Online Appendices

The Composite Overfit Analysis Framework: Assessing the Out-of-sample Generalizability of Construct-based Models Using Predictive Deviance, Deviance Trees, and Unstable Paths

Online Appendix A: Predictive Deviants Versus Outliers

We distinguish predictive deviants from outliers and influential cases by comparing these three types of cases. Using the data collected for the empirical demonstration (described in Section 4) we calculate each of the following types of outliers and influential cases and report their comparison in Table A1. First, we calculate predictive deviance, then we identify outlying cases in the X space using *leverage* (calculated on the indicators of exogenous constructs)*;* and *outliers* in the Y space using Mahalanobis distance (calculated on the indicators of endogenous constructs). We report the residual for each case to evaluate the model fit for the individual case, and report Cook's Distance as an indicator of the influence of each case on overall model fit. The table is sorted in descending magnitude according to *PD*, and we highlight and bold the 10% most extreme cases for each method.

Each method for identifying outliers or influential cases places emphasis on different criteria. Mahalanobis distance and leverage measure the distance of each case from the average (here in terms of endogenous and exogenous items, respectively). Evaluating residuals and influence of cases on model parameters identifies cases that have a large impact on model fit (Aguinis et al. 2013). We note that methods for identifying poor model fit are closely related to methods for identifying poor predictive power and so these methods have several cases in common. However, it is important to note that predictive deviance similarly weights cases with poor fit (106, 81), as evidenced by high residuals, as it does cases with better fit (99, 81, 12) which have smaller residuals. Thus, we conclude that predictive deviance is concerned with predictive accuracy and not fit accuracy.

Table A1: Predictive deviance, leverage, outlier, residual, and Cook's distance for Empirical Demonstration in Section 4. Cases sorted by predictive deviance, first 22 (10%) cases shown, bolded cases are in the largest 10% of values.

Case	Predictive Deviance	Leverage (h) (x-space outlier)	Mahalanobis distance (y-space outlier)	Residual (e_{in})	Cook's D (influence)
99	-0.168	**0.399**	2.821	-1.252	**0.054**
106	-0.130	0.156	1.154	**-1.473**	**0.047**
81	0.119	0.151	1.864	**1.390**	**0.041**
12	-0.109	0.153	2.821	-0.738	0.021
187	-0.102	**0.303**	3.021	-0.861	0.022
93	0.102	0.129	3.887	**1.320**	**0.032**
40	-0.098	0.050	1.154	**-2.148**	**0.048**
180	0.094	**0.539**	2.821	0.367	0.016
134	0.092	0.071	3.887	**2.190**	**0.047**
37	0.092	0.225	**14.794**	1.015	0.019
109	0.091	0.137	1.864	**1.487**	**0.032**
151	0.090	0.172	3.887	**1.338**	**0.025**
27	0.089	**0.311**	**17.289**	0.617	0.014
32	-0.078	0.137	2.821	**-1.779**	**0.031**
208	0.074	**0.353**	**37.466**	1.029	0.017
76	-0.068	**0.305**	2.821	-0.723	0.013
69	-0.066	0.100	3.021	-1.176	0.018
110	-0.064	0.161	3.445	-0.673	0.011
153	-0.064	0.131	2.821	-0.955	0.015
63	-0.063	0.134	**12.816**	**-1.391**	0.020
211	0.062	0.209	0.033	0.952	0.013
96	0.061	0.083	3.887	**1.470**	**0.022**

Reference

Aguinis H, Gottfredson RK, and Joo H (2013) Best-practice recommendations for defining, identifying, and handling outliers. *Organizational Research Methods* 16(2):270-301.

Online Appendix B: Unobserved Heterogeneity

We conducted a small study to determine if the predictive deviants detected by our COA framework coincide with heterogenous groups in the sample. Heterogeneity presents another situation that biases parameter estimates and the inferences drawn from them (Rigdon et al. 2010). An example of heterogeneity salient to the management and information systems field is experienced versus inexperienced technology users considered in technology adoption models (Becker et al., 2013).

Multiple solutions for identifying unobserved heterogeneity have been proposed in the literature such as the finite-mixture model (FIMIX; Jedidi et al., 1997) and the prediction-oriented segmentation algorithm (PLS-POS; Becker et al, 2013). These methods have garnered attention within construct-based modeling, with the literature seeing several recent improvements and refinements (Hair et al., 2016; Matthews et al., 2016; Sarstedt et al., 2017). These methods seek to identify two or more subsets of cases within a sample, who form a segment that exhibits behavior that is homogenous within the subset, but heterogenous when compared to that of the sample as a whole.

We seek to distinguish our conceptualization of predictive deviants, which are cases that display moderate fit in-sample but exhibit exceptional out-of-sample error, from the notion of heterogenous cases. There are two important dimensions that differentiate between these two approaches – granularity of analysis, and fit-versus-predictive evaluation (Table B1).

Table B1: Conceptual differences between Unobserved Heterogeneity and Predictive Deviants

Dimension	Unobserved heterogeneity	Predictive Deviants
Granularity	Homogenous group behavior within the subset	Individual behavior compared to entire sample
Predictive consideration	Only the in-sample estimated parameters are considered	Out-of-sample predictive error is evaluated

There are several points of departure between heterogeneity and predictive deviance. To identify unobserved heterogeneity, we must generate substantial subsets (or segments) of cases that display behavior that is homogenous within the subset identified. In contrast, predictive deviants are evaluated on an individual, case-level basis and are expected to form very small groups, if any. Predictive deviants form small groups and even together constitute a small part of the data. In

contrast, when all the different homogenous segments are combined, they entirely encompass the data. Unobserved heterogeneity considers only the estimated in-sample model parameters and the effect of the subsets on these parameters (i.e. in terms of model fit), while predictive deviants consider the out-of-sample performance of the model on each observation (i.e. in terms of prediction accuracy).

To validate these operational and conceptual differences, we conducted a simulation-driven demonstration of the results of identifying predictive deviants versus extant methods for identifying unobserved heterogeneity.

Empirical Demonstration

In order to identify potential unobserved heterogeneity in the dataset collected for the empirical demonstration section of this article (as described in Section 4 of the paper), we apply the procedure first proposed by Becker et al. (2013), and later refined by Sarstedt et al. (2017). We use SmartPLS V3 (Ringle et al., 2015) to run FIMIX-PLS, identifying the number of segments that minimizes BIC and CAIC, while ensuring the Normed Entropic Statistic (EN) is greater than 0.5 (Sarstedt et al. 2017). When we executed this method, it suggested two segments might be present. We then ran the PLS-POS algorithm using the FIMIX-PLS results to allocate the individual cases into two discrete segments - segment 0 with 179 (82.87%) cases and segment 1 with 37 (17.13%) cases. Further, we applied the predictive deviant algorithm results from the empirical demonstration section and allocate each case into a discrete segment (non-predictive deviant / predictive deviant) using 10% cut-off as suggested in the paper. It is important to note that the predictive deviants did not form a segment of the data, but were being treated as such only for demonstration purposes. See Table B2 for a cross-tabulation of the two sets of segments generated by the two algorithms.

Table B2: Cross-tabulation of Segments by FIMIX vs. Predictive Deviants/Non-Predictive Deviants. Values indicate counts.

		PLS-POS Segment		
		Segment 0	**Segment 1**	**Total**
Predictive Deviants	**Non-Deviants**	165	29	194
	Deviants	14	8	22
	Total	179	37	216

The results of this analysis demonstrate that the predictive deviants favor neither group, and thus do not conform to either of the segments determined by the PLS-POS algorithm. The empirical demonstration illustrates that these two algorithms are identifying groups of cases with only slight agreement (80% agreement, Cohen's Kappa = 0.1645), and therefore should not be considered identical.

References

Becker J.M., Rai A., Ringle C.M., and Völckner F (2013). Discovering unobserved heterogeneity in structural equation models to avert validity threats. *MIS quarterly* 37(3):665-694.

Ringle C.M., Wende S., Will A (2005) SmartPLS 2.0.M3. Hamburg: SmartPLS. Retrieved from http://www.smartpls.com.

Rigdon EE, Ringle CM, and Sarstedt M (2010) Structural modeling of heterogeneous data with Partial Least Squares. *Review of marketing research* (Armonk, NY).

Sarstedt M, Ringle CM, and Hair JF (2017). Treating unobserved heterogeneity in PLS-SEM: A multi-method approach. *Partial least squares path modeling* (Springer, Cham).

Online Appendix C: Evaluating the stability of the COA framework under alternate model estimation algorithm

Within the R Statistical Environment there are several implementations of PLS path modeling, with the most notable in terms of longevity and package support being semPLS (Monecke and Leisch, 2012) and matrixpls (Rönkkö, 2016). While matrixPLS provides prediction methods, semPLS does not. In addition, matrixPLS has a different algorithm to SEMinR. We thus apply matrixPLS to our empirical example and find that the fitted and predicted scores are identical to the third decimal. The calculations of subsequent overfit, predictive deviance and application of the deviance tree does not yield different results.

To provide a more rigorous test of the robustness of the framework, we next considered a completely different statistical method for estimating composite models, namely GSCA (Hwang et al., 2004), as implemented in the gesca R package (Hwang et al., 2017). GSCA serves as a natural comparison for PLS when investigating the performance of the framework on estimates derived from different statistical approaches. The PLS-SEM algorithm employs an alternating two-stage least squares algorithm which estimates model parameters by minimizing the model's squared error, while GSCA employs an iterative alternating least squares algorithm which optimizes a single criterion to estimate all parameters concurrently (Hwang et al., 2020). PLSPM is a limited-information method (Tenenhaus, 2008), whereas GSCA is a full-information method.

The differences in estimation algorithms for PLS-SEM and GSCA are fairly substantial, although the conceptual model remains identical. As a result, it is important to note that since overfit is a function of the estimation algorithm, differences should be observed in the results. Different algorithms might minimize the harm of 'deviants' and indeed the *overfit ratio* (equation (3) in main manuscript) might be one way to distinguish or select the estimation algorithm. Using the data collected for the empirical demonstration (described in Section 4) we employ both PLS and GSCA as estimation algorithms to generate fitted and predicted scores and PD for the focal construct BI and then conduct the COA framework on the results. We identify the 5% most extreme predictive deviants (for comparative purposes), calculate the in-sample and out-of-sample MSE, overfit ratio, and then

apply the Deviance Tree to identify deviant groups and individuals. We identify the unstable paths related to these groups of deviants and compare the results.

In Table C1 we find that the prediction metrics for the GSCA model are slightly better in terms of improved fit, predictive power and reduced overfit, although the difference is only half a percentage and might not be meaningful. An analysis of the predictive deviance estimated for the cases in the deviant zone for each model highlights some larger discrepancies. First, the GSCA and PLS algorithms identify slightly different deviant zone cases, with GSCA including case 151 as opposed to PLS that includes case 180. GSCA also generates substantially less extreme values for cases 99, 106, and 180 (with a difference in PD of -.034, -.040, and .045 respectively between PLS and GSCA). A fact which we believe contributes to the lower levels of overfit reflected in the overfit ratio for GSCA. Next, we apply the Deviance Tree to identify groups of cases and identify differences.

Table C1: Predictive metrics, overfit ratio, and the 5% most extreme predictive deviants for the empirical example model using PLS and GSCA

Notes: * denotes predictive deviants shown for comparative reasons but not in the 5% deviant zone. Greyed cells highlight cases with a large difference.

	Case #	PLS	GSCA	PLS-GSCA
$MSE_{(in)}$		0.435	0.433	0.002
$MSE_{(out)}$		0.480	0.475	0.005
Overfit Ratio		0.104	0.099	0.005
Predictive Deviance	**12**	-0.109	-0.111	0.002
	32	-0.078	-0.078	0.000
	37	0.092	0.085	0.007
	40	-0.098	-0.098	-0.000
	81	0.119	0.126	-0.007
	93	0.102	0.103	-0.001
	99	**-0.168**	**-0.134**	**-0.034**
	106	**-0.130**	**-0.090**	**-0.040**
	109	0.091	0.089	0.002
	134	0.092	0.090	0.002
	151	*0.090	0.078	0.012
	180	**0.094**	***0.049**	**0.045**
	187	-0.102	-0.100	-0.002

In Table C2 we see that there is a great deal of overlap in three of the groups (A, B, and E) while the remaining two groups (C, D) are very different. We attribute this result to the different predictive deviance of cases 99 and 106 across algorithms – indeed these two cases now form a group in the GSCA algorithm. The rules for groups A and E are nearly identical despite some changes in the construct score estimates. The rules for group B are very similar, despite one omitted rule in the GSCA algorithm. The balance of the groups and their rules are incomparable. Altogether, the deviant groups reveal similar patterns in the constructs that play the largest role in identifying the groups. Both algorithms identify constructs PE, SI, and HAB as affecting deviants, while GSCA tends to place more emphasis on HAB in determining predictive deviance, while PLS places more emphasis on FC.

Table C2: Deviance Tree applied to estimates from PLS and GSCA algorithms.

	PLS		**GSCA**	
Deviant Group	**Cases**	**Rules**	**Cases**	**Rules**
A	81, 109	BI >= 0.46 SI < -1.63	81, 109	BI > 0.49 SI < -1.60
B	93, 134, 151	BI >= 1.76 HAB < 0.22 SI >= -1.63 EE >= 0.65	93, 96, 134, 151	BI > 1.70 HAB < 0.20 SI > -1.60
C	106, 187	BI < 0.46 -2.45 <= FC < -1.28 HAB > 1	32, 40	HAB > 0.43 EE > -2.90 HM < 0.91 BI < -0.81 PE > -1.00
D	12, 71, 99	BI < -0.53 -2.45 <= FC < -0.93 HAB < 1.00 PE => -1.06	99, 106	HM > 0.91 BE < -0.81 PE > -1.00
E	27, 37, 180	BI < 0.46 FC < -2.45 PE < -1.64	27, 37, 180	BI < 0.49 FC < -2.70 -1.00 > PE > -2.80

Finally, we consider the LDGO analysis applied to these deviant groups. For comparative purposes we look at the stability of the paths when each of the most similar groups (A, B, and E) are removed one-at-a-time from the data (Table C3).

Table C3: Comparison of LDGO analysis applied to deviant groups A, B, and E generated from PLS and GSCA

Notes: In the Original columns we display the original estimated path coefficients and the difference between those of PLS and GSCA. In the subsequent columns for group A, B, and E we show the difference between the original estimate and the estimate generated by the LDGO analysis applied to that group for both PLS and GSCA *and* the difference-in-differences. ΔPLS is the difference between original PLS estimate and LDGO analysis estimate; ΔGSCA is the difference between original GSCA estimate and LDGO analysis estimate; ΔPLS - ΔGSCA is the difference between the two differences.

DV: BI	**Original**			**LDGO Group A (81, 109)**			**LDGO Group B (93, 134, 151)**	**LDGO Group B (93, 96, 134, 151)**		**LDGO Group E (27, 37, 180)**		
	PLS	**GSCA**	**PLS - GSCA**	**ΔPLS**	**ΔGSCA**	**ΔPLS - ΔGSCA**	**ΔPLS**	**GSCA**	**ΔPLS - ΔGSCA**	**ΔPLS**	**GSCA**	**ΔPLS - ΔGSCA**
PE	.172	.187	-.016	-.004	-.002	-.001	.016	**.036**	-.020	.001	-.001	.002
EE	-.094	-.084	-.011	.009	.015	-.006	.016	.013	.003	**-.019**	**-.016**	-.004
SI	.213	.219	-.007	**-.046**	**-.047**	.001	**.041**	**.034**	.007	**.014**	.008	.006
FC	.059	.026	.034	-.008	-.015	.007	.013	.021	-.008	**-.023**	**-.014**	-.009
HM	.210	.210	.000	-.006	-.006	.000	**-.026**	**-.030**	.004	-.003	-.001	-.001
PV	.008	.010	-.002	-.009	-.009	.000	.011	.027	-.016	-.010	-.009	-.001
Hab	.292	.286	.007	**.051**	**.051**	.000	**-.047**	**-.066**	.020	.003	.005	-.002
Exp	-.076	-.082	.005	.007	.008	-.001	.003	.015	-.012	.006	.005	.001
Age	.069	.072	-.003	-.013	-.013	.000	-.008	-.001	.002	-.007	-.006	-.002
Gen	.006	.005	.001	.018	.018	.000	-.005	-.012	.007	.007	.004	.002

First, we consider group A consisting of cases 81 and 109. When removed from the data both PLS and GSCA identify the paths SI and Hab as being most affected. Group B consisting of cases 93, 134, 151 (and 96 for GSCA) when removed from the data for both PLS and GSCA also identify SI and Hab, but in addition identify HM. Overall, the LDGO analysis is fairly robust when comparing the performance on the GSCA algorithm to that of PLS.

Overall, our conclusion is that despite the fundamental differences in estimation algorithm and optimization criterion, the COA framework does well to robustly identify deviant cases, deviant groups, and paths that need further investigation. An additional insight for future research might be to investigate the potential for GSCA to perform slightly better in terms of overfit.

References:

Hwang, H., Kim, S., Lee, S., and Park, S. (2017). gesca: Generalized Structured Component Analysis (GSCA). R package version 1.0.4. https://CRAN.R-project.org/package=gesca

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.

Monecke, A., & Leisch, F. (2012). semPLS: structural equation modeling using partial least squares.

Rönkkö, Mikko (2021). matrixpls: Matrix-based Partial Least Squares Estimation. R package. version 1.0.13.

Tenenhaus M (2008) Component-based structural equation modelling. *Total Quality Management* 19(7–8):871–886

Online Appendix D: Evaluating the stability of the COA framework under slight sample size modification

Using the data and model from the empirical demonstration (described in Section 4), we now investigate the stability of the COA framework when the sample size is slightly varied (recall that large sample size changes can, and are expected to, affect overfit). Specifically, we modify the sample size by omitting 5% (n = 11) of the cases randomly. We considered also increasing the sample size by randomly adding 5% of the cases, but this results in duplication of deviant cases and does not provide evidence that deviant cases would be identified consistently in different configurations. We simulate a sample size change by first removing the 12 predictive deviants, then randomly sampling 193 cases from the remaining non-deviant cases, then adding back the 12 deviants such that we have 95% (n = 205) of the original data and retain all predictive deviants. This ensures that all the original cases in the deviant zone are present in each subsample. We then calculate the prediction metrics and predictive deviance for the subsample, perform the deviance tree to identify subsample specific groups, and then run the LDGO analysis. We conduct 500 replications of the simulation. The results are shown in Figure D1.

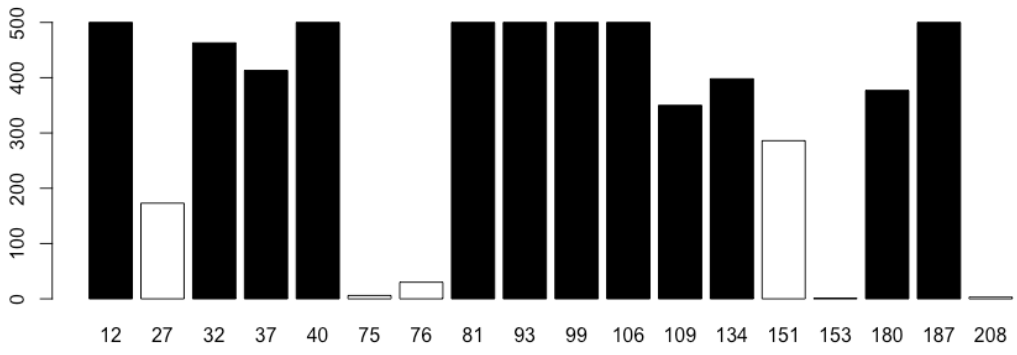

Figure D1: Number of times the deviants are identified in the subsets

Notes: Black columns are deviants identified in the 5% deviant zone in the original dataset, while white are not identified.

We observe that the cases identified in the original 5% deviant zone are identified in between 70% (case 109, Figure D1) and 100% (cases 12, 40, 81, 93, 99, 106, and 187) of the subsets in which they

occur. Interestingly there are two cases (27 and 151) which are identified in 34.6% and 57.2% of the subsets respectively but are not identified as extreme deviants or included in deviant groups in the original sample. This indicates that in some subsets these cases are relatively more poorly predicted versus fitted (or predictively deviant). However, the focal predictive deviants are reliably identified across subsets.

Next, we consider the deviant groups identified by the deviance tree algorithm across subsets. The most salient finding is that the five groups most often identified in the subsets are near-identical to those identified in the entire sample (Table C2, column 1 and 2). Group D (cases 12, 71, and 99; Table D1) is grouped only in 51% of subsets, while Group A (cases 81, and 109) and Group B (cases 93, 134, 151) show up in a very robust 97% and 81% of the subsets. Further, Group C (cases 106 and 187) and Group E (cases 27, 37, and 180) are grouped in only 47% and 41% of subsets respectively. However, the cases in these groups occur in several combinations: (1) cases 27 and 37 (51 times); (2) cases 37 and 180 (95 times). The regular grouping of these cases indicates their similarity and the validity of the groups.

Table D1: Deviant group distribution: number of times the deviant groups were identified in subsamples, and the group names
Notes: Due to the long-tail effect, only those results in excess of 50 (10% of 500 samples) times shown

Deviant Group	**27, 37**	**40, 106**	**37, 180**	**27, 37, 180**	**106, 187**	**12, 71, 99**	**93, 134, 151**	**81, 109**
Group Name				E	C	D	B	A
Times grouped	51 (10.2%)	85 (17%)	95 (19%)	207 (41.4%)	235 (47%)	257 (51.4%)	406 (81.2%)	487 (97.4%)

Finally, we consider the robustness of the impact of the deviant groups on the unstable paths. We calculate the change in the paths due to LDGO analysis for each of the n times the group was observed in the subsets. In each subset for which group A, B, or E was identified, we calculated the difference in the structural paths' coefficients. We then calculated the mean of these differences in path coefficients, and the standard deviation (Table D2). The impact of the deviant groups in subsamples is on average very similar to that observed in the original sample. We conclude that the framework, and the results identified therefrom are robust across small variations to sample size.

Table D2: Mean Diff and standard deviation of difference for LDGO analysis applied in subsets

DV: BI	Original Estimate	LDGO Group A (n = 487)		LDGO Group B (n = 406)		LDGO Group E (n = 207)	
		Mean Diff	SD	Mean Diff	SD	Mean Diff	SD
PE	.172	-.004	.002	.017	.002	.001	.002
EE	-.094	.009	.002	.017	.002	-.020	.002
SI	.213	-.048	.002	.043	.001	.015	.002
FC	.059	-.008	.002	.013	.002	-.025	.002
HM	.210	-.006	.002	-.027	.002	-.003	.001
PV	.008	-.010	.002	.011	.001	-.010	.001
Hab	.292	.054	.007	-.049	.002	.004	.001
Exp	-.076	.008	.001	.003	.001	.006	.001
Age	.069	-.014	.001	-.008	.001	-.008	.001
Gen	.006	.019	.001	-.005	.001	.007	.001

Online Appendix E: Evaluating the stability of the COA framework under slightly modified models (dropping control variables)

The UTAUT model applied in our empirical example includes three control variables which are not subject to hypothesis. We now evaluate the impact of these variables on the prediction metrics and the COA framework. To conduct this analysis, we iteratively remove a control variable and rerun the COA framework on the model and data collected for the empirical demonstration (described in Sec. 4.1 and 4.2). In Table E1 we report the prediction metrics, overfit ratio, deviants identified as most extreme, the subsequent groups and the LDGO analysis to explore the impact of these deviant groups on the paths.

Table E1: Prediction metrics, overfit ratio and PD for models iteratively excluding the control variables.
Notes: Greyed rows indicate cases with the largest change in predictive deviance.

	Case #	PLS	Excluding Gender		Excluding Experience		Excluding Age	
		Original Estimate	Estimate	Diff	Estimate	Diff	Estimate	Diff
$MSE_{(in)}$		0.435	0.437	-0.002	0.441	-0.005	0.440	-0.004
$MSE_{(out)}$		0.480	0.476	0.004	0.483	-0.003	0.480	0.000
Overfit Ratio		0.104	0.090	0.005	0.096	0.008	**0.093**	**0.011**
Predictive Deviance	**12**	-0.109	-0.108	-0.001	-0.117	0.008	-0.116	0.007
	32	-0.078	-0.074	-0.004	-0.078	0.000	-0.072	-0.007
	37	0.092	0.089	0.003	0.085	0.006	0.089	0.003
	40	-0.098	**-0.078**	**-0.020**	-0.098	0.000	**-0.060**	**-0.038**
	76	-0.068	-0.065	-0.003	-0.069	0.001	**-0.039**	**-0.029**
	81	0.119	0.110	0.009	0.115	0.004	0.111	0.008
	93	0.102	0.092	0.010	0.103	-0.002	0.103	-0.001
	99	-0.168	-0.159	-0.009	-0.171	0.003	-0.166	-0.002
	106	-0.130	-0.121	-0.009	-0.132	0.002	-0.130	0.000
	109	0.091	0.081	0.009	0.087	0.004	0.082	0.009
	134	0.092	0.086	0.006	**0.071**	**0.021**	0.092	0.000
	151	0.090	0.086	0.005	0.087	0.004	0.084	0.006
	153	-0.064	-0.054	-0.010	-0.065	0.001	-0.065	0.001
	180	0.094	*0.091	0.003	0.101	-0.007	0.101	-0.007
	187	-0.102	-0.096	-0.006	**-0.045**	**-0.057**	-0.092	-0.010

Omitting the control variables results in a slightly lower overfit ratio in all three models. This is not surprising because typically the addition of variables with small or non-significant relationships can lead to improved model fit, but can damage predictive accuracy (Hastie et al., 2013). Interestingly the

omission of the control variables has a much larger impact on the individual cases - reducing the predictive deviance of cases 40, 76, 134, and 187 substantially (Table E1). This indicates that in some cases the contribution of the control variable to the prediction is responsible for the extreme predictive deviance. These cases might be further investigated to identify if they have extreme or unusual values for the control variables and if they are valid cases for the study.

When we compare the grouping of the predictive deviants, as shown in Table E2, the impact of the control variables on the predictive deviance of individual cases is more pronounced. We see that groups C, D, and E remain largely similar and share approximately similar deviance tree rules. However, groups A and B are substantially different in both their cases and in the deviance tree rules defining the group. The LDGO analysis for these similar groups C, D, and E remains consistent with prior LDGO analyses. We thus focus on these new groups: in Table E3 we examine the LDGO analysis applied to groups A and B of the model omitting the Exp control, and in Table E4 we examine the LDGO analysis omitting the Age control.

Table E2: Deviance Tree applied to various models excluding control variables

Group	Original		No Gen		No Exp		No Age	
	Cases	**rules**	**Cases**	**rules**	**Cases**	**rules**	**Cases**	**rules**
A	106, 187	BI < 0.46 -2.45 <= FC < -1.28 HAB > 1	106, 187	BI < 0.46 FC >= -2.45 HAB >= 0.10 FC < -1.28	12, 99, 106	BI < -0.54 PE >= -0.57 FC < - 0.93	99, 106	BI< 0.46 -1.03 > FC >= -2.45 HM >= 1.10
B	12, 71, 99	BI < -0.53 -2.45 <= FC < -0.93 HAB < 1.00 -1.06 <= PE	12, 99	BI < 0.46 -0.93 > FC >= -2.45 HAB < 0.1 PV < -2.44	32, 40, 63	BI < -0.54 PE >= -0.57 FC >= -0.93 HAB >= 0.43	32, 62, 153	HM < 1.10 EE >= -3.40 HAB >= -0.78 BI < -1.14 FC >= 0.85
C	27, 37, 180	BI < 0.46 FC < -2.45 PE < -1.64	37, 180	BI < 0.46 FC < -2.45 PE < -2.03	37, 180	BI < -0.54 FC < -2.45 PE < -2.03	37, 180	BI < 0.46 FC < -2.45 PE < -2.03
D	93, 134, 151	1.76 <= BI HAB < 0.22 -1.63 <= SI 0.65 <= EE	93, 134, 151	SI> = -1.63 HAB < 0.22 BI >= 1.76 HM < 0.62	93, 134, 151	HAB < 0.22 BI >= 1.46 HM < 0.62	93, 96, 134, 151	SI >= -1.63 HAB < 0.22 BI >= 1.76
E	81 and 109	BI >= 0.46 SI < -1.63	81, 109	BI >= 0.46 SI < -1.63	81, 109	BI >= - 0.54 HAB >= 0.22 SI < -1.63	81, 109	BI >= 0.46 SI < -1.63

Table E3: Diff for LDGO analysis applied in models excluding Exp control variable

DV: BI	Original	LDGO Group A Diff	LDGO Group B Diff
PE	0.166	-0.037	-0.023
EE	-0.102	-0.026	0.001
SI	0.217	0.020	0.006
FC	0.061	0.057	-0.001
HM	0.211	-0.029	0.007
PV	0.009	0.023	-0.010
Hab	0.297	0.008	-0.018
Age	0.059	0.013	-0.001
Gen	0.007	0.004	0.026

Table E4: Diff for LDGO analysis applied in models excluding Age control variable

DV: BI	Original	LDGO Group A Diff	LDGO Group B Diff
PE	0.153	-0.026	-0.035
EE	-0.107	-0.042	-0.004
SI	0.212	0.027	0.036
FC	0.068	0.059	-0.031
HM	0.220	-0.026	0.017
PV	0.011	0.014	0.018
Hab	0.306	-0.003	-0.011
Exp	-0.068	-0.003	-0.004
Gen	0.018	0.002	-0.004

The LDGO analysis applied to group A in both the model excluding the Exp control variable and that excluding the Age control variable indicates path instability in the majority of the variables with hypotheses. These groups are similar, including cases 99 and 106 and this provides evidence that cases 99 and 106 should be more deeply investigated to determine the reasons for their influence on these paths.

The control variables serve an important purpose in the testing of theoretical hypotheses and so cannot simply be dropped. However, researchers might need to pay more attention to the necessity for and importance of such constructs to their research given the impact on overfit. Further, researchers might also apply the LDGO analysis excluding the control variables. Such an investigation will

eliminate the effect of the control variables on the predictive deviance and allow the researcher to evaluate the impact of these cases on the focal paths under hypotheses.

References:

Hastie T, Tibshirani R, and Friedman JH (2013) *The elements of statistical learning: Data mining, inference, and prediction* 2nd ed. (Springer, New York, NY).

Online Appendix F: Evaluating the stability of the COA framework under alternate deviance tree estimation algorithms

There are three generally used implementations of classification and regression trees in R. These are *rpart* (Therneau & Atkinson, 2019; over 4 million downloads), *tree* (Ripley, 2019; approximately 1.1 million downloads), and *party* (Hothorn et al., 2006; approximately 2.5 million downloads). These three implementations have several distinctions in terms of the splitting criteria and partitioning algorithm. *Rpart* and *tree* make use of binary recursive partitioning using the Gini coefficient or a deviance metric as splitting criterion (Therneau & Atkinson, 2019; Ripley, 2019). In contrast, *Party* conducts partitioning by means of conditional inference (Hothorn et al., 2006). Due to the very different nature of the spitting criteria in *rpart* and *tree* versus *party*, we anticipated that *party* would build a very different tree. Nonetheless, we evaluate the performance of these three alternate algorithms for building our deviance trees.

To compare the three algorithms for creating the tree, we inspect (1) the resulting deviant groups; and (2) the rules associated with the groups when applied to the model and data collected for the empirical demonstration (described in Sec. 4). We follow the same conceptual method for building the deviance tree as discussed in Section 3.2.

Table F1 shows the deviant groups and deviance tree rules for the three algorithms. The first finding is that the five deviant groups identified by each deviance tree algorithm are very similar. Those groups identified by *rpart* and *tree* are identical in terms of cases, but slightly differ in the rules that generate the groups. Specifically, for group C *tree* selects HM while *rpart* favors PE. Similarly, for group D *tree* again favors HM, while *rpart* favors EE. *rpart* yields nearly identical cases in groups C, D, and E but includes the additional case 96 in D, and 208 in C. The composition of Groups A and B is different in *party* versus *rpart* and the rules are not directly comparable. Overall, we find that the groups remain very consistent when we compare the two more commonly employed algorithms - r*part* and *tree*, and remain fairly robust even when the tree building method is very different as seen in *party*.

Since *rpart* is the most widely used R package for trees, and is continuously maintained and updated, we continue to use this package for building the deviance tree and the COA Framework.

Table F1: Deviant groups and deviance tree rules for the three algorithms building the deviance tree

Notes: Bolded rules highlight differences across algorithms

Group	rpart		tree		party	
	Cases	**Rules**	**Cases**	**Rules**	**Cases**	**Rules**
A	106, 187	BI < 0.46 -2.45 <= FC < -1.28 HAB >= 1.00	106, 187	BI < 0.46 -2.5 < FC < -1.30 HAB > 1.00	**63** and 187	-0.65 < BI < 0.37 HAB > 1.00 **EXP < -1.32** **PE > -0.67**
B	12, 71, 99	BI < -0.53 -2.45 <= FC < -0.93 HAB < 1.00 PE >= -1.06	12, 71, 99	BI < -0.53 -2.45 < FC < -0.93 HAB < 1.00 PE > -1.06	99, **106, 166**	BI < -0.65 HM > 0.59 PE > -0.67
C	27, 37, 180	BI < 0.46 FC < -2.45 PE < -1.64	27, 37, 180	BI < 0.46 FC < -2.45 **HM < 2.04**	27, 37, 180, **208**	BI < 0.372 FC < -2.75 PE < -0.67 **HAB < -0.83**
D	93, 134, 151	BI >= 1.76 HAB < 0.22 SI >= -1.63 EE >= 0.65	93, 134, 151	BI > 1.76 HAB < 0.22 SI > -1.6 **HM < 0.6**	93, **96**, 134, 151	BI > 1.57 HAB < 0.19
E	81, 109	BI >= 0.46 SI < -1.63	81, 109	BI > 0.46 SI < -1.6	81, 109	0.37 < BI < 1.57 SI < -2.00

References:

Hothorn, T., Hornik, K., and Zeileis, A. (2006). Unbiased Recursive Partitioning: A Conditional Inference Framework. *Journal of Computational and Graphical Statistics*, 15(3), 651--674.

Ripley, B. (2019). tree: Classification and Regression Trees. R package version 1.0-40. https://CRAN.R-project.org/package=tree

Therneau, T., & Atkinson, B. (2019). rpart: Recursive Partitioning and Regression Trees. R package version 4.1-15. https://CRAN.R-project.org/package=rpart