",x$Y)
  }
  if( length(upper.part)>1 ){
    ux <- combn(upper.part,2)
    r.edges <- paste(r.edges,paste(ux[1,],"<->",ux[2,]),collapse="\n")
  }
  if( length(lower.part)>1 ){
    ux <- combn(lower.part,2)
    r.edges <- paste(r.edges,paste(ux[1,],"<->",ux[2,]),collapse="\n")
  }
  as.dagitty(
    paste("dag{",r.edges,"}")
  )
}

# This function takes a SEM g (in dagitty syntax) and a data frame d, and evaluates
# - the overall fit of g to d
# - the fit of the structural model only, given the measurement model
# - the fit of each individual implication of the structural model, given the
#   measurement model
evaluatePathModel <- function( g, d ){
  g <- as.dagitty(g)
  N <- nrow(d)
  m <- toString(g,"lavaan")
  m.SS <- toString(saturateStructure(g),"lavaan")
  m.SN <- toString(measurementPart(g),"lavaan")
  tst.m <- attributes(lavaan( m, d, auto.var=T, std.lv=T ))$test[[1]]
```

```
  tst.SS <- attributes(lavaan( m.SS, d, auto.var=T, std.lv=T ))$test[[1]]
  tst.SN <- attributes(lavaan( m.SN, d, auto.var=T, std.lv=T ))$test[[1]]
  r <- as.data.frame(rbind(
    (c(stat=(tst.m$stat),df=(tst.m$df))),
    (c(stat=(tst.m$stat - tst.SS$stat),df=(tst.m$df-tst.SS$df)))
  ))
  rownames(r) <- c("Total","Structural")

  gs <- structuralPart(tested.model)
  gm <- measurementPart(tested.model)

  latents(gs)<-list()
  dseps <- impliedConditionalIndependencies(gs)
  for( i in seq_along(dseps) ){
    g.i <- c(singleImplicationGraph(dseps[[i]],names(gs)),gm)
    m.imp <- toString(g.i,"lavaan")
    tst.imp <- attributes(lavaan( m.imp, d, auto.var=T, std.lv=T ))$test[[1]]
    r.i <- data.frame(stat=(tst.imp$stat-tst.SS$stat),df=1)
    rownames(r.i) <- dseps[[i]]
    r <- rbind(r,r.i)
  }
  r$p.value <- apply( r, 1,
    function(x) pchisq(x[1],x[2],lower.tail=FALSE) )
  r$RMSEA <- apply( r, 1,
    function(x) sqrt(max(0,(x[1]/x[2]-1)/(N-1))) )
  r$C9 <- apply( r, 1,
    function(x) (tst.SN$stat-x[1]-tst.SS$stat)/(tst.SN$stat-tst.SS$stat) )
  r$C10 <- apply( r, 1,
    function(x) (x[1])/(tst.SN$stat-tst.SS$stat) )
  # C9 and C10 are path-based indices and as such undefined for the whole model.
  r[1,c("C9","C10")] <- NA
  r
}

# This graph is the same as in our previous partial mediation example.
# All variables are now latent variables, measured by 3 indicators each.
real.model <- dagitty("dag{
X[u]Y[u]M1[u]M2[u]M3[u]U1[u]
X␣->␣U1␣[beta=.4]
X␣->␣M2␣[beta=.4]
X␣->␣M3␣[beta=.4]
U1␣->␣Y␣[beta=.4]
U1␣->␣M1␣[beta=.4]
M2␣->␣Y␣[beta=.4]
M3␣->␣Y␣[beta=.4]
M2␣<->␣M3␣[beta=.4]

X␣->␣{x1␣x2␣x3}
Y␣->␣{x4␣x5␣x6}
M1␣->␣{x7␣x8␣x9}
M2␣->␣{x10␣x11␣x12}
M3␣->␣{x13␣x14␣x15}
}")
```

```
plot( graphLayout( real.model ) )

# The tested model has a correct measurement part, but an incorrect
# structural part.
tested.model <- c(
  measurementPart( real.model ),
  "dag{X[u]Y[u]M1[u]M2[u]M3[u]
␣␣X␣->␣{␣M1␣M2␣M3␣}␣->␣Y}"
)

# Run the analysis and print the results.
results <- evaluatePathModel(tested.model,simulateSEM(real.model,.8,.8))
print(signif(results,3))
```