Supplementary Material for "Panel Smooth Transition Model with Covariate-Dependent Thresholds and Its Application to the Nexus Between Investment and Cash Flow", *Studies in Nonlinear Dynamics and Econometrics, 2025.*

This supplementary material consists of 7 sections: SM.A provides simulation results of parameter estimation for the proposed model with the lagged dependent variable being the transition variable. SM.B provides simulation results comparing the performance of asymptotic and bootstrap confidence intervals. SM.C shows simulation results for the comparison of various optimization algorithms. SM.D presents simulation results for variable selection based on mixed integer optimization algorithms. SM.E provides simulation results for the proposed model with two covariate-dependent thresholds. SM.F presents simulation results for the parameter regions where the F-type tests perform poorly. SM.G reports simulation results on the performance of variable selection when a time trend is included as a covariate.

SM.A: Simulation Results of Parameter Estimation for the Model with the Lagged Dependent Variable Being the Transition Variable

In this section, we conduct a Monte Carlo simulation to evaluate the finite sample performance of the proposed estimation procedure for the model with lagged dependent variable being the transition variable. To do so, we consider the following data generating process (DGP):

$$y_{it} = \beta_1 x_{it} + \beta_2 x_{it} g(y_{i,t-1}; k, \gamma_{it}) + \beta_3 z_{it} + \alpha_i + u_{it}, \tag{SM.1}$$

$$g(y_{i,t-1}; k, \gamma_{it}) = (1 + \exp(-k(y_{i,t-1} - \gamma_{it})))^{-1}, k > 0, \tag{SM.2}$$

$$\gamma_{it} = \gamma_0 + \gamma_1 s_{it}, \tag{SM.3}$$

where $x_{it} = 1 + 0.25\alpha_i + u_{x,it}$, $z_{it} = 0.5\alpha_i + u_{z,it}$, $\alpha_i \overset{i.i.d}{\sim} N(0,1)$, $s_{it} \overset{i.i.d}{\sim} N(2,1)$, $u_{it} \overset{i.i.d}{\sim} N(0, 0.5^2)$, $u_{x,it} \overset{i.i.d}{\sim} N(0,1)$ and $u_{z,it} \overset{i.i.d}{\sim} N(0,1)$. The innovation processes $u_{x,it}$ and $u_{z,it}$ are independent of each other. Following Ramírez-Rondán (2020), the data for the variables are generated from $t = -10$ to $t = T$. To generate y_{it}, we initialize $y_{i,t-10} = 0$. For the estimation, we discard the first 10 observations, retaining the observations from $t = 0$ to $t = T$. The number of replications is set to 1000.

The true parameters are set as $(k, \gamma_0, \gamma_1) = (5, 0.2, 0.5)$ and $(\beta_1, \beta_2, \beta_3) = (1, 2, 1)$. The simulation results are reported in Table SM.1, including the mean and standard deviation for each parameter. These results indicate that the means of parameters are very close to

Table SM.1: Estimates of the parameters for the model with $y_{i,t-1}$ being the transition variable.

T	N	$\beta_1 = 1$		$\beta_2 = 2$		$\beta_3 = 1$		$k = 5$		$\gamma_0 = 0.2$		$\gamma_1 = 0.5$	
		Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
5	50	1.013	0.052	1.977	0.066	1.000	0.036	5.722	2.799	0.194	0.135	0.503	0.061
	100	1.014	0.035	1.976	0.046	1.000	0.024	5.289	0.905	0.199	0.079	0.501	0.035
	200	1.011	0.026	1.978	0.032	0.999	0.017	5.182	0.527	0.197	0.050	0.501	0.023
10	50	1.005	0.035	1.990	0.042	0.999	0.025	5.156	0.700	0.198	0.072	0.500	0.033
	100	1.005	0.024	1.989	0.029	0.999	0.016	5.094	0.492	0.200	0.048	0.500	0.022
	200	1.006	0.016	1.989	0.021	0.999	0.012	5.079	0.348	0.200	0.033	0.500	0.014
20	50	1.005	0.024	1.992	0.030	1.000	0.016	5.065	0.473	0.199	0.047	0.500	0.022
	100	1.003	0.017	1.995	0.022	1.000	0.011	5.062	0.344	0.201	0.031	0.499	0.014
	200	1.003	0.012	1.995	0.015	1.000	0.008	5.019	0.234	0.199	0.022	0.501	0.010

the true values across all sample sizes, and the standard deviations decrease to zero with the sample size. The bias and standard deviation of the parameter estimation for the smoothness parameter are slightly larger than other parameters, as discussed in Section 2. The simulation results demonstrate that the proposed estimation procedure works well in the model with lagged dependent variable being the transition variable.

SM.B: Bootstrap Confidence Intervals for Parameters

In this section, we construct confidence intervals for model parameters using a bootstrap method, which is suitable as it does not depend on any assumption regarding population distributions and covariance structures (e.g. Nevitt and Hancock 2001). Following the threshold literature (e.g. Hansen 1999, 2017), our bootstrap procedure is given as follows.

Algorithm G. *Confidence intervals for parameters*

Step 1. Use the original sample $(y_{it}, \boldsymbol{x}'_{it}, q_{it}, \boldsymbol{s}'_{it})$'s to estimate model (1), obtain $(\hat{\boldsymbol{\beta}}, \hat{k}, \hat{\boldsymbol{\gamma}})$ and the residual $\hat{\varepsilon}_{it} = \hat{u}_{it} - \bar{\hat{u}}_i$, where $\hat{u}_{it} = y_{it} - \hat{\boldsymbol{\beta}}'\boldsymbol{x}_{it} g(q_{it}; \hat{k}, \hat{\gamma}_{it})$, $\bar{\hat{u}}_i = \frac{1}{T}\sum_{t=1}^{T} \hat{u}_{it}$.

Step 2. Generate $i.i.d$ draws u^*_{it} from the $N(0,1)$ distribution for $i = 1, ..., N$ and $t = 1, ..., T$, and set $\varepsilon^*_{it} = \hat{\varepsilon}_{it} u^*_{it}$ and $y^*_{it} = \hat{\boldsymbol{\beta}}'\boldsymbol{x}_{it} g(q_{it}; \hat{k}, \hat{\gamma}_{it}) + \bar{\hat{u}}_i + \varepsilon^*_{it}$.

Step 3. Use the observations $(y^*_{it}, \boldsymbol{x}'_{it}, q_{it}, \boldsymbol{s}'_{it})$'s to estimate model (1), obtaining the parameter estimates $(\hat{\boldsymbol{\beta}}^*, \hat{k}^*, \hat{\boldsymbol{\gamma}}^*)$ and $SSR^*_{NT}(\hat{k}^*, \hat{\boldsymbol{\gamma}}^*) = \frac{1}{NT}(\ddot{\boldsymbol{Y}} - \ddot{\boldsymbol{X}}(\hat{k}^*, \hat{\boldsymbol{\gamma}}^*)\hat{\boldsymbol{\beta}}(\hat{k}^*, \hat{\boldsymbol{\gamma}}^*))'(\ddot{\boldsymbol{Y}} - \ddot{\boldsymbol{X}}(\hat{k}^*, \hat{\boldsymbol{\gamma}}^*)\hat{\boldsymbol{\beta}}(\hat{k}^*, \hat{\boldsymbol{\gamma}}^*))$.

Step 4. Repeat Steps 2 and 3 B times, and obtain the estimates for simulated sample $(\hat{\boldsymbol{\beta}}^*, \hat{k}^*, \hat{\boldsymbol{\gamma}}^*)$.

Step 5. Then, we can use percentile interval method to construct the percentile bootstrap $1 - \alpha$ confidence interval for each parameter. For example, for γ_0, the confidence interval is $[\hat{\gamma}^*_{\alpha/2}, \hat{\gamma}^*_{1-\alpha/2}]$, where $\hat{\gamma}^*_{\alpha/2}$ and $\hat{\gamma}^*_{1-\alpha/2}$ are the $\alpha/2$ and $1 - \alpha/2$ quantiles of the bootstrap distribution of $\hat{\gamma}^*_0$ respectively.

We next conduct a Monte Carlo simulation to compare the performance of asymptotic

Table SM.2: Performance of asymptotic and bootstrap confidence intervals.

T	N	Method	β_1		β_2		β_3		k		γ_0		γ_1	
			CP	AL	CP	AL	CP	AL	CP	AL	CP	AL	CP	AL
2	10	Bootstrap	0.549	0.490	0.517	0.274	0.527	0.437	0.557	7.733	0.537	0.169	0.504	0.164
		Asymptotic	0.642	0.618	0.609	0.314	0.637	0.552	0.582	6.105	0.545	0.136	0.534	0.136
	20	Bootstrap	0.690	0.309	0.692	0.167	0.680	0.292	0.710	1.758	0.656	0.068	0.622	0.060
		Asymptotic	0.849	0.455	0.842	0.246	0.840	0.426	0.848	2.537	0.807	0.098	0.784	0.087
	40	Bootstrap	0.776	0.217	0.771	0.120	0.765	0.205	0.724	1.105	0.756	0.048	0.679	0.041
		Asymptotic	0.905	0.315	0.910	0.175	0.914	0.299	0.898	1.607	0.890	0.070	0.817	0.060
5	10	Bootstrap	0.835	0.275	0.861	0.152	0.858	0.261	0.853	1.469	0.841	0.061	0.795	0.054
		Asymptotic	0.849	0.295	0.865	0.164	0.851	0.278	0.861	1.539	0.828	0.065	0.789	0.056
	20	Bootstrap	0.887	0.199	0.912	0.109	0.877	0.189	0.895	0.998	0.883	0.044	0.868	0.039
		Asymptotic	0.914	0.221	0.932	0.121	0.901	0.210	0.919	1.089	0.910	0.049	0.891	0.042
	40	Bootstrap	0.903	0.142	0.905	0.078	0.904	0.134	0.916	0.705	0.872	0.031	0.878	0.027
		Asymptotic	0.933	0.160	0.940	0.087	0.937	0.152	0.944	0.786	0.907	0.035	0.907	0.030
10	10	Bootstrap	0.905	0.199	0.909	0.109	0.904	0.188	0.907	1.001	0.896	0.044	0.873	0.038
		Asymptotic	0.902	0.200	0.901	0.111	0.894	0.190	0.885	0.992	0.888	0.045	0.851	0.038
	20	Bootstrap	0.923	0.142	0.913	0.078	0.911	0.134	0.920	0.698	0.906	0.031	0.893	0.027
		Asymptotic	0.919	0.147	0.918	0.081	0.928	0.140	0.919	0.725	0.907	0.033	0.904	0.028
	40	Bootstrap	0.941	0.100	0.910	0.055	0.922	0.096	0.916	0.498	0.929	0.022	0.920	0.019
		Asymptotic	0.949	0.106	0.927	0.058	0.934	0.101	0.941	0.525	0.946	0.024	0.931	0.020

and bootstrap confidence intervals, using the same DGP as in Section 5.1. Table SM.2 reports empirical coverage probabilities (CPs) and average lengths (ALs) of 95 % confidence intervals for parameters, based on 1000 replications. As shown in Table SM.2, the asymptotic method generally achieves coverage probabilities closer to the nominal 95 % level for most parameters and sample sizes. However, it tends to yield wider intervals, particularly in smaller samples (e.g. $T = 2, N = 10$), as evidenced by higher AL values. In contrast, the bootstrap method generates shorter intervals at the expense of lower coverage probabilities in smaller samples. When the sample size increases (e.g. $T = 10, N = 40$), the asymptotic method maintains higher CP values while producing intervals of comparable or even shorter lengths than the bootstrap method, demonstrating the superiority of asymptotic confidence intervals. The bootstrap method exhibits higher CP values as the sample size increases, approaching the nominal 95 % level, which suggests that the bootstrap could be a viable alternative. In conclusion, we recommend prioritizing the asymptotic method in practical applications due to its higher coverage probabilities, acceptable interval lengths, and computational efficiency (by circumventing the resampling burden of the bootstrap).

SM.C: Simulation Results for the Comparison of Various Optimization Algorithms

In this section, we conduct a Monte Carlo simulation to evaluate the performance of the proposed simulated annealing algorithm (SA) against seven optimization methods: Broyden–Fletcher–Goldfarb–Shanno (BFGS), Gauss-Newton (GN), Simplex, Genetic Algorithm (GA), Differential Evolution (DE), Conjugate Gradient (CG), and Markov chain Monte Carlo (MCMC).

We provide a brief description of the algorithms mentioned above. (1) Broyden–Fletcher–Goldfarb–Shanno (BFGS): The BFGS is a quasi-Newton algorithm for unconstrained optimization, which approximates the Hessian matrix to update search directions. It has been applied in parameter estimation for state-dependent threshold Smooth Transition Autoregressive (STAR) models (e.g. Dueker et al. 2013). (2) Gauss-Newton (GN): The GN algorithm is a modification of Newton's method for nonlinear least squares problems. It approximates the Hessian using the Jacobian matrix, avoiding direct computation of second derivatives (e.g. Osborne 2007). (3) Simplex: As a derivative-free method, the Simplex has been applied to parameter estimation in the STR-type model (e.g. Omay and Emirmahmutoğlu 2017). As demonstrated by Omay and Emirmahmutoğlu (2017), the Simplex method outperforms gradient-based methods (e.g. BFGS) in obtaining accurate critical values for nonlinear unit root tests. (4) Genetic algorithm (GA): It is a heuristic optimization technique inspired by the principles of natural selection and heredity, and this technique has been applied to estimation and modeling problems (e.g. El-Shagi 2011). The algorithm begins with an initial population of candidate solutions. Through iterative processes of selection, crossover, and random mutation, new solutions are generated and evaluated based on a fitness function (objective function). The population evolves until convergence criteria are satisfied. (5) Differential Evolution (DE): The DE method is a population-based heuristic similar to the Genetic Algorithm (GA), but it requires simpler parameter tuning (e.g. Hegerty, Hung, and Kasprak 2009). Moreover, DE outperforms GA in many numerical single objective optimization problems (e.g. Tušar and Filipič 2007). (6) Conjugate Gradient (CG): The CG method is an iterative algorithm for minimizing unconstrained nonlinear functions. It requires minimal computer storage, making it particularly suitable for large-scale optimization problems. (7) Markov chain Monte Carlo (MCMC): MCMC is a Bayesian sampling technique that generates parameter samples from posterior distributions. Recent studies (e.g. Yu and Fan 2021; Yang 2024; Yang et al. 2024) propose MCMC as an alternative to grid search. This approach reduces computational cost when estimating models with numerous parameters.

The DGP follows the PSTCT model, with parameters identical to those in Section 5.1. The number of replications is set to 1000, and the simulation results are reported in Table SM.3. The criteria used for comparison include the means and standard deviations of the parameter estimates and the average computational time (in seconds) of 1000 estimations for each algorithm. To make the comparisons more intuitive, we use a relatively large sample size $(T, N) = (20, 200)$.

Table SM.3 presents simulation results for the different algorithms including the means and standard deviations for the parameters $\boldsymbol{\beta}$, k and $\boldsymbol{\gamma}$, along with the time consumption for estimation. As shown in Table SM.3, the estimates of parameter k obtained from various algorithms exhibit sizable differences. The SA, Simplex, GA, DE algorithms yield better results, while the BFGS, GN, CG and MCMC methods perform poorly. Specifically, among the better-performing algorithms, SA is the best algorithm in terms of estimation accuracy and time consumption. The Simplex method is fastest, but the standard deviation of k is slightly larger than that of SA. The means and standard deviations obtained by the GA and DE are close to those of SA, albeit with longer time consumption. Moreover, in terms of time consumption, the SA, BFGS and Simplex can obtain the estimation results within a few seconds. The GA and DE require dozens of seconds. The GN takes the longest time, taking several hundred seconds.

Overall, in terms of the accuracy in parameter estimation and the time consumption, the simulated annealing (SA) and Simplex are optimal methods in this optimization problem, while the Gauss-Newton (GN), Conjugate Gradient (CG) and Markov chain Monte Carlo (MCMC) perform poorly. The Differential Evolution (DE) approach outperforms the Genetic Algorithm (GA) in terms of estimation precision and time consumption, as confirmed by Tušar and Filipič (2007). Taken together, the simulation results indicate that the proposed simulated annealing algorithm is more efficient both in parameter estimation and in time

Table SM.3: Performance comparison among various algorithms

Method	Time	$\beta_1 = 1$		$\beta_2 = 2$		$\beta_3 = 1$		$k = 5$		$\gamma_0 = 0.2$		$\gamma_1 = 0.5$	
		Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
SA	1.364	1.000	0.008	2.000	0.005	1.001	0.008	5.000	0.041	0.200	0.002	0.500	0.002
BFGS	2.903	1.045	0.068	1.910	0.131	1.000	0.012	11.448	10.755	0.200	0.013	0.498	0.014
GN	430.619	1.164	0.032	1.673	0.045	1.001	0.020	47.526	5.913	0.200	0.013	0.497	0.011
Simplex	1.333	1.000	0.008	2.000	0.005	1.001	0.008	5.000	0.046	0.200	0.003	0.500	0.002
GA	56.030	1.000	0.009	2.000	0.006	1.001	0.008	5.002	0.079	0.200	0.003	0.500	0.003
DE	26.079	1.000	0.008	2.000	0.005	1.001	0.008	5.000	0.041	0.200	0.002	0.500	0.002
CG	3.840	1.168	0.023	1.666	0.011	1.001	0.021	49.841	1.417	0.200	0.013	0.498	0.012
MCMC	23.228	1.147	0.052	1.707	0.091	1.001	0.019	38.390	17.361	0.200	0.027	0.496	0.026

consumption. It exhibits both high estimation accuracy and low computational time.

SM.D: Simulation Results for Variable Selection Based on Mixed Integer Optimization Algorithms

In this section, we proposed two variable selection methods based on mixed integer optimization (MIO) as alternatives to the LASSO approach. As suggested by Bertsimas, King, and Mazumder (2016), the classical best subset selection problem can be formulated as an MIO problem, which can significantly improve computational efficiency. Consequently, we first adopt an MIO-based method as an alternative to LASSO for variable selection within the framework of the linear approximation model (Equation 18). Consider the following ℓ_0-estimator

$$\hat{\boldsymbol{\beta}}_M^{\lambda} = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}} \{\widehat{SSR}_{NT}(\boldsymbol{\beta}) + \lambda_M \|\boldsymbol{\beta}\|_{l0}\}, \tag{SM.4}$$

where $\widehat{SSR}_{NT}(\boldsymbol{\beta}) = \frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}(\ddot{y}_{it} - \boldsymbol{\beta}'\ddot{\boldsymbol{x}}_{it})^2$. λ_M is the tuning parameter that controls the sparsity level of the model. The Bayesian Information Criterion (BIC) can be used to select the tuning parameter λ_M:

$$\hat{\lambda}_M = \arg\min_{\lambda_M\in\Lambda} \left\{ \log \widehat{SSR}_{NT}(\boldsymbol{\beta}) + \lambda_{NT}\frac{\log(NT)}{NT}|S| \right\}, \tag{SM.5}$$

where $|S|$ presents the number of non-zero parameters, Λ is a grid of λ_M values, and λ_{NT} is a deterministic function of the sample size. Following Yang, Yao, and Xie (2024), We use $\lambda_{NT} = \log(\log(NT))$ in the algorithm. Once $\hat{\lambda}_M$ is determined, the corresponding best subset is identified.

The estimator (SM.5) can be computed using a cyclic coordinate descent (CCD) with local combinatorial search algorithm proposed by Hazimeh and Mazumder (2020). The idea of the algorithm is straightforward: it first employs CCD to quickly find a solution, and then uses a local combinatorial search formulated as an MIO problem to refine the solution obtained in the previous step. This algorithm can be implemented via the function 'L0Learn.fit' of R package 'L0Learn' (Hazimeh and Mazumder 2020).[1]

Although the Lasso-based variable selection method described in Section 3 performs well within the linear approximation framework, two potential limitations of this framework should be considered. First, the validity of the linear approximation for variable selection depends on the existence of the threshold effect and the restriction that $\boldsymbol{x}_{it} \neq \boldsymbol{s}_{it}$. Second, it remains unclear how the linear approximation model approximates the PSTCT model, since first-order asymptotic approximation may not be very accurate. To address these limitations, we adapt the block coordinate descent (BCD) algorithm proposed by Lee et al. (2021) as an alternative estimation strategy. [2] As noted by Lee et al. (2021), the Mixed-Integer Quadratic Programming (MIQP) step in BCD may run slowly when the dimension of $\boldsymbol{x}_{it}$ is large. To enhance computational efficiency, we modify the BCD algorithm by replacing the MIQP step with a grid search over threshold parameters and introducing a cut-off point to terminate iterations early. [3]

Next, we conduct Monte Carlo simulations to examine the finite-sample performance of the proposed algorithms. To ensure comparability, we retain the DGP, parameter settings and evaluation criteria from Section 5.3. In the BCD algorithm, the threshold parameter spaces are defined as $[q_{(\eta)}, q_{(1-\eta)}] \times \cdots \times [q_{(\eta)}, q_{(1-\eta)}]$, where $q_{(\eta)}$ denotes the η-th order statistic of q_{it} with $\eta = 0.15$. The parameter spaces for slope parameters are set as $[-2, 2] \times ... \times [-2, 2]$. We set the time limit Maxtime = 60s and maximum number of iterations $K^* = 2$, since K^* $= 2$ iterations would suffice in the algorithm. The number of replications is set to 1000, and the corresponding results are reported in Table SM.4.

Table SM.4 shows that the CCD algorithm achieves comparable variable selection accuracy to LASSO, with superior performance in smaller sample sizes (e.g. $N \times T < 250$). For the BCD algorithm, simulations under the PSTCT DGP with moderate smoothness ($k = 5$) demonstrate its ability to identify true covariates. In conclusion, our simulation results indicate that MIO-based algorithms perform well in variable selection, providing a viable tool for empirical robustness checks in empirical applications.

[1]For brevity, technical details of the implementation are omitted here but are available upon request.

[2]The original BCD algorithm was developed for factor-driven threshold models and may not perform optimally in our setting with small smoothness parameters. Future research could further explore the BCD algorithm in the context of our proposed model.

[3]For brevity, technical details of the implementation are omitted here but are available upon request.

Table SM.4: Variable selection performance of MIO-based algorithms.

T	N	PS	Accuracy	ERR	Precision	Recall	F-score
Panel A: Variable selection performance of the CCD algorithm							
5	50	0.9620	0.9810	0.0190	0.9661	0.9970	0.9813
	100	0.9910	0.9955	0.0045	0.9911	1.0000	0.9955
	200	0.9940	0.9970	0.0030	0.9940	1.0000	0.9970
10	50	0.9870	0.9935	0.0065	0.9872	1.0000	0.9995
	100	0.9990	0.9995	0.0005	0.9990	1.0000	0.9990
	200	1.0000	1.0000	0.0000	1.0000	1.0000	1.0000
20	50	0.9980	0.9990	0.0010	0.9980	1.0000	0.9990
	100	1.0000	1.0000	0.0000	1.0000	1.0000	1.0000
	200	1.0000	1.0000	0.0000	1.0000	1.0000	1.0000
Panel B: Variable selection performance of the BCD algorithm							
5	50	0.8060	0.9020	0.0980	0.8407	0.9920	0.9101
	100	0.8940	0.9470	0.0530	0.9042	1.0000	0.9497
	200	0.9480	0.9740	0.0260	0.9506	1.0000	0.9747
10	50	0.8830	0.9415	0.0585	0.8953	1.0000	0.9447
	100	0.9560	0.9780	0.0220	0.9579	1.0000	0.9785
	200	0.9680	0.9840	0.0160	0.9690	1.0000	0.9843
20	50	0.9410	0.9705	0.0295	0.9443	1.0000	0.9713
	100	0.9660	0.9830	0.0170	0.9671	1.0000	0.9833
	200	0.9710	0.9855	0.0145	0.9718	1.0000	0.9857

SM.E: Simulation Results for the Model with Two Covariate-Dependent Thresholds

In this section, we conduct Monte Carlo simulations to evaluate the finite-sample performance of the proposed panel smooth transition model with two covariate-dependent thresholds. To this end, we consider the following data-generating process (DGP):

$$y_{it} = \beta_1 x_{it} + \beta_2 x_{it} g_1(q_{it}; k_1, \gamma_{1,it}) + \beta_3 x_{it} g_2(q_{it}; k_2, \gamma_{2,it}) + \beta_4 z_{it} + \alpha_i + u_{it}, \tag{SM.6}$$

$$g_j(q_{it}; k_j, \gamma_{j,it}) = (1 + \exp(-k_j(q_{it} - \gamma_{j,it})))^{-1}, j = 1, 2, k_j > 0, \tag{SM.7}$$

$$\gamma_{j,it} = \gamma_{j0} + \gamma_{j1} s_{it}, j = 1, 2, \gamma_{1,it} < \gamma_{2,it} \tag{SM.8}$$

where $x_{it} = 5 + 0.25\alpha_i + u_{x,it}$, $z_{it} = 2 + 0.5\alpha_i + u_{z,it}$, $\alpha_i \overset{i.i.d}{\sim} N(0,1)$, $q_{it} \overset{i.i.d}{\sim} N(0.2, 1)$ and $s_{it} \overset{i.i.d}{\sim} N(0,1)$. The innovation processes $u_{x,it}$ and $u_{z,it}$ are independent of each other. $u_{it} \overset{i.i.d}{\sim} N(0, 0.5^2)$, $u_{x,it} \overset{i.i.d}{\sim} N(0,1)$ and $u_{z,it} \overset{i.i.d}{\sim} N(0,1)$. The number of replications is 1000.

To evaluate the performance of the proposed estimation procedure for multiple covariate-dependent thresholds, we set the true parameters as $(\beta_1, \beta_2, \beta_3, \beta_4) = (1, 2, 3, 1)$ and $(k_1, \gamma_{10}, \gamma_{11}, k_2, \gamma_{20}, \gamma_{21}) = (5, -0.6, 0.3, 5, 0.3, 0.5)$ and examine the empirical means and standard deviations of all parameters, which are reported in Table SM.5. The simulation results show that the mean of each parameter is close to its true value for all combinations of T and N, and the standard deviations decrease with the sample size. These results indicate that the estimation procedure works well in the case of multiple thresholds.

In the second experiment, we evaluate the finite-sample properties of the test statistic F_2 (defined in (28)), which tests one threshold against two thresholds. The size and power of F_2 are assessed under the null model (1)-(3) for one threshold and the alternative model (SM.6)-(SM.8) for two thresholds, respectively. The simulation results are reported in Table SM.6. These results indicate that the test statistic exhibits good size and power properties in finite samples.

Table SM.5: Estimates of the parameters obtained by using the estimator proposed in Section 4.

T	N	$\beta_1 = 1$		$\beta_2 = 2$		$\beta_3 = 3$		$\beta_4 = 1$		$k_1 = 5$		$\gamma_{10} = -0.6$		$\gamma_{11} = 0.3$		$k_2 = 5$		$\gamma_{20} = 0.3$		$\gamma_{21} = 0.5$	
		Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
5	50	0.999	0.044	2.009	0.093	2.992	0.087	1.001	0.036	5.004	0.352	-0.599	0.018	0.300	0.010	5.017	0.185	0.301	0.011	0.501	0.007
	100	1.000	0.031	2.003	0.060	2.998	0.056	1.000	0.025	5.006	0.234	-0.600	0.012	0.300	0.006	5.008	0.121	0.300	0.007	0.500	0.004
	200	1.000	0.021	2.002	0.041	2.998	0.039	1.000	0.018	5.004	0.167	-0.599	0.008	0.300	0.004	5.003	0.087	0.300	0.005	0.500	0.003
10	50	1.000	0.029	2.001	0.057	2.998	0.054	0.999	0.024	5.017	0.226	-0.600	0.011	0.300	0.006	5.009	0.119	0.300	0.007	0.500	0.004
	100	0.999	0.020	2.002	0.039	2.999	0.037	1.000	0.017	5.000	0.155	-0.600	0.008	0.300	0.004	5.002	0.085	0.300	0.005	0.500	0.003
	200	1.000	0.015	2.002	0.029	2.999	0.027	1.000	0.011	4.998	0.113	-0.600	0.006	0.300	0.003	5.001	0.058	0.300	0.004	0.500	0.002
20	50	1.000	0.020	1.998	0.039	3.001	0.036	1.000	0.016	5.009	0.152	-0.600	0.008	0.300	0.004	5.002	0.083	0.300	0.005	0.500	0.003
	100	1.000	0.014	2.000	0.027	3.000	0.026	1.000	0.012	5.003	0.109	-0.600	0.005	0.300	0.003	5.001	0.059	0.300	0.003	0.500	0.002
	200	1.000	0.010	1.999	0.020	3.000	0.018	1.000	0.008	5.005	0.075	-0.600	0.004	0.300	0.002	5.001	0.041	0.300	0.002	0.500	0.001

Table SM.6: Finite-sample size and power of the F_2 test statistic.

T	N	Size	Power
5	50	0.043	1.000
	100	0.047	1.000
	200	0.055	1.000
10	50	0.035	1.000
	100	0.045	1.000
	200	0.039	1.000
20	50	0.066	1.000
	100	0.049	1.000
	200	0.056	1.000

SM.F: Simulation Experiments for the Parameter Regions Where the F-Type Tests Perform Poorly

In this section, more detailed Monte Carlo simulations are conducted to examine the parameter regions where the powers of the F-tests are weak. The DGP is the same as that in Section 5.1. We set $(\gamma_0, \gamma_1) = (0.2, 0.5)$, $(\beta_1, \beta_3) = (1, 1)$ and vary (β_2, k) to investigate the parameter regions where the F_1 test has low power. Meanwhile, we set $(k, \gamma_0) = (5, 0.2)$, $(\beta_1, \beta_3) = (1, 1)$ and vary (β_2, γ_1) to investigate the parameter regions where the F_C test performs poorly. The number of replications is 100 to reduce computational cost.

The first experiment presents a detailed examination of the powers of the F-tests for a range of parameter combinations for sample sizes such as $(T, N) = (5, 100)$.[4] The magnitude of the threshold effect depends on both β_2 and k, and thus affects the power of the test F_1. Therefore, we investigate the power of the test F_1 by varying the combinations of β_2 and k. Threshold constancy is examined only when the threshold effect exists, and γ_1 determines the magnitude of the threshold constancy. Thus, we examine the power of the F_C test by varying β_2 and γ_1, which determine the magnitude of the threshold effect and threshold constancy, respectively. From these simulations, we can observe how the powers of the F-type tests vary with the parameters. The simulation results are reported in Table SM.7. To present the simulation results more intuitively, the parameter regions where the F-type tests perform poorly (below 0.9) are highlighted in gray.

As shown in Table SM.7, small values of β_2 and k reduce the power of F_1. For example, when $\beta_2 = 0.01$, the power of F_1 falls below 0.2 across all values of k. For $k \leq 0.05$, the power of F_1 remains below 0.7 across all tested β_2 values. In such cases, the test F_1 fails to differentiate between the linear model and the proposed model effectively, due to the relatively small threshold effect (e.g. $\beta_2 \leq 0.01$, $k \leq 0.05$). Next, we focus on the power of the test F_C. For $\beta_2 \leq 0.05$ (insufficient threshold effect), the power of F_C remains below 0.7

[4]We also conduct simulations for sample sizes $(T, N) = (5, 50)$ and $(T, N) = (10, 50)$. To save space, we only outline the results for sample sizes $(T, N) = (5, 100)$ in this online Appendix.

Table SM.7: Powers of the F-type test statistics under different parameter combinations.

Test for threshold effect F_1												
$k\backslash\beta_2$	0.01	0.05	0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90	1.00
0.01	0.050	0.030	0.030	0.040	0.040	0.040	0.050	0.050	0.070	0.110	0.110	0.100
0.05	0.050	0.050	0.050	0.100	0.130	0.150	0.200	0.300	0.340	0.400	0.530	0.610
0.10	0.070	0.050	0.080	0.140	0.330	0.420	0.590	0.800	0.900	0.940	0.990	1.000
0.50	0.060	0.180	0.530	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.00	0.060	0.440	0.970	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.50	0.090	0.750	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
2.00	0.110	0.840	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
2.50	0.110	0.890	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
3.00	0.100	0.900	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
3.50	0.080	0.940	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
4.00	0.110	0.920	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
4.50	0.100	0.960	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
5.00	0.100	0.950	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Test for threshold constancy F_C												
$\gamma_1\backslash\beta_2$	0.01	0.05	0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90	1.00
0.01	0.050	0.050	0.050	0.030	0.060	0.080	0.090	0.090	0.110	0.140	0.180	0.200
0.05	0.020	0.030	0.070	0.200	0.330	0.550	0.740	0.890	0.980	0.980	1.000	1.000
0.10	0.070	0.090	0.230	0.580	0.890	0.980	1.000	1.000	1.000	1.000	1.000	1.000
0.15	0.040	0.130	0.410	0.890	0.980	1.000	1.000	1.000	1.000	1.000	1.000	1.000
0.20	0.040	0.180	0.450	0.990	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
0.25	0.020	0.200	0.650	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
0.30	0.080	0.370	0.820	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
0.35	0.020	0.380	0.890	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
0.40	0.040	0.360	0.950	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
0.45	0.050	0.490	0.980	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
0.50	0.050	0.610	0.970	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000

Table SM.8: Powers of the F-type test statistics under specific parameter settings.

T	N	Test for threshold effect F_1		Test for threshold constancy F_C	
		$(\beta_2 = 0.05, k = 5)$	$(\beta_2 = 1, k = 0.05)$	$(\beta_2 = 0.1, \gamma_1 = 0.3)$	$(\beta_2 = 1, \gamma_1 = 0.03)$
5	50	0.630	0.410	0.410	0.620
	100	0.890	0.590	0.860	0.930
	200	1.000	0.930	0.960	1.000
10	50	0.990	0.630	0.870	0.920
	100	1.000	0.920	0.990	1.000
	200	1.000	1.000	1.000	1.000
20	50	1.000	0.970	0.990	1.000
	100	1.000	1.000	1.000	1.000
	200	1.000	1.000	1.000	1.000

regardless of γ_1 values. When $\gamma_1 = 0.01$ (indicating nearly constant threshold), the power of F_C remains below 0.2 even with $\beta_2 = 1.00$ (a strong threshold effect). It is possible to enhance the powers of the F-type tests by increasing the parameter values and/or sample size. For instance, the power of the test F_1 reaches a high level when $\beta_2 = 0.10$ and $k = 1.00$. The power of the test F_C reaches 0.990 when β_2 is 0.20 and γ_1 is 0.20.

The second experiment examines the specific parameter settings where the power of the F-type tests is weak across all $T = 5, 10, 20$ and $N = 50, 100, 200$. For the F_1 test, we simulate small threshold effects by setting (β_2, k) to (0.05,5) and (1,0.05). For the F_C test, we consider two cases: (i) a small threshold effect with large threshold constancy $(\beta_2, \gamma_1) = (0.1, 0.3)$, and (ii) a large threshold effect with small threshold constancy $(\beta_2, \gamma_1) = (1, 0.03)$. Table SM.8 shows that F_1 has low power when either β_2 or k is small, and F_C performs poorly if either β_2 or γ_1 is small.

In conclusion, the F_1 test performs poorly in the parameter regions with small k (e.g. 0.05) or β_2 (e.g. 0.05), and the F_C test performs poorly when either γ_1 (e.g. 0.01) or β_2 (e.g. 0.05) is small. These simulation results demonstrate the parameter regions where the F-type tests perform poorly, which also indicate the robustness of our proposed test statistics.

SM.G: Simulation Results of the Performance of Variable Selection When the Time Trend Is Included as a Covariate

In this section, we conduct a Monte Carlo simulation to examine the performance of the variable selection procedure when the time trend is included as a covariate. To this end, we set $s_{1,it} = t/T$; other aspects of the DGP are identical to those in Section 5.3. The number of replications is set to 1000, and the simulation results are reported in Table SM.9. The results show that PS, $Accuracy$, $Precision$, $Recall$ and $F\text{–}score$ increase as T or N increases, while ERR decreases. Perfect variable selection is achieved when $N \times T > 1000$. These results indicate that the variable selection procedure remains robust when a time trend is included as a covariate.

Table SM.9: Performance of variable selection when time trend as a covariate.

T	N	PS	Accuracy	ERR	Precision	Recall	F-score
5	50	0.9030	0.9515	0.0485	0.9989	0.9040	0.9491
	100	0.9830	0.9915	0.0085	1.0000	0.9830	0.9914
	200	1.0000	1.0000	0.0000	1.0000	1.0000	1.0000
10	50	0.9850	0.9925	0.0075	1.0000	0.9850	0.9924
	100	0.9990	0.9995	0.0005	1.0000	0.9990	0.9995
	200	1.0000	1.0000	0.0000	1.0000	1.0000	1.0000
20	50	0.9990	0.9995	0.0005	1.0000	0.9990	0.9995
	100	1.0000	1.0000	0.0000	1.0000	1.0000	1.0000
	200	1.0000	1.0000	0.0000	1.0000	1.0000	1.0000

References

Bertsimas, D., A. King, and R. Mazumder. 2016. "Best Subset Selection via a Modern Optimization lens." *Annals of Statistics* 44(2): 813-852.

Dueker, M. J., Z. Psaradakis, M. Sola, and F. Spagnolo. 2013. "State-Dependent Threshold Smooth Transition Autoregressive Models." *Oxford Bulletin of Economics and Statistics* 75(6): 835-854.

El-Shagi, M. 2011. "An Evolutionary Algorithm for the Estimation of Threshold Vector Error Correction Models." *International Economics and Economic Policy* 8: 341-362.

Hansen, B. E. 1999. "Threshold Effects in Non-Dynamic Panels: Estimation, Testing, and Inference." *Journal of Econometrics* 93(2): 345-368.

Hansen, B. E. 2017. "Regression Kink with an Unknown Threshold." *Journal of Business & Economic Statistics* 35(2): 228-240.

Hazimeh, H., and R. Mazumder. 2020. "Fast Best Subset Selection: Coordinate Descent and Local Combinatorial Optimization Algorithms." *Operations Research* 68(5): 1517-1537.

Hegerty, B., C. C. Hung, and K. Kasprak. 2009. "A Comparative Study on Differential Evolution and Genetic Algorithms for Some Combinatorial Problems." *In: Proceedings of 8th Mexican International Conference on Artificial Intelligence* (pp. 9-13).

Lee, S., Y. Liao, M. H. Seo, and Y. Shin. 2021. "Factor-Driven Two-Regime Regression." *Annals of Statistics* 49(3): 1656-1678.

Nevitt, J., and G. R. Hancock. 2001. "Performance of Bootstrapping Approaches to Model Test Statistics and Parameter Standard Error Estimation in Structural Equation Modeling." *Structural Equation Modeling* 8(3): 353-377.

Omay, T., and F. Emirmahmutoğlu. 2017. "The Comparison of Power and Optimization Algorithms on Unit Root Testing with Smooth Transition." *Computational Economics* 49: 623-651.

Osborne, M. R. 2007. "Separable Least Squares, Variable Projection, and the Gauss-Newton Algorithm." *Electronic Transactions on Numerical Analysis* 28(2): 1-15.

Ramírez-Rondán, N. R. 2020. "Maximum Likelihood Estimation of Dynamic Panel Threshold Models." *Econometric Reviews* 39(3): 260-276.

Tušar, T., and B. Filipič. 2007. "Differential Evolution Versus Genetic Algorithms in Multi-objective Optimization." *In: Proceedings of International Conference on Evolutionary Multi-Criterion Optimization* (pp. 257-271).

Yang, L. 2024. "Panel Threshold Model with Covariate-Dependent Thresholds and its Application to the Cash Flow/Investment Relationship." *Studies in Nonlinear Dynamics & Econometrics* 28(4): 645-659.

Yang, L., I. P. Chen, C. Lee, and M. Ren. 2024. "Panel Threshold Model with Covariate-dependent Thresholds and Unobserved Individual-Specific Threshold Effects." *Econometric Reviews* 43(7): 452–489.

Yang, L., L. Yao, and Y. Xie. 2024. "Panel Kink Threshold Model with Multiple Covariate-Dependent Thresholds." *Applied Economics Letters* 1-4.

Yu, P., and X. Fan. 2021. "Threshold Regression with a Threshold Boundary." *Journal of Business & Economic Statistics* 39(4): 953-971.