

**Deep learning generalized structured component analysis: An interpretable artificial
neural network model with composite indexes**

Gyeongcheol Cho and Heungsun Hwang

McGill University, Montreal, Canada

Author Note

This is an Accepted Manuscript of an article published by Taylor & Francis in *Structural Equation Modeling: A Multidisciplinary Journal* on August 25, 2023. The Version of Record is available at <https://doi.org/10.1080/10705511.2023.2234086>.

Correspondence concerning this article should be addressed to Heungsun Hwang,
Department of Psychology, McGill University, 2001 McGill College Avenue, Montreal, QC
H3A 1G1, Canada. Email: heungsun.hwang@mcgill.ca

Deep learning generalized structured component analysis: An interpretable artificial neural network model with composite indexes

Abstract

Generalized structured component analysis (GSCA) is a multivariate method for specifying and examining interrelationships between observed variables and components. Despite its data-analytic flexibility honed over the decade, GSCA always defines every component as a linear function of observed variables, which can be less optimal when observed variables for a component are nonlinearly related, often reducing the component's predictive power. To address this issue, we combine deep learning and GSCA into a single framework to allow a component to be a nonlinear function of observed variables without specifying the exact functional form in advance. This new method, termed deep learning generalized structured component analysis (DL-GSCA), aims to maximize the predictive power of components while their directed or undirected network remains interpretable. Our real and simulated data analyses show that DL-GSCA produces components with greater predictive power than those from GSCA in the presence of nonlinear associations between observed variables per component.

Keywords: Generalized structured component analysis, deep learning, nonlinear component, composite index

1. Introduction

Generalized structured component analysis (GSCA; Hwang & Takane, 2004, 2014) is a multivariate method for examining path-analytic relationships between observed variables and components. GSCA estimates parameters via an iterative least squares algorithm, generating components that minimize the sum of (in-sample) prediction errors for all dependent variables in the model that researchers specify based on prior knowledge and theory. In this regard, GSCA's component may be considered an aggregated measure of observed variables, or a *composite index* created under a certain rule that a component is to explain the total variance of its observed variables and to be highly related to other components (Cho, Sarstedt, et al., 2022).

GSCA can include a broad array of component analysis methods as special cases, including (constrained) principal component analysis (Takane et al., 1995), (generalized) canonical correlation analysis (e.g., Carroll, 1968; Kettenring, 1971; Tenenhaus et al., 2017), principal covariate regression (de Jong & Kiers, 1992), redundancy analysis (van den Wollenberg, 1977), extended redundancy analysis (Takane & Hwang, 2005), canonical regression analysis (van der Leeden, 1990, p. 47), hierarchical structural component analysis (e.g., Choi et al., 2020), and partial or global least squares path modeling (Hwang et al., 2020; Hwang & Cho, 2020), particularly under the unidimensionality assumption that one component is extracted from each set of observed variables (refer to Hwang & Takane, 2014, Chapter 2).

GSCA has been extended in various ways to deal with more complex data structures and analyses. For instance, GSCA was combined with fuzzy clustering to capture cluster-level heterogeneity in observations (Hwang et al., 2007). It was also extended to accommodate interactions between components while addressing potential multicollinearity via regularization (e.g., Hwang, 2009; Hwang et al., 2010, 2021). GSCA and all these extensions to date commonly

assume that each component is defined as a *linear* function (or a weighted sum) of its observed variables, also referred to as (composite) indicators (Bollen & Bauldry, 2011). This linearity assumption can be useful for the interpretation of individual component scores, particularly when all component weights for indicators are positive. For example, if an individual has a high score on a component relative to other individuals, the individual will also tend to have relatively high scores for the component's indicators (e.g., refer to Cho, 2023, Section 3.3).

However, defining a component as a linear function of indicators (i.e., a linear component) all the time can be rather too restrictive in practice, ignoring potential nonlinear associations between the indicators. For example, two scatterplots in Figure 1 exhibit the relationships among four indicators for the Human Development Index (HDI; UNDP, 1990), including countries' gross national income (GNI), life expectancy (LifeExp), expected schooling years (ExpSchl), and mean schooling years (MeanSchl), from the Human Development Report 2010 (UNDP, 2010). The scatterplots show clear curvilinear relationships between GNI and the other indicators. Likewise, it has been reported that two indicators for socioeconomic status (SES), i.e., schooling year and income, are nonlinearly related (e.g., Gensowski et al., 2011, pp. 13–15; P. V. Le, 2014; Park, 1994). In these cases, adopting a linear component for the HDI or SES indicators can lead to less explanatory power for in-sample data.

=== Insert Figure 1 about here ===

In addition, linear components may have less predictive power for unseen observations or out-of-sample data. For example, Takane and Hwang (2007) investigated the relationships between two components—country-level food consumption and mortality rate. They showed that when both components were defined as linear ones, food consumption had substantially lower

predictive power for mortality rate, as compared to when they were defined as nonlinear functions of their indicators.

We may consider adding polynomial and/or product terms of indicators to capture some nonlinear relationships between the indicators. Nonetheless, this way of modeling requires prior knowledge on what forms of nonlinear relationships between indicators is expected and which terms should be added to capture the nonlinearity sufficiently well. Deep learning (Lecun et al., 2015; Rosenblatt, 1962) can be useful to address such nonlinearity of indicators without relying on prior knowledge on their relationships. As in GSCA, deep learning begins by defining multiple components as linear deterministic functions of observed variables. Unlike GSCA, however, it typically further transforms the linear components in a nonlinear manner. This multivariate and vector-valued function, which transforms multiple input variables (e.g., observed variables) into multiple outcome variables (e.g., transformed components of the input variables), is called a layer. The outcome variables of a layer are fed into another layer, in which they are again combined into several linear components and (nonlinearly) transformed. The operation of producing a function as a function of another function—function composition—is a key element that enables deep learning to successfully approximate nonlinear relationships between input and outcome variables (Urban & Gates, 2021). It is mathematically proven that any continuous function can be approximated by deep learning’s function in a prescribed accuracy under some mild conditions—the universal approximation theorem (For the conditions in detail, Csaji, 2001; Cybenko, 1989; Lu et al., 2017).

In the paper, therefore, we propose to combine deep learning (DL) with GSCA into a single framework. This extension, termed *deep learning generalized structured component analysis* (DL-GSCA), aims to conduct path-analytic modeling of indicators and components as

in GSCA. At the same time, DL-GSCA relaxes the linearity assumption between indicators and components thereof, allowing components to be nonlinear functions of their indicators without the need of pre-determining their exact functional forms. It then estimates the functional form of each component from the data by utilizing DL's artificial neural networks, such that the component can maximize its predictive power for dependent variables in the model. In this regard, DL-GSCA is distinct from GSCA that always fixes the component's functional form to be linear. Nonetheless, DL-GSCA still assumes that the relationships among components remain linear to facilitate their interpretation as in GSCA.

The paper is organized as follows. In Section 2, we provide a technical account of DL-GSCA, including its model specification, parameter estimation, and model evaluation. We here focus on providing a rather general description of the proposed method to avoid excessive notational burden, while all technical details with formal mathematical notations are relegated to appendices. In Section 3, we present the application of DL-GSCA to the Human Development Index datasets to illustrate its relative explanatory and predictive performance to GSCA. In Section 4, we conduct a Monte-Carlo simulation study to investigate whether DL-GSCA's comparative advantages over GSCA can be generalized to conditions where different forms of nonlinearity exist between indicators. In the last section, we summarize and discuss the implications, limitations, and potential extensions of the proposed method.

2. The Proposed Method

2.1. Model specification

We begin by describing GSCA's model specification to facilitate understanding of DL-GSCA's model specification. To specify path-analytic relationships between indicators and components,

GSCA involves three sub-models: weighted relation, component measurement, and structural models. The weighted relation model indicates that components are defined as linear deterministic functions of their respective indicators. The weights assigned to indicators to form their components are called (component) weights. The component measurement model shows that indicators are explained by their components and these relationships are signified by (component) loadings. The structural model specifies path-analytic linear associations between components, represented by path coefficients.

Let γ and $\mathbf{z}$ denote random vectors of components and indicators, respectively, both of which are assumed to be standardized. Let $\mathbf{W}$, $\mathbf{C}$, and $\mathbf{B}$ denote matrices consisting of weights, loadings, and path coefficients, respectively. Let ξ denote a vector of errors for $\mathbf{z}$ in the measurement model. Let ζ denote a vector of errors for γ in the structural model. Then, the three sub-models of GSCA are expressed as

$$\gamma = \mathbf{W}'\mathbf{z} \text{ (weighted relation model)} \quad (1)$$

$$\mathbf{z} = \mathbf{C}'\gamma + \xi \text{ (component measurement model)} \quad (2)$$

$$\gamma = \mathbf{B}'\gamma + \zeta \text{ (structural model),} \quad (3)$$

where $\mathbf{X}'$ denotes the transpose of a matrix $\mathbf{X}$.

DL-GSCA also involves the same sub-models. However, components in DL-GSCA are defined as any continuous (linear or nonlinear) deterministic functions of their indicators in the weighted relation model and are allowed to be nonlinearly related to their respective indicators in the component measurement model. Let f_W and f_C denote continuous functions of $\mathbf{z}$ and γ , respectively. Then, DL-GSCA's three sub-models are given as

$$\gamma = f_W(\mathbf{z}) \text{ (weighted relation model)} \quad (4)$$

$$\mathbf{z} = f_C(\gamma) + \xi \text{ (component measurement model)} \quad (5)$$

$$\gamma = \mathbf{B}'\gamma + \zeta \text{ (structural model).} \quad (6)$$

If f_W and f_C are confined to be linear, (4) and (5) become equivalent to (1) and (2), respectively, indicating that DL-GSCA includes GSCA as a special case. A more detailed description of DL-GSCA’s model specification is provided in Appendix A.

2.2. Parameter estimation

In DL-GSCA, we utilize DL for estimating f_W and f_C in a data-driven manner. Let h_W and h_C denote a set of DL’s artificial neural networks that approximate f_W and f_C , respectively. Two hyperparameters—the number of hidden layers and the number of hidden units per hidden layer **for each component**—determine the basic forms of h_W and h_C , as discussed in detail in Appendix B.

Given the matrix of standardized indicators and the two hyperparameters, we aim to estimate h_W , h_C , and $\mathbf{B}$ by minimizing a single optimization criterion that is equivalent to the average residual variance of all dependent variables in the model. We develop an alternating least squares (ALS) algorithm to minimize the optimization criterion iteratively. The ALS algorithm begins by dividing the model parameters into two sets and alternately updates each set of the parameters with the other set fixed until the criterion’s difference between consecutive iterations becomes smaller than a prescribed tolerance level (e.g., .0001). Once h_W , h_C , and $\mathbf{B}$ are estimated, the standard errors and confidence intervals of path coefficient estimates are obtained based on the bootstrap method (Efron, 1979, 1982). **We provide a detailed description of the ALS algorithm in Appendix C.**

Prior to the implementation of the ALS algorithm, it is important to predetermine the values of the two hyperparameters. These values affect the potential capacity of h_W and h_C to approximate f_W and f_C (Lu et al., 2017). For example, by increasing the number of hidden layers and/or that of hidden units per layer for h_W and h_C , one can approximate a more intricate form of

f_W and f_C , thereby reducing the bias in h_W and h_C . However, this approach can require estimating a large number of model parameters, which in turn tends to increase the variance in h_W and h_C . This phenomenon is often referred to as the bias-variance trade-off (Hastie et al., 2001, Chapter 7.3). If the increased variance outweighs the bias reduction, the model may overfit (e.g., Strang, 2019, p. 374) and fails to generate optimal components for predicting dependent variables in test samples. It is, therefore, essential to optimally calibrate the two hyperparameters of h_W and h_C to ensure the maximal predictive power of the components.

Nonetheless, it should be noted that there is, as yet, no universally agreed-upon method for determining hyperparameter values of deep learning methods (Mas & Flores, 2008). Within the context of DL-GSCA, we adopt the predictive feedforward search algorithm (Cho et al., 2022), which was originally proposed for variable selection in GSCA. This search algorithm gradually increases the hyperparameter values until any further increase results in a higher expected prediction error of the model. A detailed description of the search algorithm is presented in Appendix D.

Lastly, it is worth noting that DL-GSCA’s approach to estimating components can be categorized as unsupervised learning. This is due to its primary focus on “generating” components capable of accurately predicting their outcomes, including indicators and dependent components, without necessitating a pre-collected dataset of components. In fact, if researchers do not specify the structural model (6), the DL-GSCA model becomes identical to a set of autoencoders (e.g., Q. V. Le, 2015; Rumelhart et al., 1986), each of which takes a specific set of indicators as both inputs and outputs and includes a component as a hidden unit in its hidden layer (refer to Figures B1 and B3).

2.3. Model evaluation

DL-GSCA provides **three** types of indices for evaluating the overall performance of a specified model: FIT for dependent variables (FIT^D), test error for dependent variables (TE^D), and **out-of-bag prediction error for dependent variables (OPE^D)**. We provide the formulae of these overall model fit indices and relevant local model fit indices that we will also discuss below in Appendix E.

FIT^D is a rescaled version of the traditional goodness-of-fit measure FIT in GSCA (Hwang & Takane, 2014, pp. 26–30). It shows how much variance of dependent variables (both indicators and components) in the model is explained by components on average. This index is equivalent to the average of the R^2 values for all dependent variables, which ranges from 0 to 1. The larger the FIT^D value, the more variance is explained. Also, $1 - FIT^D$ is indicative of the average in-sample prediction error for dependent variables in the model. This in-sample prediction error indicates the model's estimated prediction error for the training sample. When every variable is specified as a dependent variable, FIT^D becomes equivalent to FIT. In addition, DL-GSCA provides two local goodness-of-fit indices: FIT_M^D and FIT_S^D . The former shows how much variance of dependent indicators in the component measurement model is explained by components on average (i.e., average R^2 for all dependent indicators), whereas the latter shows how much variance of dependent components in the structural model is explained by other components on average (i.e., average R^2 for all dependent components).

On the other hand, TE^D assesses the average *out-of-sample* prediction error for dependent variables in the model. The value of TE^D indicates the ratio of the average prediction errors of a specified model and the null model that uses the training-sample means of indicators and components as their predicted scores in a test sample. A value of TE^D smaller than 1 indicates

that the estimated model shows better predictive performance than the null model in the test sample. If the estimated model perfectly predicts dependent variables in the test sample, its TE^D value will be zero. DL-GSCA also provides TE_M^D and TE_S^D for respectively assessing the average out-of-sample prediction errors for dependent indicators in the component measurement model and for dependent components in the structural model, relative to the corresponding null models (i.e., in a test sample, the null component measurement model uses the training-sample means of indicators as the indicators' predicted scores and the null structural model uses the training-sample means of components as the components' predicted scores).

In practice, researchers may not have a designated test sample, or the sample size of their dataset may be too small. This can make it infeasible to set aside a portion of the dataset as a test sample and to calculate the TE^D value of the trained DL-GSCA model. To address this issue, DL-GSCA provides OPE^D , which is a rescaled version of the traditional cross-validation index OPE in GSCA (Cho et al., 2019). This index can estimate the expected prediction error of the DL-GSCA model, when a test sample is not available. The OPE^D value has the same interpretation as that of TE^D , but it is primarily used for comparing models (Cho et al., 2019). For example, when there are several competing models, the model with the smallest OPE^D value can be selected as the final one in terms of predictive generalizability. DL-GSCA also provides OPE_M^D and OPE_S^D , which aim to estimate the expected prediction error of the measurement and structural models, respectively.

When a test sample is given, researchers can additionally evaluate the predictive power of individual predictor components through the use of $\Delta TE_{p,q}$, which assesses the contribution of one predictor component, denoted by component p , to prediction of its dependent component, denoted by component q , in the model. If the value of $\Delta TE_{p,q}$ is negative, including the predictor

component in the model worsens the prediction of the dependent component. Conversely, if the value of $\Delta TE_{p,q}$ is positive, adding the predictor component contributes to better predicting the dependent component.

3. Empirical Application

The datasets we analyze come from two Human Development Reports (HDR) published by the United Nations Development Programme (UNDP, 2010, 2016) in 2010 and 2016. Under the premise that “the real wealth of nations” is people, not the real and financial assets accumulated (UNDP, 1990, p. 9), the UNDP created a composite index for measuring countries’ development levels for humans, named the Human Development Index (HDI), and has published each country’s HDI score and ranking annually since 1990. The HDI is evaluated in terms of three domains of human development—health, knowledge, and decent standard of living. Specifically, since the revision of the HDI formula in 2010, life expectancy at birth (LifeExp) has been used as the indicator for health, mean schooling years (MeanSchl) and expected schooling years (ExpSchl) as the indicators for knowledge, and gross national income per capital (GNI) as the indicator for decent standard of living.

Although the provision of the HDI plays a role in shifting a national paradigm of development from materialistic growth to human well-being in many countries (UNDP, 2016, p. 2), this index has been constantly criticized in two respects. First, many researchers point out that the formula used to calculate the HDI is rather arbitrary and difficult to justify because the formula has been heavily affected by the developers’ opinions (e.g., Noorbakhsh, 1998; Nübler, 1995). Second, researchers are concerned that the HDI is not developed to sufficiently account for other important domains of human development, including ecological sustainability (e.g.,

Sagar & Najam, 1998), political freedom (e.g., Ranis et al., 2006), and subjective well-being (e.g., Blanchflower & Oswald, 2005). In fact, the HDI has been developed based solely on a group of experts' judgements on which change/growth in the country should be indicative of development. However, it seems to be of particular importance to take into account subjective well-being, which informs "whether progress has indeed occurred if the (partial) metrics of human development suggest it appears to have" (Hall & Helliwell, 2014, p. 11), reflecting people's actual opinions about which changes in the HDI domains they perceive as real-life developments in their country (e.g., Blanchflower & Oswald, 2005). Thus, some researchers suggest determining the weights for the HDI indicators based on their contributions to subjective well-being (e.g., Blanchflower & Oswald, 2005; Nübler, 1995).

DL-GSCA may be used for *statistically* addressing these issues with the HDI. It does not need to pre-determine a formula for combining the four HDI indicators or equivalently how the weights for the indicators are to be determined. Instead, DL-GSCA estimates the functional form between the HDI and its indicators in a data-driven fashion such that the index maximizes the total explained variance of dependent variables in the DL-GSCA model. Particularly, subjective well-being can be specified as a focal dependent variable for the HDI in the model. Then, DL-GSCA will estimate the weights for the HDI indicators in such a way that the index is highly associated with subjective well-being.

Figure 2 displays the HDI model that we specified for illustrative purposes. We consider the HDI an exogenous component (i.e., $\gamma_1 = \text{HDI}$), which is associated with four indicators (i.e., $z_{1,1} = \text{LifeExp}$, $z_{1,2} = \text{MeanSchl}$, $z_{1,3} = \text{ExpSchl}$, and $z_{1,4} = \text{GNI}$). We also assume that the HDI influences two types of subjective well-being. One type is the average level of satisfaction with the country's environment for development (SED; $\gamma_2 = \text{SED}$), which is measured by three

indicators: $z_{2,1}$ = proportion of the people satisfied with healthcare quality (SatisHQ), $z_{2,2}$ = proportion of the people satisfied with education quality (SatisEQ), and $z_{2,3}$ = proportion of the people satisfied with state of living (SatisSL). The other type is overall life satisfaction (OLS; γ_3 = OLS), which is measured by a single indicator, $z_{3,1}$ = the average life satisfaction of the people living in a country (Cantril, 1965). We further assume that SED influences OLS, as it seems reasonable that an individual satisfied with his or her country's environment is more likely to be satisfied with his or her overall life (e.g., Silva et al., 2012).

=== Insert Figure 2 about here ===

We apply DL-GSCA to fit the model to the 2010 HDR data. Of 169 countries, we use 142 countries that have no missing observations ($N = 142$). **We wrote a MATLAB code for DL-GSCA¹ for this analysis.** We consider three levels of the number of hidden layers, denoted by L_p in Appendix B, for HDI and SED ($L_p = 0, 1$, and 2). When the number of hidden layers is greater than zero, we consider four levels of hidden units per hidden layer, denoted by $R_p^{(l)}$ in Appendix B ($R_p^{(l)} = 2, 3, 4$, and 5). We employ **Cho et al.'s (2022) search algorithm with** five-fold cross validation method for deciding the two hyperparameters. We then apply the ALS algorithm to each training sample, using ten different sets of initial values. The parameter estimates obtained from the set of initial values, which results in the largest FIT^D , are chosen as the final estimates. The tolerance level used in the study is .0001. For comparison, we also apply GSCA to the same data. The total number of bootstrap samples is 1000.

We choose one hidden layer for both HDI and SED and four and three hidden units per hidden layer for HDI and SED, respectively, based on the predictive feedforward search algorithm. We then proceed to apply the ALS algorithm with ten different sets of random initial

¹ The MATLAB code and the HDR data are available at https://osf.io/wdz79/?view_only=39cbbe18a7f442d381e2483b4bcd177

values. Figure 3 displays how the value of the optimization criterion (C.2) changes as the ALS algorithm iterates through its two steps with a set of randomly assigned initial values. With each iteration, the value of (C.2) always decreases, indicating that the ALS algorithm is successfully finding parameter values that result in a smaller value of (C.2) than the previous iteration.

=== Insert Figure 3 about here ===

DL-GSCA provides $FIT^D = .81$, indicating that the model explains 81% of the total variance of all dependent variables in the model. Also, DL-GSCA provides $FIT_M^D = .86$ and $FIT_S^D = .65$, indicating that the measurement model accounts for 86% of the total variance of all dependent indicators, whereas the structural model explains 65% of the total variance of all dependent components. Conversely, GSCA provides $FIT^D = .75$, $FIT_M^D = .79$, and $FIT_S^D = .60$. This indicates that DL-GSCA provides smaller in-sample prediction errors for both indicators and dependent components than GSCA.

Figure 4 displays the indicators' values predicted by DL-GSCA against their original values. The three scatterplots show that the HDI and SED components appear to capture the nonlinear relationships of their respective indicators sufficiently well, indicating that they can serve as a good summary of their indicators. The average R^2 value of the HDI and SED components for their respective indicators are .89 and .82, respectively.

=== Insert Figure 4 about here ===

Figure 5 depicts the relationships among the HDI and SED components and their indicators obtained from DL-GSCA. The individual scores of the components are obtained by plugging their indicators' scores in the estimated weighted relation model (i.e., $\mathbf{g}_{n,p} = h_{w,p}(\mathbf{d}_{n,p})$ in Eq. (C.1) in Appendix C). In general, an increase in the indicators' scores is associated with an increase in their component scores, indicating that an improvement in each domain of the HDI

and SED is likely to be related to an increase in the level of human development and people's satisfaction with their living environment for development. It seems intriguing that a small amount of increase in GNI is linked to a steep growth in the HDI for countries with extremely low levels of GNI, suggesting that an economic growth may contribute substantially to human development for these countries. On the other hand, for those with relatively high levels of GNI (i.e., greater than around a standardized score of 1.5 or equivalently 36,654 US dollars), an increase in GNI is associated with only a marginal increase in the HDI. This may be because GNI is negatively associated with the other domains of the HDI among the countries with high GNI levels, as displayed in the first two scatterplots of Figure 5.

=== Insert Figure 5 about here ===

In the structural model, the HDI has a positive and statistically significant effect on SED ($b_{1,2} = .76$, $SE = .04$, 95% CI = [.69, .83]), indicating that a higher level of human development tends to lead to a higher level of people's satisfaction with their living environment. Also, SED has a positive and statistically significant effect on OLS ($b_{2,3} = .45$, $SE = .06$, 95% CI = [.33, .58]), indicating that people satisfied with their living environment are likely to have a high level of overall life satisfaction. Moreover, the HDI has a positive and statistically significant effect and on OLS ($b_{1,3} = .44$, $SE = .07$, 95% CI = [.30, .56]), indicating that people living in a country with a relatively high level of human development tend to have a relatively high level of satisfaction with their life, controlling for satisfaction with their living environment. Furthermore, the HDI has a positive and statistically significant indirect effect on OLS, mediated by SED ($b_{1,2} \times b_{2,3} = .35$, $SE = .05$, 95% CI = [.25, .46]). The total effect of the HDI on OLS (.79) is also positive and statistically significant ($SE = .03$, 95% CI = [.73, .84]).

To evaluate the model's predictive power, we evaluate its out-of-sample prediction error, using the 2016 HDR dataset as a test sample. We use 156 countries without missing observations. DL-GSCA provides $TE^D = .24$, indicating that on average, DL-GSCA's prediction error is 24% of the null model's prediction error. This value is smaller than that of GSCA ($TE^D = .29$). DL-GSCA provides $TE_M^D = .17$ and $TE_S^D = .52$, indicating that DL-GSCA's prediction errors for the measurement and structural models are 17% and 52% of the respective null models. They are also smaller than the counterparts of GSCA ($TE_M^D = .23$ and $TE_S^D = .57$). These results show the superior predictive performance of DL-GSCA to GSCA. In addition, all $\Delta TE_{p,q}$ values obtained from DL-GSCA are positive (i.e., $\Delta TE_{1,2} = .98$, $\Delta TE_{2,3} = .17$, and $\Delta TE_{1,3} = .09$), indicating that all the predictor components contribute to predicting their dependent components.

For illustrative purposes, suppose that we do not have any additional dataset to use as a test sample. In such a case, as typically recommended in the literature (e.g., Hastie et al., 2001, p. 222), we might use about 75% of observations in the 2010 HDR dataset for model training and the rest for model testing. However, due to the limited number of countries in this dataset, we opt to use OPE^D , OPE_M^D , and OPE_S^D values to evaluate the out-of-sample errors of the competing DL-GSCA and GSCA models, rather than TE^D , TE_M^D , and TE_S^D values. DL-GSCA yields $OPE^D = .20$, whose value is smaller than that of GSCA ($OPE^D = .27$). This suggests that DL-GSCA can be chosen over GSCA in terms of predictive generalizability. Additionally, DL-GSCA provides $OPE_M^D = .15$ and $OPE_S^D = .36$, both of which are also smaller than GSCA's counterparts ($OPE_M^D = .23$ and $OPE_S^D = .43$). This indicates that DL-GSCA's nonlinear components are expected to have smaller prediction error for outcome variables in both measurement and structural models than GSCA's linear components.

Taken together, DL-GSCA is applied to construct a composite index for human development. Without assuming any prescribed formula for calculating the HDI, DL-GSCA estimates the formula in a data-driven manner, which leads the HDI to be constructed to sufficiently capture potential nonlinear associations among its four indicators while predicting two types of subjective well-being. In addition, DL-GSCA's components, including the HDI, have higher in-sample and out-of-sample prediction powers than GSCA's components.

4. Simulation Study

We conduct a simulation study to examine the explanatory and predictive power of DL-GSCA when nonlinear relationships exist between indicators in the model. For this study, we design a population DL-GSCA model that imitates the structural model as the HDI model, as shown in Figure 6. We describe how to generate data based on the population model in Appendix F.

=== Insert Figure 6 about here ===

We consider four levels of sample size ($N = 100, 200, 500$, and 1000), for each of which we generate 500 training samples. Additionally, we generate one test sample of $N = 2000$. Figure 7 displays the association patterns of each block of indicators in the test sample. In this simulation study, we employ 30% of each training sample as a validation sample to determine the values of the hyperparameters (i.e., L_p and $R_p^{(l)}$) from prescribed candidate values per hyperparameter (i.e., $L_p = 0, 1$, and 2 ; $R_p^{(l)} = 2, 3, 4$, and 5). We use the same number of initial values and tolerance level used in the previous section.

=== Insert Figure 7 about here ===

We compute the values of FIT^D , FIT_M^D , and FIT_S^D based on the final parameter estimates to evaluate the model's explanatory power (or in-sample prediction error) for each training

sample. We also calculate the TE^D , TE_M^D , and TE_S^D values based on the test sample to assess the model's predictive power (or out-of-sample prediction error). We apply traditional GSCA as a benchmark for comparing DL-GSCA's performance in both in-sample and out-of-sample prediction errors.

Figure 8 displays the average values of FIT^D , FIT_M^D , and FIT_S^D obtained from DL-GSCA and GSCA per sample size. As shown in the figure, DL-GSCA provides greater average FIT^D values than GSCA in all sample sizes, indicating that overall, the nonlinear components of DL-GSCA explain the variance of the dependent variables better than the linear components of GSCA. The difference in the average FIT^D values between the two methods remains unchanged across the sample sizes. The same patterns are observed in the average FIT_M^D and FIT_S^D values of the methods.

=== Insert Figure 8 about here ===

Figure 9 shows the average values of $1 - TE^D$, $1 - TE_M^D$, and $1 - TE_S^D$ from DL-GSCA and GSCA per sample size. DL-GSCA provides smaller average TE^D values than GSCA in all sample sizes, indicating that DL-GSCA generally outperforms GSCA in terms of predictive power. The TE^D values of both methods tend to decrease on average as the sample size increases, but the difference in the values between the methods does not diminish. The average TE_M^D and TE_S^D values of the methods show the same patterns.

=== Insert Figure 9 about here ===

To further compare the predictive performance of DL-GSCA and GSCA, we generate another training sample of $N = 2000$ and estimate their parameters from this sample. Then, we use the estimates to obtain the predicted values of the indicators in the test sample. Figure 10 displays the indicators' values predicted by DL-GSCA and GSCA against their original values in

the test sample. It clearly shows that DL-GSCA's components can produce relatively accurate predictions of the three blocks of indicators, efficiently capturing their nonlinear and linear relationships. On the other hand, GSCA's components fail to make accurate predictions of the first two blocks of indicators ($z_{1,1}$ to $z_{2,3}$), leading its indicators' predicted values to deviate substantially from their original values. This indicates that GSCA is not optimal for predicting the nonlinear associations among the indicators for components 1 and 2, as expected.

=== Insert Figure 10 about here ===

In sum, our simulation study shows that when some indicators are nonlinearly related, on average, DL-GSCA results in smaller in-sample and out-of-sample prediction errors than GSCA in all conditions, speaking to DL-GSCA's superior performance to GSCA in terms of both explanatory and predictive power.

5. Concluding Remarks

We proposed a new extension of GSCA, termed deep learning GSCA (DL-GSCA), which incorporates DL's functions to produce components that can capture nonlinear associations among their indicators in a data-driven manner without the need of pre-specifying the exact functional forms between the components and indicators. We also provided model evaluation indices to assess a model's explanation (in-sample) and prediction (out-of-sample) errors.

We conducted both real and simulated data analyses and demonstrated the superior performance of DL-GSCA to GSCA in the presence of nonlinear relationships among indicators per component. Moreover, we illustrated DL-GSCA's potential for creating the HDI while capturing nonlinear associations among the index's indicators and considering its hypothetical relationships with other variables. Unlike the traditional procedures for developing the HDI, in

which a group of experts determine its computational formula without considering its relationships with other variables, DL-GSCA statistically estimated the formula for the HDI such that it could be a good summary of its indicators and good predictors for two types of subjective well-being.

DL-GSCA contributes substantially to broadening the scope of GSCA beyond linear modeling and improving prediction accuracy. Nonetheless, it has limitations as well. For example, the proposed method in its current form assumes that components are always linearly related in the structural model. We here intend to focus on capturing nonlinear associations among indicators per component in the component measurement model while keeping the interpretation of the relationships among components as simple as possible. However, some researchers may want to specify nonlinear associations between components while still defining components as linear functions of their indicators (e.g., Basco et al., 2021; Hwang et al., 2010). For example, they may be interested in examining interaction effects of components, keeping the component scores easily interpretable (e.g., Hwang et al., 2021). Thus, it may be necessary to incorporate DL's functions into the structural model as well, so that researchers can freely decide which sub-model(s) can be nonlinear.

Moreover, DL-GSCA currently assumes that all dependent variables are continuous. Consequently, it is not suitable to estimate models with categorical dependent variables. For example, in genomic modeling, a set of single nucleotide polymorphisms (SNPs) within a gene is typically categorical variables (e.g., Romdhani et al., 2015). To handle such cases, we may need to extend DL-GSCA to adopt a different activation function for categorical variables (e.g., sigmoid or softmax; see Nwankpa et al., 2021), minimizing a mixture of two optimization

criteria, e.g., the mean squares error for continuous variables and cross-entropy for categorical variables.

Furthermore, DL-GSCA needs to accommodate various technical extensions of GSCA. For instance, it should be extended to deal with higher-order components (Hwang & Takane, 2014, pp. 99–110). In the empirical application, for example, the HDI may be considered a second-order component which is linked to the three domains of human development (i.e., health, knowledge, and decent standard of living) as its first-order components. Also, it would be beneficial to extend DL-GSCA to efficiently handle missing observations which frequently occur in practice (Graham, 2008). We may adopt GSCA’s model-based imputation approach (Hwang & Takane, 2014, pp. 123–125), which treats missing values as parameters and estimates them along with the other model parameters by minimizing a single optimization criterion. In addition, we may consider combining DL-GSCA with regularization techniques, such as ridge and lasso, to avoid potential overfitting. DL-GSCA may be more susceptible to the overfitting issue than GSCA because the former will likely have a far larger number of parameters than the latter. We may borrow the same idea of regularized GSCA (Hwang, 2009; Hwang & Takane, 2014, Chapters 8 and 9) to shrink DL-GSCA’s parameter estimates toward zero or to exact zero.

In closing, DL-GSCA represents a flexible, nonlinear multivariate method for estimating complex path-analytic models involving components. It aims to capture potential nonlinear associations between components and indicators while examining linear associations among components. Thus, DL-GSCA can substantially improve the predictive power of its components while maintaining the interpretability of the relationships among the components. We concentrate on proposing DL-GSCA’s general framework in the paper and will need to refine and extend the method in various ways to deal with a broad range of data-analytic issues,

including those enumerated above. Moreover, it will be important to develop a software program for DL-GSCA to make it more accessible to researchers and practitioners. For example, DL-GSCA can be included in GSCA’s free user-friendly software – GSCA Pro (Hwang et al., 2023). This can also contribute to the application of DL-GSCA to a greater variety of real-world problems and more thorough investigations of its practical usefulness.

Finally, we considered a maximum of two hidden layers per neural network (L_p) in our empirical analyses. This may give the impression that DL-GSCA can only handle a shallow neural network with a small number of hidden layers. However, in our DL-GSCA model, the overall structure is not represented merely by the layers of a singular nested neural network. As described in Appendix B, the model’s architecture consists of several interconnected neural networks. Therefore, the total number of hidden layers in the entire model is always significantly higher than the number presented in any single nested network (L_p). In addition, DL-GSCA does not impose any limit on the number of hidden layers (L_p) in each individual network. Researchers, depending on the availability of computational resources, can consider a greater value of L_p than the ones employed in our empirical studies.

Appendix A. Model specification in DL-GSCA

Let $\boldsymbol{\gamma} = [\gamma_1, \gamma_2, \dots, \gamma_P]'$ denote a P by 1 random vector of components, where γ_p is the p th component ($p = 1, 2, \dots, P$), and P is the total number of components. Let $\mathbf{z}_p = [z_{1,p}; z_{2,p}; \dots; z_{J_p,p}]$ denote a J_p by 1 random vector of indicators for γ_p , called a block of indicators for γ_p , where $z_{i,p}$ is the i th indicator in $\mathbf{z}_p$ ($i = 1, 2, \dots, J_p$) and J_p is the number of indicators for γ_p . Let $\mathbf{z} = [\mathbf{z}_1; \mathbf{z}_2; \dots; \mathbf{z}_P]$ denote a J by 1 random vector of indicators, where J is the total number of indicators.

Both indicators and components are assumed to be standardized, i.e., $E(z_{i,p}) = E(\gamma_p) = 0$ and $var(z_{i,p}) = var(\gamma_p) = 1$. Each component γ_p is defined as $\gamma_p = f_{w,p}(\mathbf{z}_p)$, where $f_{w,p}: \mathbb{R}^{J_p} \rightarrow \mathbb{R}$ denotes a continuous function of $\mathbf{z}_p$ to γ_p . Let $f_{c,p}: \mathbb{R} \rightarrow \mathbb{R}^{J_p}$ denote a continuous function of γ_p to $\hat{\mathbf{z}}_p$, where $\hat{\mathbf{z}}_p$ is the predicted value of $\mathbf{z}_p$ given γ_p . DL-GSCA assumes at default that γ_p is to explain the variance of $\mathbf{z}_p$. If $f_{w,p}$ and $f_{c,p}$ are linear, then $f_{w,p}(\mathbf{z}_p)$ and $f_{c,p}(\gamma_p)$ can be expressed as $f_{w,p}(\mathbf{z}_p) = \mathbf{w}_p' \mathbf{z}_p$ and $f_{c,p}(\gamma_p) = \mathbf{c}_p \gamma_p$ where $\mathbf{w}_p$ is a J_p by 1 vector of weight parameters for $\mathbf{z}_p$, and $\mathbf{c}_p$ is a J_p by 1 vector of weight parameters for $\mathbf{z}_p$. If γ_p is not assumed to explain $\mathbf{z}_p$, then $f_{c,p}(\gamma_p) = \mathbf{0}$. Let $\mathbf{b}_p = [b_{1,p}, b_{2,p}, \dots, b_{P,p}]'$ denote a P by 1 vector of path coefficients relating $\boldsymbol{\gamma}$ to γ_p , where $b_{q,p}$ is non-zero if γ_q is assumed to influence γ_p and zero otherwise ($q = 1, 2, \dots, P$ and $q \neq p$). Let $f_w(\mathbf{z}) \equiv [f_{w,1}(\mathbf{z}_1); f_{w,2}(\mathbf{z}_2); \dots; f_{w,P}(\mathbf{z}_P)]$, $f_c(\boldsymbol{\gamma}) \equiv [f_{c,1}(\gamma_1), f_{c,2}(\gamma_2), \dots, f_{c,P}(\gamma_P)]'$, and $\mathbf{B} \equiv [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_P]$. Let $\boldsymbol{\xi} = [\xi_1; \xi_2; \dots; \xi_P]$ denote a J by 1 vector of errors for $\mathbf{z}$ in the **component** measurement model, where ξ_p is a J_p by 1 vector of errors for $\mathbf{z}_p$. Let $\boldsymbol{\zeta} = [\zeta_1, \zeta_2, \dots, \zeta_P]'$ denote a P by 1 vector of errors for $\boldsymbol{\gamma}$ in the structural model, where ζ_p is an error for γ_p . Then, the three sub-models of DL-GSCA are expressed as (4), (5), and (6). Furthermore, DL-GSCA combines the sub-models into a single equation as follows.

$$\begin{aligned}
[\mathbf{z}; \boldsymbol{\gamma}] &= [f_c(\boldsymbol{\gamma}); \mathbf{B}'\boldsymbol{\gamma}] + [\boldsymbol{\xi}; \boldsymbol{\zeta}] \\
\leftrightarrow [\mathbf{z}; f_w(\mathbf{z})] &= f_A(\boldsymbol{\gamma}) + [\boldsymbol{\xi}; \boldsymbol{\zeta}] \\
\leftrightarrow f_v(\mathbf{z}) &= f_A(f_w(\mathbf{z})) + \boldsymbol{\varepsilon},
\end{aligned} \tag{A.1}$$

DEEP LEARNING GSCA

where $f_V(\mathbf{z}) = [\mathbf{z}; f_W(\mathbf{z})]$, $f_A(\gamma) = [f_C(\gamma); \mathbf{B}'\gamma]$, and $\boldsymbol{\varepsilon} = [\boldsymbol{\xi}; \boldsymbol{\zeta}]$. This is called the DL-GSCA model.

Appendix B. Approximations to DL-GSCA's f_W and f_C via deep learning's artificial neural networks

DL-GSCA utilizes deep learning (DL) for estimating each f_W and f_C in a data-driven manner. As f_W and f_C can be seen as function sets of $f_{w,p}$ and $f_{c,p}$ across components ($p = 1, 2, \dots, P$), respectively, DL-GSCA splits f_W and f_C into their small ingredients (i.e., $f_{w,p}$ and $f_{c,p}$) and processes each of them separately. Let $h_{w,p}$ and $h_{c,p}$ denote DL's two artificial neural networks that approximate $f_{w,p}$ and $f_{c,p}$, respectively. As briefly discussed earlier, these functions are built based on function composition, indicating that $h_{w,p}$ and $h_{c,p}$ map their input arguments to the output through a sequence of functions. Specifically, $h_{w,p}$ is defined as $\gamma_p = h_{w,p}(\mathbf{z}_p) = h_{w,p}^{(L_p+1)}(h_{w,p}^{(L_p)}(\dots(h_{w,p}^{(1)}(\mathbf{z}_p))))$, where $h_{w,p}^{(l)}$ is the l th function in a sequence of functions for $h_{w,p}$ ($l=1, 2, \dots, L_p+1$) and L_p+1 is the total number of functions in the sequence for γ_p . Likewise, $h_{c,p}(\gamma_p)$ is defined as $\hat{\mathbf{z}}_p = h_{c,p}(\gamma_p) = h_{c,p}^{(L_p+1)}(h_{c,p}^{(L_p)}(\dots(h_{c,p}^{(1)}(\gamma_p))))$, where $h_{c,p}^{(l)}$ is the l th function in a sequence of functions for $h_{c,p}$ ($l=1, 2, \dots, L_p+1$). As stated earlier, $h_{w,p}^{(l)}$ and $h_{c,p}^{(l)}$ are called layers in DL. Once $h_{w,p}$ and $h_{c,p}$ are defined, h_W and h_C are defined as $h_W(\mathbf{z}) = [h_{w,1}(\mathbf{z}_1); h_{w,2}(\mathbf{z}_2); \dots; h_{w,P}(\mathbf{z}_P)]$ and $h_C(\gamma) = [h_{c,1}(\gamma_1); h_{c,2}(\gamma_2); \dots; h_{c,P}(\gamma_P)]$, which approximate f_W and f_C , respectively. Taken together, DL-GSCA employs a total of $2P$ artificial neural networks to approximate f_W and f_C in (4) and (5).

There are three types of layers: input, hidden, and output layers. The input layer refers to a vector of input arguments, for instance, $\mathbf{z}_p$ for $h_{w,p}$ and γ_p for $h_{c,p}$, whereas the output layer refers to the last layer in a sequence of functions, such as, $h_{w,p}^{(L_p+1)}$ for $h_{w,p}$ and $h_{c,p}^{(L_p+1)}$ for $h_{c,p}$. The other functions between the input and output layers constitute hidden layers, for example, $h_{w,p}^{(L_p)}, h_{w,p}^{(L_p-1)}, \dots, h_{w,p}^{(1)}$ for $h_{w,p}$ and $h_{c,p}^{(L_p)}, h_{c,p}^{(L_p-1)}, \dots, h_{c,p}^{(1)}$ for $h_{c,p}$. Only the hidden and output layers are called computational layers, as the input layer simply transmits its input arguments to the next layer (Aggarwal, 2018, p. 6). The output values of each hidden layer are

called hidden units in the hidden layer, and all indicators, components, and hidden units are collectively called nodes. The number of hidden units in the l th hidden layer is denoted by $R_p^{(l)}$.

Figure B1 depicts an example of layers for $h_{w,p}$. The input layer includes three indicators for a component, whereas the output layer contains the component. There are two hidden layers, each of which contains four hidden units, signified by dotted hexagons. A hidden unit refers to an element of the output vector of each hidden layer. In this example, there are three computational layers for $h_{w,p}$.

=== Insert Figure B1 about here ===

Each node per computational layer is obtained by transforming the weighted sum of nodes in the previous layer in a linear or nonlinear fashion. Specifically, let $s_{p,a}^{(l)}$ denote the a th node in the l th computational layer for $h_{w,p}$ ($a = 1, 2, \dots, R_p^{(l)}; l = 1, 2, \dots, L_p + 1$). Let $\mathbf{s}_p^{(l)} \equiv [s_{p,1}^{(l)}, s_{p,2}^{(l)}, \dots, s_{p,R_p^{(l)}}^{(l)}]'$ and $\mathbf{s}_p^{(0)} \equiv \mathbf{z}_p$. Let $\mathbf{W}_p^{(l)}$ denote an $R_p^{(l-1)}$ by $R_p^{(l)}$ matrix of weights in the l th computational layer, where $R_p^{(0)} \equiv J_p$. Let $\boldsymbol{\beta}_{w,p}^{(l)}$ denote a $R_p^{(l)}$ by 1 vector of biases/intercepts in the l th computational layer. Let $\alpha(\mathbf{x}) = [\alpha_1(x_1), \alpha_2(x_2), \dots, \alpha_M(x_M)]'$ denote an activation function that transforms each entry of the input vector $\mathbf{x}$ linearly or nonlinearly, where $\mathbf{x} = [x_1, x_2, \dots, x_M]'$ is any vector of size M and α_m is an activation function that is applied to x_m ($m = 1, 2, \dots, M$). Then, $h_{w,p}^{(l)}$ in $h_{w,p}$ can be expressed as $\mathbf{s}_p^{(l)} = h_{w,p}^{(l)}(\mathbf{s}_p^{(l-1)}) = \alpha(\mathbf{W}_p^{(l)}\mathbf{s}_p^{(l-1)} + \boldsymbol{\beta}_{w,p}^{(l)})$, indicating that a set of nodes in the l th computational layer for $h_{w,p}$ is defined as a linear or nonlinear function of a weighted sum of nodes in the $(l-1)$ th computational layer for $h_{w,p}$.

We can consider several types of activation function, including identity, sigmoid, tangent hyperbolic (tanh), bounded linear unit (BLU), and rectified linear unit (ReLU). Refer to Nwankpa, Ijomah, Gachagan, and Marshall (2021) for the characteristics of these activation functions. The ReLU function is one of the most popular activation functions for hidden units in

modern deep learning architectures because of its superior performance in multiple-layer cases (e.g., Aggarwal, 2018; Nwankpa et al., 2021). Thus, in DL-GSCA, by default, the ReLU function is used for hidden units, which can be expressed as

$$\alpha_m(x_m) = \max(0, x_m). \quad (\text{B.1})$$

Figure B2(a) displays the ReLU function, which is a continuous, piecewise, linear function.

Based on the property of a piecewise linear function, $h_{w,p}$ involving the ReLU also becomes continuous and piecewise linear in $\mathbf{z}_p$ (Strang, 2019, p. 375). This indicates that $h_{w,p}$ approximates $f_{w,p}$ by a linear combination of multiple line pieces. On the other hand, two activation functions, identity and BLU (Zhangyang et al., 2016), are recommended for the output layer, which can be written as

$$\alpha_m(x_m) = x_m \quad (\text{identity}) \quad (\text{B.2})$$

$$\alpha_m(x_m) = \max(\min(x_m, \tau_{2,m}), \tau_{1,m}) \quad (\text{BLU}), \quad (\text{B.3})$$

where $\tau_{1,m}$ and $\tau_{2,m}$ are the lower and upper limits of x_m , respectively.² Figure B2(b) and Figure B2(c) exhibit the two activation functions. In DL-GSCA, the BLU function is preferred to the identity function for the output layer of $h_{w,p}$ because it can suppress the occurrence of extreme component scores from a test sample by truncating them to lie in the range of the component scores estimated from a training sample (e.g., $[-3, 3]$). Without such a truncation scheme, abnormally large component scores can be obtained from a test sample if this sample includes outliers, leading to large out-of-sample prediction error.

=== Insert Figure B2 about here ===

Figure B3 depicts an example of layers for $h_{c,p}$, which consists of the same number of computational layers as those for $h_{w,p}$. The input layer in $h_{c,p}$ includes a component, whereas the

² In the original BLU function, $\tau_{1,m}$ and $\tau_{2,m}$ are restricted as $1 = \tau_{2,m} = -\tau_{1,m}$.

output layer includes the predicted values of the component's indicators. Let $t_{p,a}^{(l)}$ denote the a th node in the l th computational layer for $h_{c,p}$ ($a=1, 2, \dots, R_p^{(l)}; l=1, 2, \dots, L_p+1$). Let $\mathbf{t}_p^{(l)} \equiv [t_{p,1}^{(l)}, t_{p,2}^{(l)}, \dots, t_{p,R_p^{(l)}}^{(l)}]'$ and $\mathbf{t}_p^{(0)} \equiv \gamma_p$. Let $\mathbf{C}_p^{(l)}$ denote an $R_p^{(l-1)}$ by $R_p^{(l)}$ matrix of loadings, and $\boldsymbol{\beta}_{c,p}^{(l)}$ denote a $R_p^{(l)}$ by 1 vector of biases. Then, $h_{c,p}^{(l)}$ in $h_{c,p}$ can be expressed as $\mathbf{t}_p^{(l)} = h_{c,p}^{(l)}(\mathbf{t}_p^{(l-1)}) = \alpha(\mathbf{C}_p^{(l)}\mathbf{t}_p^{(l-1)} + \boldsymbol{\beta}_{c,p}^{(l)})$, indicating that a set of nodes in the l th computational layer for $h_{c,p}$ is defined as a linear or nonlinear function of a weighted sum of nodes in the $(l-1)$ th computational layer for $h_{c,p}$. The ReLU function is again used for all hidden layers of $h_{c,p}$, while the identity function is for the output layer. The BLU function is not used for the output layer because the values of $\hat{\mathbf{z}}_p$ are not expected to be abnormally extreme as the argument of $h_{c,p}$ (i.e., the score of γ_p) is already truncated not to have an extreme value.

=== Insert Figure B3 about here ===

Appendix C. Parameter estimation **procedure** for DL-GSCA

Given the data of indicators, DL-GSCA aims to estimate $h_{w,p}$, $h_{c,p}$, and $\mathbf{B}$ in such a way that components minimize the average unexplained variance of all dependent variables in the model. Let $\mathbf{d}_{n,p}$ denote a J_p by 1 vector of the standardized scores of $\mathbf{z}_p$ for the n th individual in a sample of N individuals. Let $\mathbf{d}_n \equiv [\mathbf{d}_{n,1}; \mathbf{d}_{n,2}; \dots; \mathbf{d}_{n,P}]$ and $\mathbf{D} \equiv [\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_N]'$. Let $g_{n,p}$ denote the n th individual's standardized score for γ_p , which can be obtained by

$$\mathbf{g}_{n,p} = h_{w,p}(\mathbf{d}_{n,p}). \quad (\text{C.1})$$

Let $\mathbf{g}_n \equiv [g_{n,1}, g_{n,2}, \dots, g_{n,P}]'$ and $\mathbf{G} \equiv [\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_N]'$. Let $\mathbf{e}_n$ denote a $J + P$ by 1 vector of residuals for $[\mathbf{d}_n; \mathbf{g}_n]$. Let $SS(\mathbf{X}) \equiv \text{trace}(\mathbf{X}'\mathbf{X})$, where $\mathbf{X}$ is any matrix. Let $\text{blkdiag}()$ denote an operator that converts a set of input matrices into a block-diagonal matrix that includes each input matrix as its diagonal block. Let $\mathbf{O}_z \equiv \text{blkdiag}(\mathbf{O}_{z,1}, \mathbf{O}_{z,2}, \dots, \mathbf{O}_{z,P})$, where $\mathbf{O}_{z,p} = \mathbf{I}$ if $\mathbf{z}_p$ is explained by γ_p and $\mathbf{O}_{z,p} = \mathbf{0}$ otherwise. Let $\mathbf{O}_\gamma \equiv \text{blkdiag}(\mathbf{o}_{\gamma,1}, \mathbf{o}_{\gamma,2}, \dots, \mathbf{o}_{\gamma,P})$, where $\mathbf{o}_{\gamma,p}$ is one if γ_p is explained by other components and zero otherwise. Let $\mathbf{O} \equiv \text{blkdiag}(\mathbf{O}_z, \mathbf{O}_\gamma)$. Let T denote the number of dependent variables in the model. Given $\mathbf{D}$, DL-GSCA seeks to minimize the following optimization criterion

$$\begin{aligned} \varphi(h_{w,p}, h_{c,p}, \mathbf{B}) &= (TN)^{-1} \sum_{n=1}^N SS(\mathbf{e}_n \mathbf{O}) \\ &= (TN)^{-1} \sum_{p=1}^P \left(SS((\mathbf{d}_{n,p} - h_{c,p}(g_{n,p}))\mathbf{O}_{z,p}) + SS(g_{n,p} - \mathbf{g}_n \mathbf{b}_p)\mathbf{o}_{\gamma,p} \right), \end{aligned} \quad (\text{C.2})$$

subject to the standardization constraints on components. The criterion (C.2) is equivalent to the average (squared) prediction error for all the dependent variables in the model.

As this constrained optimization criterion cannot be minimized in closed form, we develop an alternating least squares (ALS) algorithm to minimize (C.2) iteratively. Specifically, the ALS algorithm begins by assigning initial values to the parameters in the following way. It utilizes the estimates of GSCA as the initial values of $\mathbf{G}$ and $\mathbf{B}$ and then, applies the He

initialization (He et al., 2015) to assign random initial values to $h_{w,p}$ and $h_{c,p}$. The He initialization is more efficient for convergence than any other procedures when the Relu function is used for deep learning (Kumar, 2017). Then, the algorithm alternates the following two steps until the difference in the value of (C.2) between consecutive iterations becomes smaller than a prescribed tolerance level (e.g., .0001).

Step 1. Update $h_{w,p}$, $h_{c,p}$, and $\mathbf{b}_p$ with $h_{w,q}$, $h_{c,q}$, and $\mathbf{b}_q$ fixed for all $q \in \mathbb{Q}_p$, where $\mathbb{Q}_p$ is a set of integers from 1 to P except for p . Let $\boldsymbol{\varsigma}_p$ denote the p th row of $\mathbf{B} - \mathbf{I}$, and $\boldsymbol{\Omega}_p$ denote $\mathbf{I} - \mathbf{B}$ with the p th row removed. Let $\boldsymbol{\pi}_p$ denote a row vector of the nonzero entries of $\boldsymbol{\varsigma}_p$. Let $\boldsymbol{\Phi}_p$ denote a matrix of the columns of $\boldsymbol{\Omega}_p$ that correspond to the non-zero entries of $\boldsymbol{\varsigma}_p$. Let $\boldsymbol{\Omega}_{p,z}$ denote a matrix of the columns of $\boldsymbol{\Omega}_p$ that correspond to the zero entries of $\boldsymbol{\varsigma}_p$. With the standardization constraints imposed on components, the objective function (C.2) can be re-expressed as

$$\begin{aligned} \varphi(h_w, h_c, \mathbf{B}) &= (TN)^{-1} \sum_{n=1}^N \left(\sum_{k=1}^P \left(SS((\mathbf{d}_{n,k} - h_{c,k}(\mathbf{g}_{n,k}))\mathbf{O}_{z,k}) \right) + SS((\mathbf{I} - \mathbf{B})'\mathbf{g}_n \mathbf{O}_\gamma) \right) \\ &= (TN)^{-1} \sum_{n=1}^N \left(\sum_{k=1}^P \left(SS((\mathbf{d}_{n,k} - h_{c,k}(\mathbf{g}_{n,k}))\mathbf{O}_{z,k}) \right) + SS((\boldsymbol{\Omega}_p'\mathbf{g}_{n,-p} - \boldsymbol{\varsigma}_p'\mathbf{g}_{n,p})\mathbf{O}_\gamma) \right) \\ &= (TN)^{-1} \sum_{n=1}^N \left(SS([\mathbf{d}_{n,p} - h_{c,p}(h_{w,p}(\mathbf{d}_{n,p}))]\mathbf{O}_{z,p}, (\boldsymbol{\Phi}_p'\mathbf{g}_{n,-p} - \boldsymbol{\pi}_p'h_{w,p}(\mathbf{d}_{n,p}))\mathbf{O}_\gamma] \right) + \psi_1, \end{aligned} \quad (\text{C.3})$$

where $\psi_1 = (TN)^{-1} \sum_{n=1}^N \left(\sum_{q \in \mathbb{Q}_p} \left(SS((\mathbf{d}_{n,q} - h_{c,q}(\mathbf{g}_{n,q}))\mathbf{O}_{z,q}) \right) + SS(\boldsymbol{\Phi}_p'\mathbf{g}_{n,-p} \mathbf{O}_\gamma) \right)$ is a constant,

indicating that $h_{c,p}$, $h_{w,p}$, and $\boldsymbol{\pi}_p$ can be estimated with $h_{w,q}$, $h_{c,q}$, and $\mathbf{b}_q$ fixed for all $q \in \mathbb{Q}_p$ in such a way that the p th component ($\mathbf{g}_{n,p}$) can minimize the sum of squared residuals for $\mathbf{d}_{n,p}$ and $\boldsymbol{\Phi}_p'\mathbf{g}_{n,-p}$. The Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm (Broyden, 1970; Fletcher, 1970; Goldfarb, 1970; Shanno, 1970) can be used to minimize (C.3) subject to the standardization constraints. **We employ the *fminunc* function in MATLAB, which allows users to implement the BFGS algorithm without manually calculating the gradient of (C.3) (The**

MathWorks Inc., 2022). Once $h_{c,p}$, $h_{w,p}$, and π_p are updated, the non-zero elements of $\mathbf{b}_p$ are updated by the estimated π_p . Then, h_w , h_c , and $\mathbf{B}$ are updated by $h_{w,p}$, $h_{c,p}$, and $\mathbf{b}_p$.

Step 2. Update $\mathbf{B}$ for h_w and h_c fixed. Let $\boldsymbol{\theta}$ denote a column vector of the non-zero elements of $\text{vec}(\mathbf{B})$, where $\text{vec}()$ is an operator to convert an input matrix into a column vector by stacking the columns of the input matrix vertically in order. The operator $\otimes$ denotes the Kronecker product. Let Ξ denote a matrix of the columns of $\mathbf{O}_\gamma \otimes \mathbf{G}$ that correspond to the non-zero elements of $\text{vec}(\mathbf{B})$. We can re-write (C.2) as

$$\begin{aligned}
& \varphi(h_{w,p}, h_{c,p}, \mathbf{B}) \\
&= (TN)^{-1} \sum_{n=1}^N (SS((\mathbf{g}_n - \mathbf{B}'\mathbf{g}_n)\mathbf{O}_\gamma)) + \psi_2 \\
&= (TN)^{-1} SS(\mathbf{G}\mathbf{O}_\gamma - \mathbf{G}\mathbf{B}\mathbf{O}_\gamma) + \psi_2 \\
&= (TN)^{-1} SS(\text{vec}(\mathbf{G}\mathbf{O}_\gamma) - (\mathbf{O}_\gamma \otimes \mathbf{G})\text{vec}(\mathbf{B})) + \psi_2 \\
&= (TN)^{-1} SS(\text{vec}(\mathbf{G}\mathbf{O}_\gamma) - \Xi\boldsymbol{\theta}) + \psi_2,
\end{aligned} \tag{C.4}$$

where $\psi_2 = (TN)^{-1} \sum_{n=1}^N (SS((\mathbf{d}_n - h_c(\mathbf{g}_n))\mathbf{O}_z))$ is a constant. The least squares estimate of $\boldsymbol{\theta}$ can be obtained by

$$\hat{\boldsymbol{\theta}} = (\Xi'\Xi)^{-1}\Xi'\text{vec}(\mathbf{G}\mathbf{O}_\gamma), \tag{C.5}$$

from which the non-zero elements of $\mathbf{B}$ are updated.

DL-GSCA repeats the ALS algorithm with different random initial values to avoid potential convergence to local minima (Hwang & Takane, 2014, p. 24). Then, DL-GSCA selects a set of parameter estimates that leads to the smallest value of (C.2) as the final one.

Appendix D. Predictive feedforward search algorithm used for tuning DL-GSCA's hyperparameters

Prior to applying the ALS algorithm, DL-GSCA should determine the values of two hyperparameters for each component (i.e., L_p and $R_p^{(l)}$). The predictive feedforward search algorithm (Cho, Hwang, et al., 2022), which was originally introduced for variable selection in GSCA, can be utilized for deciding on the values of the hyperparameters in DL-GSCA.

Given candidate sets of the hyperparameter values for a component, the search algorithm generates K pairs of training and validation samples. Then, the algorithm starts with the smallest candidate values of L_p and $R_p^{(l)}$. The DL-GSCA model with these values is fitted to a training sample, and the model's TE^D value is subsequently calculated from the corresponding validation sample. This procedure is reiterated for the remaining pairs of training and validation samples, leading to the calculation of the average TE^D value over K validation samples.

The algorithm proceeds with the next smallest $R_p^{(l)}$ value and calculates the model's average TE^D value again. Then, it compares the two average TE^D values from two different $R_p^{(l)}$ values to see if the TE^D value decreases with an increase in the $R_p^{(l)}$ value. If it does, the process continues with the next smallest $R_p^{(l)}$ value until the model's TE^D value no longer decreases with an increase in $R_p^{(l)}$, or there are no remaining $R_p^{(l)}$ candidates. The $R_p^{(l)}$ value resulting in the minimum average TE^D is selected as the optimal one for the smallest L_p value.

This process is then conducted for the next smallest L_p value with the aim of identifying the optimal $R_p^{(l)}$ value for that L_p . The algorithm subsequently compares the two average TE^D values from two different combinations of L_p and $R_p^{(l)}$ and determines whether an increase in the L_p value reduces the model's TE^D value. If so, the process proceeds with the next smallest L_p until the model's TE^D value no longer decreases with an increase in L_p or there are no further L_p

candidates. The L_p and $R_p^{(l)}$ pair that leads to the smallest average TE^D is selected as the optimal one for that component.

The search algorithm repeats the above procedures for all components. When the sample size is large, it may be too computationally costly to conduct K -fold cross validation. In this case, the algorithm is carried out based on the validation set approach that uses a prescribed percentage of the training sample (e.g., 70%) for training the model and the remaining one for calculating the model's TE^D value.

Appendix E. Formulae for DL-GSCA's model evaluation indices

The overall goodness-of-fit index, FIT^D , is given as

$$\text{FIT}^D = 1 - \frac{\sum_{n=1}^N SS(([\mathbf{d}_n, \mathbf{g}_n] - [h_C(\mathbf{g}_n), \hat{\mathbf{B}}'\mathbf{g}_n])\mathbf{O})}{\sum_{n=1}^N SS([\mathbf{d}_n, \mathbf{g}_n]\mathbf{O})}, \quad (\text{E.1})$$

where $\hat{\mathbf{B}}$ is the estimate of $\mathbf{B}$. The two local goodness-of-fit indices, FIT_M^D and FIT_S^D , are provided as

$$\text{FIT}_M^D = 1 - \frac{\sum_{n=1}^N SS((\mathbf{d}_n - h_C(\mathbf{g}_n))\mathbf{O}_z)}{\sum_{n=1}^N SS(\mathbf{d}_n\mathbf{O}_z)}, \quad (\text{E.2})$$

$$\text{FIT}_S^D = 1 - \frac{\sum_{n=1}^N SS((\mathbf{g}_n - \hat{\mathbf{B}}'\mathbf{g}_n)\mathbf{O}_\gamma)}{\sum_{n=1}^N SS(\mathbf{g}_n\mathbf{O}_\gamma)}. \quad (\text{E.3})$$

On the other hand, TE^D is defined as

$$\text{TE}^D = \frac{\sum_{n=1}^{N_{tt}} SS(([\mathbf{d}_{n,tt}, \mathbf{g}_{n,tt}] - [h_C(\mathbf{g}_{n,tt}), \hat{\mathbf{B}}'\mathbf{g}_{n,tt}])\mathbf{O})}{\sum_{n=1}^{N_{tt}} SS([\mathbf{d}_{n,tt}, \mathbf{g}_{n,tt}]\mathbf{O})}, \quad (\text{E.4})$$

where N_{tt} is the number of individuals in the test sample; $\mathbf{d}_{n,tt}$ is the n th individual's standardized scores in the test sample ($n = 1, 2, \dots, N_{tt}$); h_W , h_C , and $\hat{\mathbf{B}}$ are the parameter estimates obtained from the training sample; and $\mathbf{g}_{n,tt} \equiv h_W(\mathbf{d}_{n,tt})$. Note that $\mathbf{d}_{n,tt}$ should be standardized using the sample means and standard deviations of indicators obtained from the training sample, not from the corresponding test sample, because h_W , h_C , and $\hat{\mathbf{B}}$ are the estimates obtained from the standardized scores of indicators in the training sample. In addition, two local indices for

evaluating the prediction errors of the component measurement and structural models are defined as follows.

$$\text{TE}_M^D = \frac{\sum_{n=1}^{N_{it}} SS((\mathbf{d}_{n,it} - h_C(\mathbf{g}_{n,it}))\mathbf{O}_z)}{\sum_{n=1}^{N_{it}} SS(\mathbf{d}_{n,it} \mathbf{O}_z)}, \quad (\text{E.5})$$

$$\text{TE}_S^D = \frac{\sum_{n=1}^{N_{it}} SS((\mathbf{g}_{n,it} - \hat{\mathbf{B}}'\mathbf{g}_{n,it})\mathbf{O}_\gamma)}{\sum_{n=1}^{N_{it}} SS(\mathbf{g}_{n,it} \mathbf{O}_\gamma)}. \quad (\text{E.6})$$

Moreover, OPE^D is defined as

$$\text{OPE}^D = \frac{1}{N_{boot}} \sum_k \frac{\sum_{n=1}^{N_k} SS(([\mathbf{d}_{n,k}, \mathbf{g}_{n,k}] - [h_{C,k}(\mathbf{g}_{n,k}), \hat{\mathbf{B}}_k' \mathbf{g}_{n,k}])\mathbf{O})}{\sum_{n=1}^{N_k} SS([\mathbf{d}_{n,k}, \mathbf{g}_{n,k}]\mathbf{O})}, \quad (\text{E.7})$$

where N_{boot} is the number of in-bag and out-of-bag sample sets; N_k is the number of individuals in the k th out-of-bag sample; $\mathbf{d}_{n,k}$ is the n th individual's standardized scores in the k th out-of-bag sample; $h_{W,k}$, $h_{C,k}$, and $\hat{\mathbf{B}}_k$ are the parameter estimates obtained from the k th in-bag sample; and $\mathbf{g}_{n,k} \equiv h_W(\mathbf{d}_{n,k})$. Note that $\mathbf{d}_{n,k}$ should be standardized using the sample means and standard deviations of indicators obtained from the k th in-bag sample. The two local cross-validation indices are defined as follows.

$$\text{OPE}_M^D = \frac{1}{N_{boot}} \sum_k \frac{\sum_{n=1}^{N_k} SS((\mathbf{d}_{n,k} - h_{C,k}(\mathbf{g}_{n,k}))\mathbf{O}_z)}{\sum_{n=1}^{N_k} SS(\mathbf{d}_{n,k} \mathbf{O}_z)}, \quad (\text{E.8})$$

$$\text{OPE}_S^D = \frac{1}{N_{boot}} \sum_k \frac{\sum_{n=1}^{N_k} SS((\mathbf{g}_{n,k} - \hat{\mathbf{B}}_k' \mathbf{g}_{n,k}) \mathbf{O}_\gamma)}{\sum_{n=1}^{N_k} SS(\mathbf{g}_{n,k} \mathbf{O}_\gamma)}. \quad (\text{E.9})$$

Lastly, $\Delta\text{TE}_{p,q}$, which is used to evaluate the predictive power of each individual predictor for its dependent component, is defined as

$$\Delta\text{TE}_{p,q} = \frac{\sum_{n=1}^{N_u} (\mathbf{g}_{n,q,tt} - \hat{\mathbf{b}}_q' \mathbf{g}_{n,tt})^2 - \sum_{n=1}^{N_u} (\mathbf{g}_{n,q,tt} - (\hat{\mathbf{b}}_q \mathbf{I}_{p0})' \mathbf{g}_{n,tt})^2}{\sum_{n=1}^{N_u} (\mathbf{g}_{n,q,tt} - \hat{\mathbf{b}}_q' \mathbf{g}_{n,tt})^2}, \quad (\text{E.10})$$

where $\mathbf{g}_{n,q,tt}$ is the q th component's score in $\mathbf{g}_{n,tt}$, $\hat{\mathbf{b}}_q$ is the q th column of $\hat{\mathbf{B}}$, and $\mathbf{I}_{p0}$ is the identity matrix of order P , whose p th diagonal entry is zero.

Appendix F. **Data** generating procedure for the simulation study

Let $d_{n,p,i}$ denote the n th individual's score of the i th indicator for the p th component in the prototype model in Figure 2 ($n = 1, 2, \dots, N; i = 1, 2, 3; p = 1, 2, 3$). To generate $d_{n,p,i}$, we begin by obtaining the implied correlation matrix of the GSCA model based on Cho and Choi's (2020) procedure, under the assumption that all indicators are linearly associated. This implied covariance matrix is provided in Table F1. We draw a sample of N individuals from a multivariate normal distribution with zero means and the covariance matrix. Let $y_{n,p,i}$ denote the n th individual's score of the i th indicator for the p th linear component. We then generate $d_{n,p,i}$ as follows: $d_{n,1,1} = y_{n,1,1}$, $d_{n,1,2} = (y_{n,1,2})^2$, $d_{n,1,3} = -(y_{n,1,3})^2$, $d_{n,2,1} = y_{n,2,1}$, $d_{n,2,2} = (y_{n,2,1} \times y_{n,2,2})$, $d_{n,2,3} = (y_{n,2,2} \times y_{n,2,3})$, $d_{n,3,1} = y_{n,3,1}$, $d_{n,3,2} = y_{n,3,2}$, and $d_{n,3,3} = y_{n,3,3}$.

=== Insert Table F1 about here ===

References

- Aggarwal, C. C. (2018). An Introduction to neural networks. In C. C. Aggarwal (Ed.), *Neural Networks and Deep Learning: A Textbook* (pp. 1–52). Springer International Publishing. https://doi.org/10.1007/978-3-319-94463-0_1
- Basco, R., Hair, J. F., Ringle, C. M., & Sarstedt, M. (2021). Advancing family business research through modeling nonlinear relationships: Comparing PLS-SEM and multiple regression. *Journal of Family Business Strategy*, 100457. <https://doi.org/10.1016/j.jfbs.2021.100457>
- Blanchflower, D. G., & Oswald, A. J. (2005). Happiness and the human development index: The paradox of Australia. *Australian Economic Review*, 38(3), 307–318. <https://doi.org/10.1111/j.1467-8462.2005.00377.x>
- Bollen, K. A., & Bauldry, S. (2011). Three Cs in measurement models: Causal indicators, composite indicators, and covariates. *Psychological Methods*, 16(3), 265–284. <https://doi.org/10.1037/a0024448>
- Broyden, C. G. (1970). The convergence of a class of double-rank minimization algorithms 1. General considerations. *IMA Journal of Applied Mathematics*, 6(1), 76–90. <https://doi.org/10.1093/imamat/6.1.76>
- Cantril, H. (1965). *The pattern of human concerns*. Rutgers University Press.
- Carroll, J. D. (1968). Generalization of canonical correlation analysis to three or more sets of variables. *Proceedings of the 76th Annual Convention of the American Psychological Association*, 3, 227–228.
- Cho, G. (2023). *Constructs may or may not be latent: Studies on two domains of structural equation modeling* [Unpublished doctoral dissertation]. McGill University.
- Cho, G., & Choi, J. Y. (2020). An empirical comparison of generalized structured component analysis and partial least squares path modeling under variance-based structural equation models. *Behaviormetrika*, 47(1), 243–272. <https://doi.org/10.1007/s41237-019-00098-0>
- Cho, G., Hwang, H., Sarstedt, M., & Ringle, C. M. (2022). A prediction-oriented specification search algorithm for generalized structured component analysis. *Structural Equation Modeling: A Multidisciplinary Journal*. <https://doi.org/10.1080/10705511.2022.2057315>
- Cho, G., Jung, K., & Hwang, H. (2019). Out-of-bag prediction error: A cross validation index for generalized structured component analysis. *Multivariate Behavioral Research*, 54(4), 505–513. <https://doi.org/10.1080/00273171.2018.1540340>
- Cho, G., Sarstedt, M., & Hwang, H. (2022). A comparative evaluation of factor- and component-based structural equation modeling methods under (in)consistent model specifications. *British Journal of Mathematical and Statistical Psychology*, 75(2), 220–251. <https://doi.org/10.1111/bmsp.12255>
- Choi, S., Lee, S., Huh, I., Hwang, H., & Park, T. (2020). HisCoM-G×E: Hierarchical structural component analysis of gene-based gene–environment interactions. *International Journal of Molecular Sciences*, 21(18), 1–11. <https://doi.org/10.3390/ijms21186724>

- Csaji, B. C. (2001). *Approximation with artificial neural networks* [Eötvös Loránd University]. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.101.2647&rep=rep1&type=pdf>
- Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4), 303–314. <https://doi.org/10.1007/BF02551274>
- de Jong, S., & Kiers, H. A. L. (1992). Principal covariates regression: Part I. Theory. *Chemometrics and Intelligent Laboratory Systems*, 14(1), 155–164. [https://doi.org/10.1016/0169-7439\(92\)80100-I](https://doi.org/10.1016/0169-7439(92)80100-I)
- Efron, B. (1979). Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1), 1–26. <https://doi.org/10.1214/aos/1176344552>
- Efron, B. (1982). *The jackknife, the bootstrap and other resampling plans*. SIAM. <https://doi.org/10.1137/1.9781611970319>
- Fletcher, R. (1970). A new approach to variable metric algorithms. *The Computer Journal*, 13(3), 317–322. <https://doi.org/10.1093/comjnl/13.3.317>
- Gensowski, M., Heckman, J., & Savelyev, P. (2011). *The effects of education, personality, and IQ on earnings of high-ability men*. https://conference.iza.org/conference_files/CoNoCoSk2011/gensowski_m6556.pdf
- Goldfarb, D. (1970). A family of variable-metric methods derived by variational means. *Mathematics of Computation*, 24(109), 23–26. <https://doi.org/10.1090/S0025-5718-1970-0258249-6>
- Graham, J. W. (2008). Missing data analysis: Making it work in the real world. *Annual Review of Psychology*, 60(1), 549–576. <https://doi.org/10.1146/annurev.psych.58.110405.085530>
- Hall, J., & Helliwell, J. (2014). *Happiness and human development*. http://akwl.org/wp-content/uploads/2019/12/happiness_and_hd.pdf
- Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning*. Springer. <https://doi.org/10.1007/978-0-387-21606-5>
- He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 1026–1034. <https://doi.org/10.1109/ICCV.2015.123>
- Hwang, H. (2009). Regularized generalized structured component analysis. *Psychometrika*, 74(3), 517–530. <https://doi.org/10.1007/S11336-009-9119-Y>
- Hwang, H., & Cho, G. (2020). Global least squares path modeling: A full-information alternative to partial least squares path modeling. *Psychometrika*, 85, 947–972. <https://doi.org/10.1007/s11336-020-09733-2>
- Hwang, H., Cho, G., & Choo, H. (2023). *GSCA Pro (Version 1.2.1) [Computer software]*. <http://www.gscapro.com>.
- Hwang, H., Cho, G., Jin, M. J., Ryoo, J. H., Choi, Y., & Lee, S.-H. (2021). A knowledge-based multivariate statistical method for examining gene-brain-behavioral/cognitive relationships:

- Imaging genetics generalized structured component analysis. *PloS One*, 16(3), e0247592. <https://doi.org/10.1371/journal.pone.0247592>
- Hwang, H., Desarbo, W. S., & Takane, Y. (2007). Fuzzy clusterwise generalized structured component analysis. *Psychometrika*, 72(2), 181–198. <https://doi.org/10.1007/s11336-005-1314-x>
- Hwang, H., Ho, M.-H. R., & Lee, J. (2010). Generalized structured component analysis with latent interactions. *Psychometrika*, 75(2), 228–242. <https://doi.org/10.1007/s11336-010-9157-5>
- Hwang, H., Sarstedt, M., Cheah, J.-H., & Ringle, C. M. (2020). A concept analysis of methodological research on composite-based structural equation modeling: Bridging PLSPM and GSCA. *Behaviormetrika*, 47(1), 219–241. <https://doi.org/10.1007/s41237-019-00085-5>
- Hwang, H., & Takane, Y. (2004). Generalized structured component analysis. *Psychometrika*, 69(1), 81–99. <https://doi.org/10.1007/BF02295841>
- Hwang, H., & Takane, Y. (2014). *Generalized structured component analysis: A component-based approach to structural equation modeling*. Chapman and Hall/CRC Press.
- Kettenring, J. R. (1971). Canonical analysis of several sets of variables. *Biometrika*, 58(3), 433–451. <https://doi.org/10.1093/biomet/58.3.433>
- Kumar, S. K. (2017). *On weight initialization in deep neural networks*. <https://arxiv.org/abs/1704.08863>
- Le, P. V. (2014). *More schooling is not always better: Evidence from an Instrumental variables approach to educational reform in Vietnam*. [https://fsppm.fulbright.edu.vn/cache/More Schooling Is Not Always Better-2014-12-19-08341367.pdf](https://fsppm.fulbright.edu.vn/cache/More%20Schooling%20Is%20Not%20Always%20Better-2014-12-19-08341367.pdf)
- Le, Q. V. (2015). A tutorial on deep learning part 2: Autoencoders, convolutional neural networks and recurrent neural networks. *Google Brain*, 20, 1–20. <http://ai.stanford.edu/~quocle/tutorial2.pdf>
- Lecun, Y., Bengio, Y., & Hinton, G. E. (2015). Deep learning. *Nature*, 521(7553), 436–444. <https://doi.org/10.1038/nature14539>
- Lu, Z., Pu, H., Wang, F., Hu, Z., & Wang, L. (2017). The expressive power of neural networks: A view from the width. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems* (Vol. 30, pp. 6231–6239). Curran Associates, Inc. <https://proceedings.neurips.cc/paper/2017/file/32cbf687880eb1674a07bf717761dd3a-Paper.pdf>
- Mas, J. F., & Flores, J. J. (2008). The application of artificial neural networks to the analysis of remotely sensed data. *International Journal of Remote Sensing*, 29(3), 617–663. <https://doi.org/10.1080/01431160701352154>
- Noorbakhsh, F. (1998). The human development index: Some technical issues and alternative indices. *Journal of International Development*, 10(5), 589–605.

- [https://doi.org/10.1002/\(SICI\)1099-1328\(199807/08\)10:5<589::AID-JID484>3.0.CO;2-S](https://doi.org/10.1002/(SICI)1099-1328(199807/08)10:5<589::AID-JID484>3.0.CO;2-S)
- Nübler, I. (1995). The human development index revisited. *Intereconomics*, 30(4), 171–176.
<https://doi.org/10.1007/BF02928088>
- Nwankpa, C. E., Ijomah, W., Gachagan, A., & Marshall, S. (2021). Activation functions. *2nd International Conference on Computational Sciences and Technology*, 124–133.
https://pure.strath.ac.uk/ws/portalfiles/portal/118946797/Nwankpa_et_al_ICCST_2021_Activation_functions_comparison_of_trends_in_practice.pdf
- Park, J. H. (1994). *Returns to schooling: A peculiar deviation from linearity* (No. 714; Working Papers, Issue 714). <https://ideas.repec.org/p/pri/indrel/335.html>
- Ranis, G., Stewart, F., & Samman, E. (2006). Human development: Beyond the human development index. *Journal of Human Development*, 7(3), 323–358.
<https://doi.org/10.1080/14649880600815917>
- Romdhani, H., Hwang, H., Paradis, G., Roy-Gagnon, M. H., & Labbe, A. (2015). Pathway-based association study of multiple candidate genes and multiple traits using structural equation models. *Genetic Epidemiology*, 39(2), 101–113. <https://doi.org/10.1002/gepi.21872>
- Rosenblatt, F. (1962). *Principles of neurodynamics: Perceptrons and the theory of brain mechanisms*. Spartan Books. <https://apps.dtic.mil/sti/citations/AD0256582>
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088), 533–536. <https://doi.org/10.1038/323533a0>
- Sagar, A. D., & Najam, A. (1998). The human development index: a critical review. *Ecological Economics*, 25(3), 249–264. [https://doi.org/10.1016/S0921-8009\(97\)00168-7](https://doi.org/10.1016/S0921-8009(97)00168-7)
- Shanno, D. F. (1970). Conditioning of quasi-newton methods for function minimization. *Mathematics of Computation*, 24(111), 647–656. <https://doi.org/10.2307/2004840>
- Silva, J., de Keulenaer, F., & Johnstone, N. (2012). *Environmental quality and life satisfaction* (OECD Environment Working Papers, Issue 44). <https://doi.org/10.1787/5k9cw678dlr0-en>
- Strang, G. (2019). *Linear algebra and learning from data* (1st ed.). Wellesley-Cambridge Press.
- Takane, Y., & Hwang, H. (2005). An extended redundancy analysis and its applications to two practical examples. *Computational Statistics and Data Analysis*, 49(3), 785–808.
<https://doi.org/10.1016/j.csda.2004.06.004>
- Takane, Y., & Hwang, H. (2007). Regularized linear and kernel redundancy analysis. *Computational Statistics & Data Analysis*, 52(1), 394–405.
<https://doi.org/10.1016/j.csda.2007.02.014>
- Takane, Y., Kiers, H. A. L., & de Leeuw, J. (1995). Component analysis with different sets of constraints on different dimensions. *Psychometrika*, 60(2), 259–280.
<https://doi.org/10.1007/BF02301416>
- Tenenhaus, M., Tenenhaus, A., & Groenen, P. J. F. (2017). Regularized generalized canonical correlation analysis: A framework for sequential multiblock component methods. *Psychometrika*, 82(3), 737–777. <https://doi.org/10.1007/s11336-017-9573-x>

- The MathWorks Inc. (2022). *MATLAB Optimization toolbox: User's Guide*. The MathWorks Inc. https://www.mathworks.com/help/pdf_doc/optim/optim.pdf
- UNDP. (1990). *Human development report 1990: Concept and measurement of human development*. <http://www.hdr.undp.org/en/reports/global/hdr1990>
- UNDP. (2010). *Human development report 2010: The real wealth of nations*. <http://hdr.undp.org/en/content/human-development-report-2010>
- UNDP. (2016). *Human development report 2016: Human development for everyone*. <http://hdr.undp.org/en/content/human-development-report-2016>
- Urban, C. J., & Gates, K. M. (2021). Deep learning: A primer for psychologists. *Psychological Methods*. <https://doi.org/10.1037/met0000374>
- van den Wollenberg, A. L. (1977). Redundancy analysis an alternative for canonical correlation analysis. *Psychometrika*, 42(2), 207–219. <https://doi.org/10.1007/BF02294050>
- van der Leeden, R. (1990). *Reduced rank regression with structured residuals*. DSWO Press.
- Zhangyang, W., Yingzhen, Y., Shiyu, C., Qing, L., & Huang, T. S. (2016). Learning a deep l_∞ encoder for hashing. In S. Kambhampati (Ed.), *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence International Joint Conference on Artificial Intelligence* (pp. 2174–2180). AAAI Press.

Table F1. The correlation matrix of indicators used in the simulation study.

	y _{1,1}	y _{1,2}	y _{1,3}	y _{2,1}	y _{2,2}	y _{2,3}	y _{3,1}	y _{3,2}	y _{3,3}
y _{1,1}	1.000	0.800	0.800	0.433	0.433	0.433	0.303	0.303	0.303
y _{1,2}	0.800	1.000	0.800	0.433	0.433	0.433	0.303	0.303	0.303
y _{1,3}	0.800	0.800	1.000	0.433	0.433	0.433	0.303	0.303	0.303
y _{2,1}	0.433	0.433	0.433	1.000	0.800	0.800	-0.303	-0.303	-0.303
y _{2,2}	0.433	0.433	0.433	0.800	1.000	0.800	-0.303	-0.303	-0.303
y _{2,3}	0.433	0.433	0.433	0.800	0.800	1.000	-0.303	-0.303	-0.303
y _{3,1}	0.303	0.303	0.303	-0.303	-0.303	-0.303	1.000	0.800	0.800
y _{3,2}	0.303	0.303	0.303	-0.303	-0.303	-0.303	0.800	1.000	0.800
y _{3,3}	0.303	0.303	0.303	-0.303	-0.303	-0.303	0.800	0.800	1.000

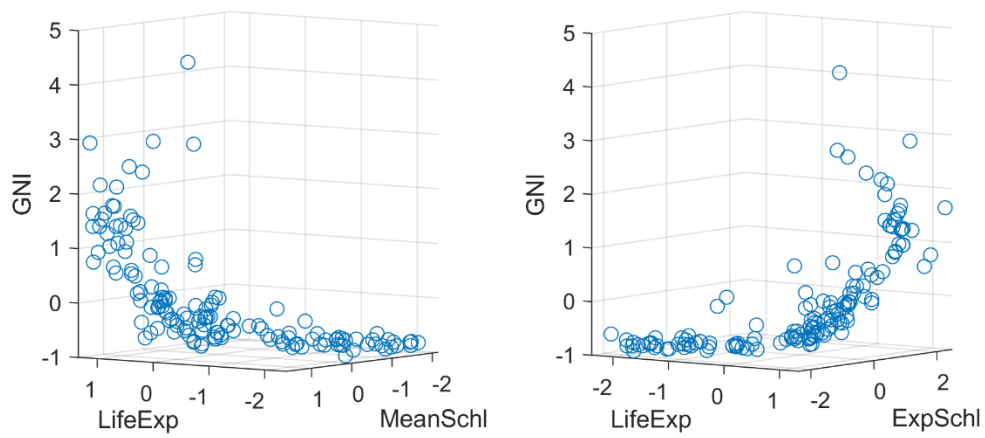


Figure 1. Scatterplots of four indicators for the Human Development Index (HDI).

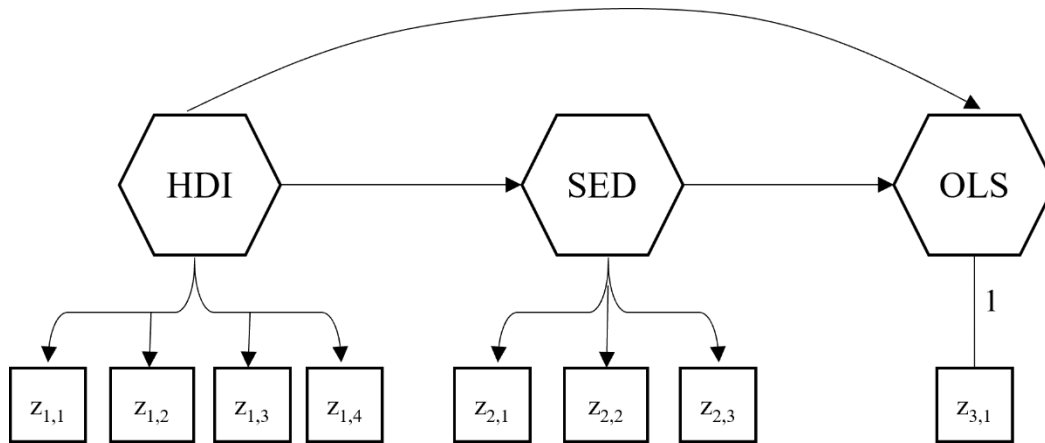


Figure 2. The Human Development Index (HDI) model specified for the empirical application.

Hexagons and squares denote components and indicators, respectively. Each set of curved arrows shows which block of indicators is to be explained by which component. A straight line labelled with 1 indicates that OLS is set to be identical to $z_{3,1}$. Error terms are omitted.

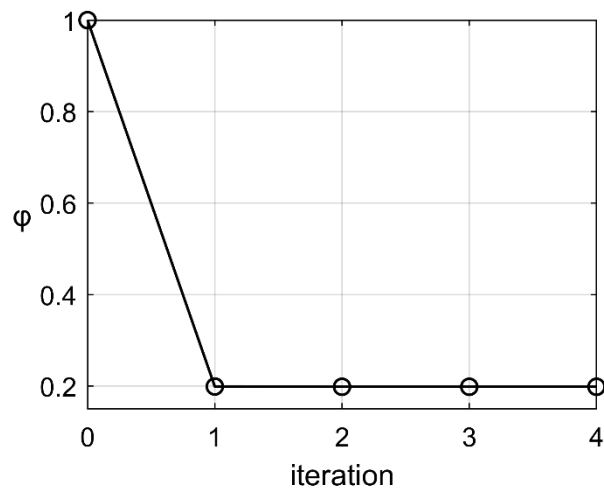


Figure 3. Plot of the optimization criterion value (C.2) versus the number of iterations.

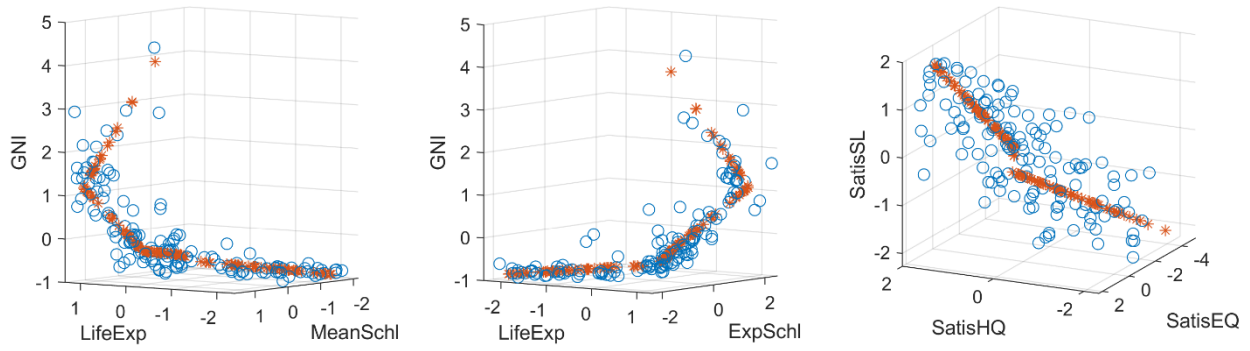
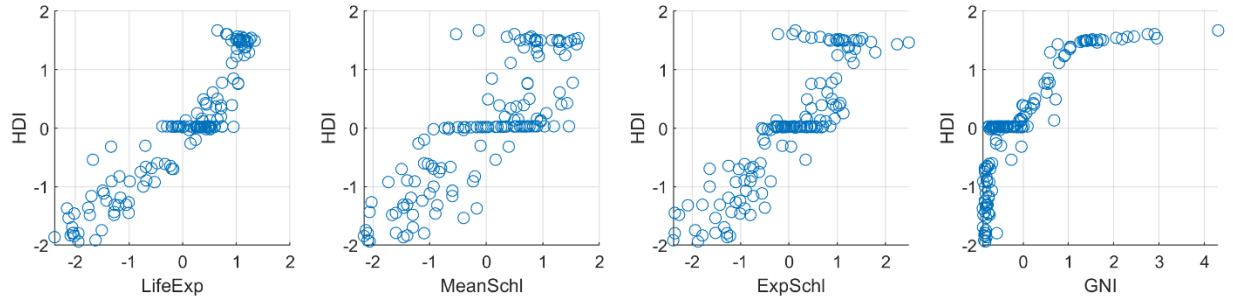


Figure 4. Scatterplots of indicators' scores (blue circles) and their predicted values (orange stars) obtained from DL-GSCA in the empirical application.

(A) HDI



(B) SED

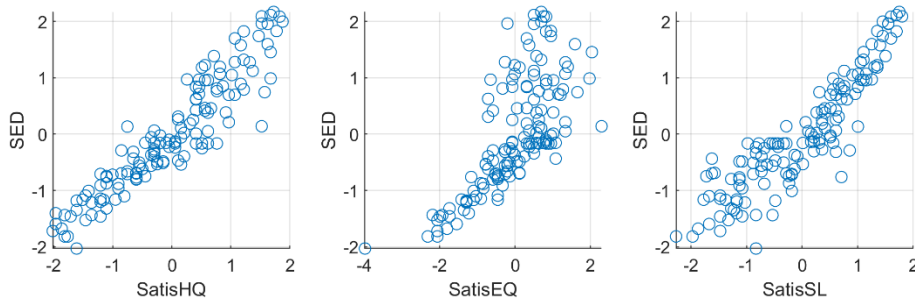


Figure 5. Scatterplots of two components (HDI and SED) and their indicators obtained from DL-GSCA in the empirical application.

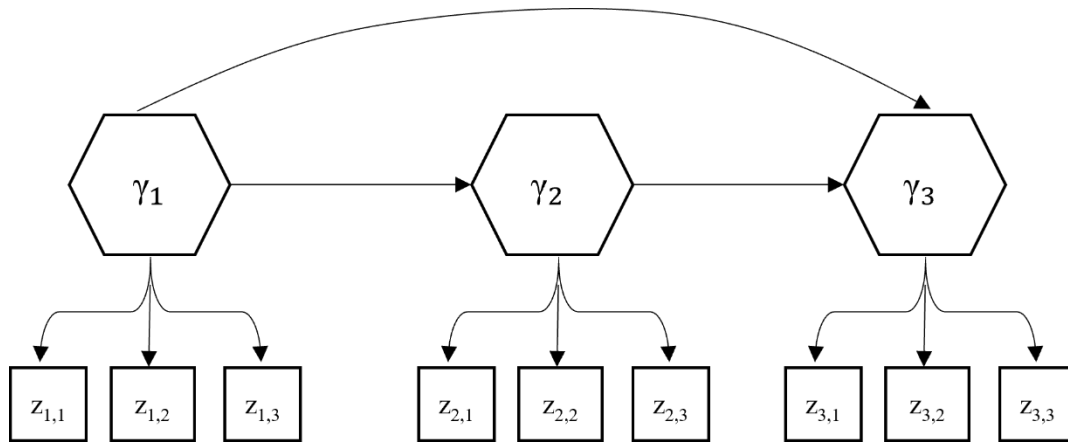


Figure 6. A population DL-GSCA model used in the simulation study.

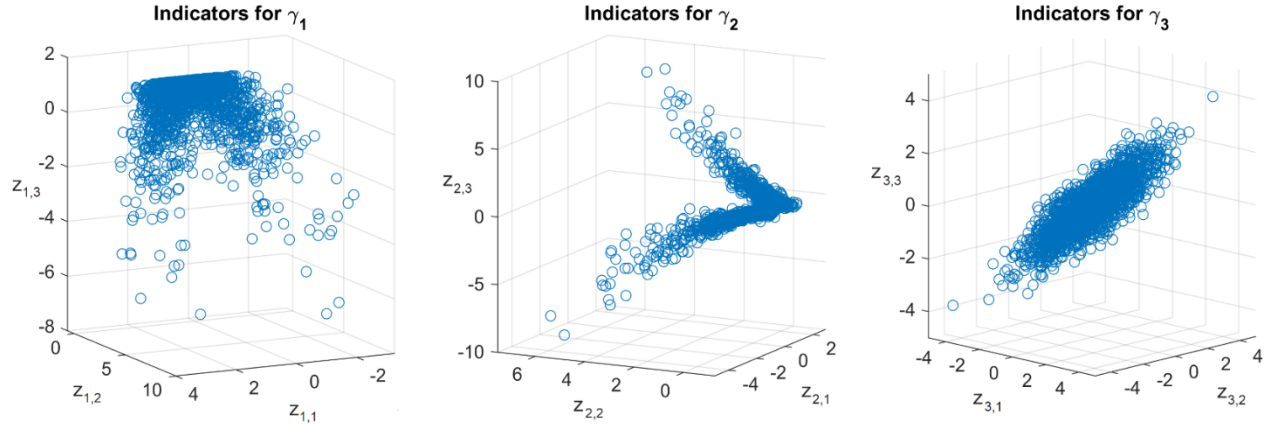


Figure 7. Scatterplots of indicators per component in the test sample for the simulation study.

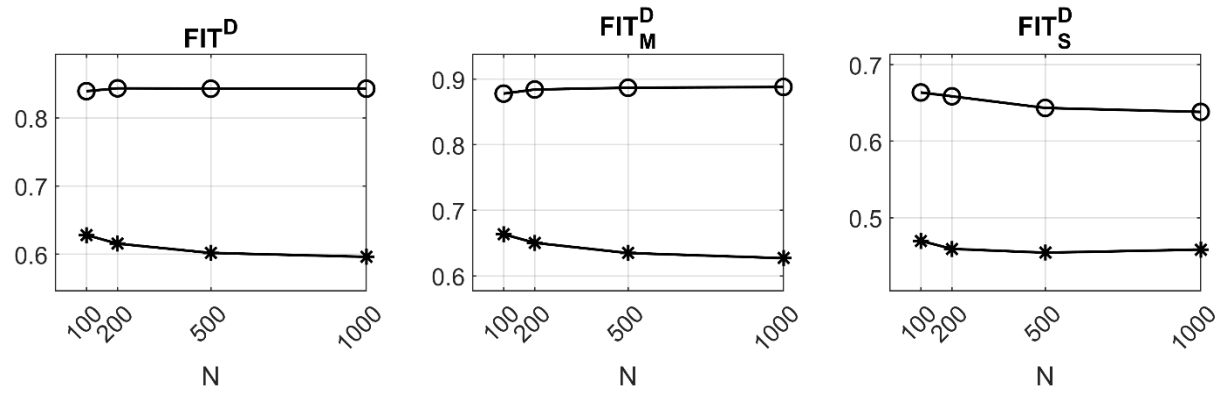


Figure 8. In-sample performance of DL-GSCA and GSCA in the simulation study. The average values of FIT^D , FIT_M^D , and FIT_S^D obtained from DL-GSCA and GSCA per sample size. O = DL-GSCA and * = GSCA.

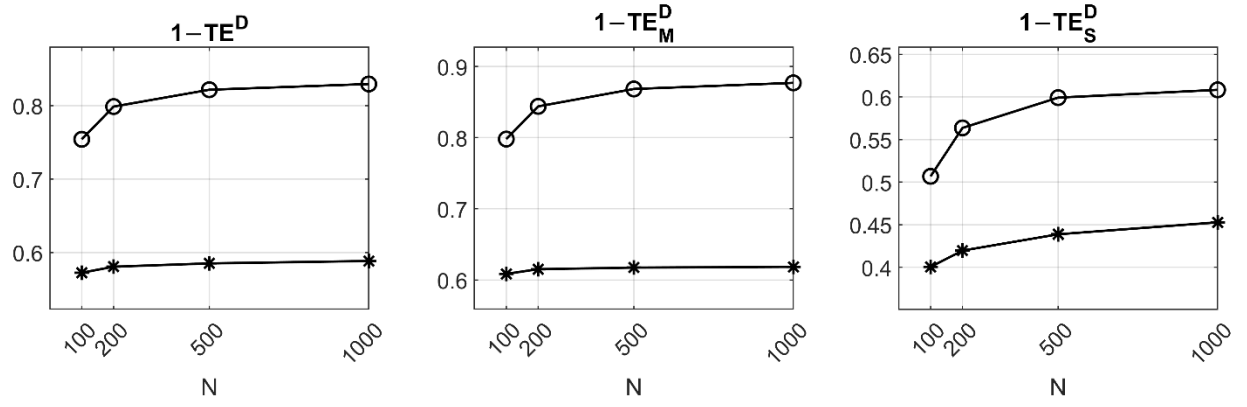
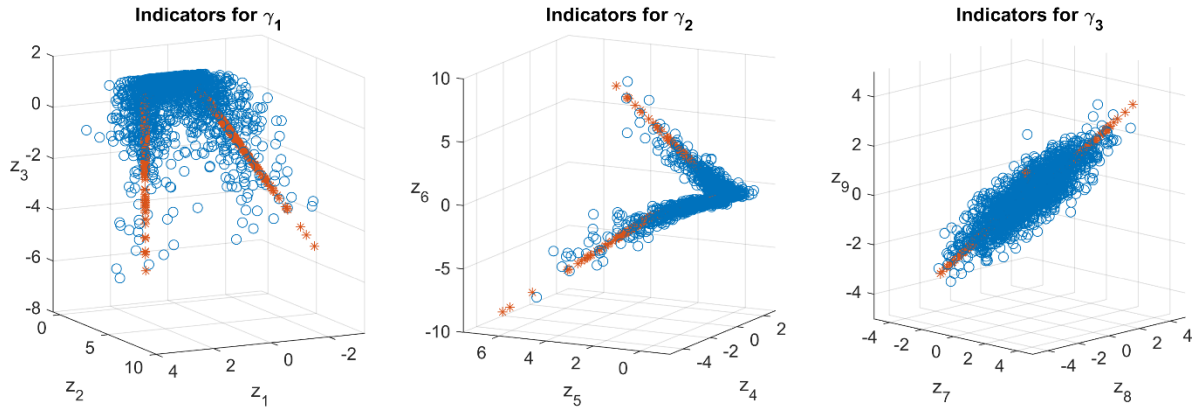


Figure 9. Out-of-sample performance of DL-GSCA and GSCA in the simulation study. The average values of $1 - TE^D$, $1 - TE_M^D$, and $1 - TE_S^D$ obtained from DL-GSCA and GSCA per sample size. O = DL-GSCA and * = GSCA.

(A) DL-GSCA



(B) GSCA

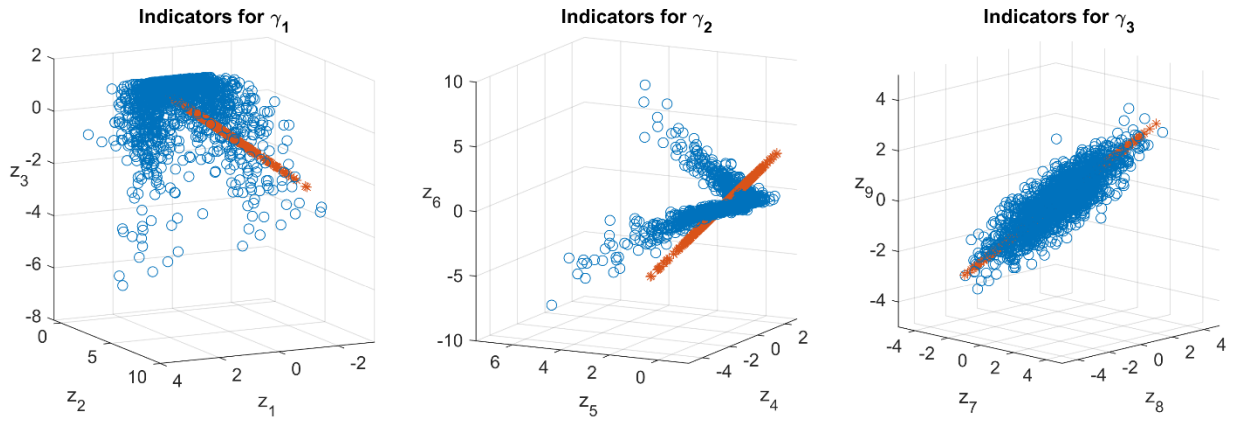


Figure 10. Scatterplots of indicators' scores (blue circles) and their predicted values (orange stars) per component obtained from DL-GSCA and GSCA in the simulation study.

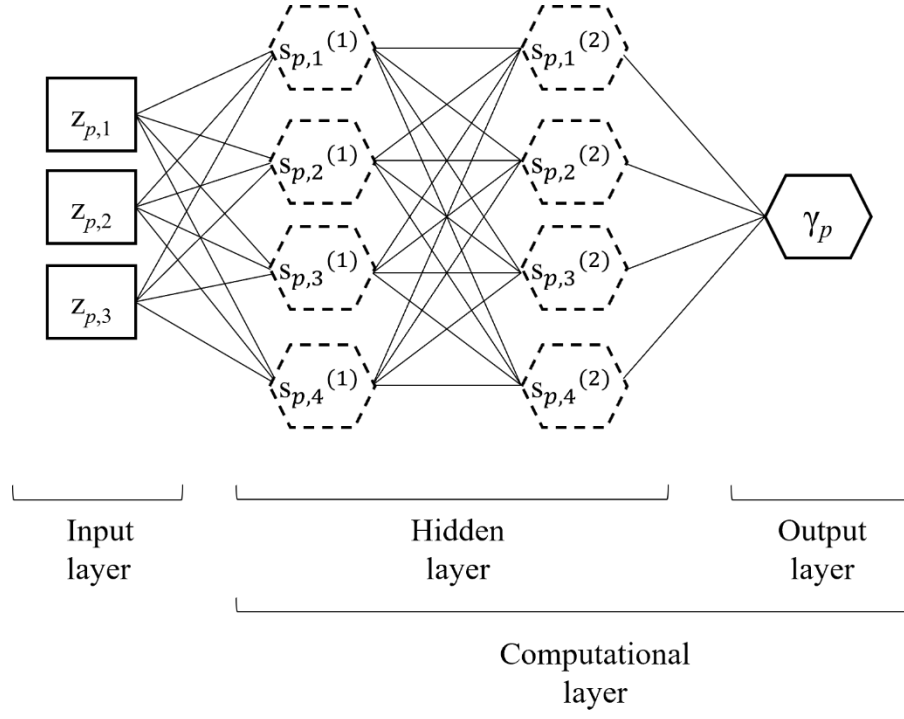


Figure B1. An example of layers for $h_{w,p}$ in DL-GSCA's weighted relation model. Dotted circles represent hidden units per hidden layer. Straight lines denote weight parameters.

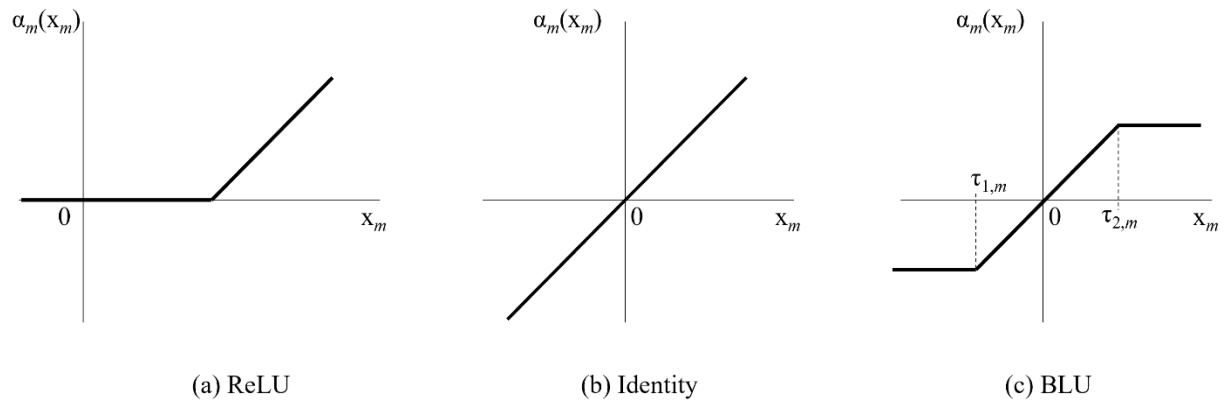


Figure B2. Three types of activation functions used in DL-GSCA.

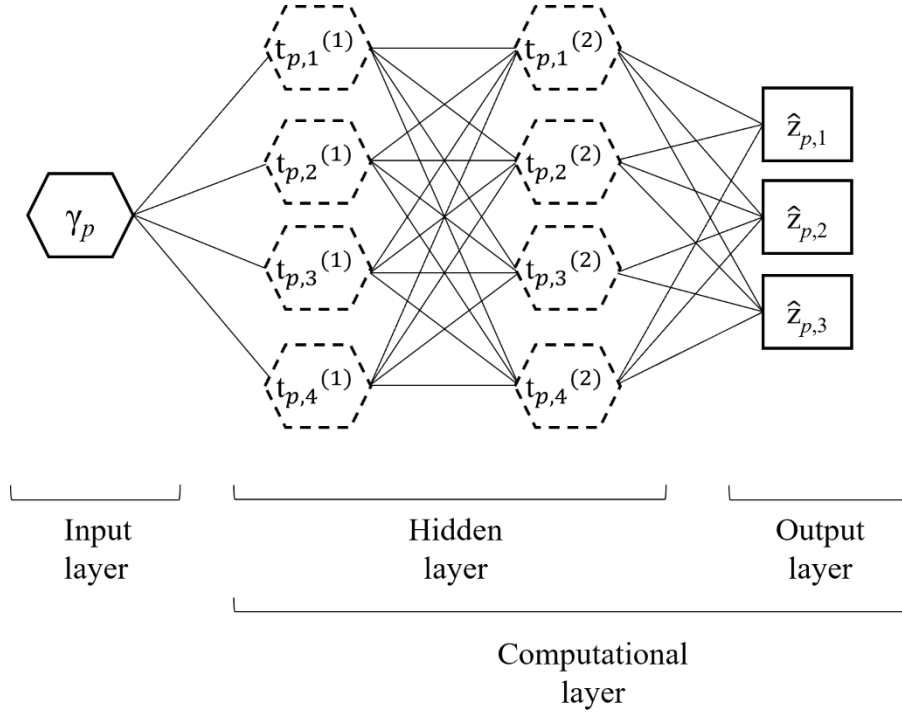


Figure B3. An example of layers for $h_{c,p}$ in DL-GSCA's component measurement model. Dotted circles represent hidden units per hidden layer in the measurement model. Straight lines denote loading parameters.