SCBench: A Testbed for Causal Inference with Time Series Panel Data

Anonymous Authors[1]

Abstract

Synthetic control (SC) is a widely used method for estimating causal effects from observational panel data, with classical approaches expressing a target trajectory as a linear combination of donor trajectories. Recent advances in foundation models pretrained on synthetic data have shown strong performance in structured, sample-limited regimes such as tabular and time-series prediction, raising the question of whether such models are also effective for synthetic control. We conduct a large-scale empirical study of traditional and foundation model approaches on over 300K simulated panels, spanning fully linear to fully nonlinear state-space dynamics, varying noise regimes, and a range of panel sizes and ranks. Comparing three foundation models (TabPFN, TabPFN-TS, and Chronos) against standard SC baselines (Robust SC, Lasso, and Simplex), we identify regimes where foundation models outperform classical baselines, including when latent dynamics are nonlinear and panels are high rank. Linear methods remain competitive or superior in low-rank and near-linear settings, with Simplex providing a reliable baseline across our testbed. These results suggest that foundation models pretrained on synthetic data are a promising direction for synthetic control in challenging regimes, and we release our benchmarks and analysis to support future work.

1. Introduction

Synthetic control (SC) is a widely used method for estimating causal effects from observational data, developed in 2002 to measure the impact of political conflict on economic growth in Basque Country. Estimating this effect is nontrivial because of the absence of an appropriate control, i.e., a region very similar to Basque Country without conflict. This challenge arises broadly in evaluating both natural interventions, such as extreme weather events, and applied interventions, such as new tax policy.

Synthetic control addresses this by constructing a control from a set of candidate units. The intuition is that while no individual region may serve as a faithful control, a combination of regions may. In the Basque example, SC first identifies a linear combination of donor regions whose pre-intervention GDPs closely track Basque Country's. Assuming this relationship would have persisted absent the intervention, the same combination is used to predict the counterfactual post-intervention GDP. The causal effect is then estimated as the gap between this counterfactual and the observed trajectory.

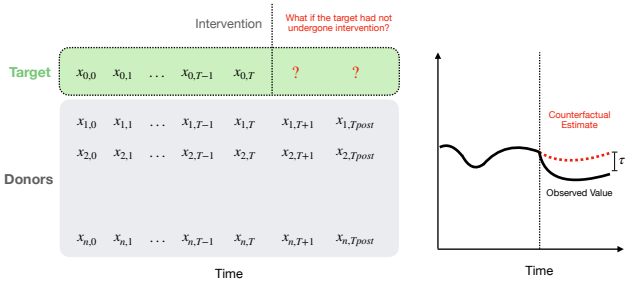


Figure 1. The synthetic control task is to predict the future time steps of a target using trajectories of a set of donors (left). The treatment effect is then estimated by comparing the predictions to the observed values under intervention (right).

The problem of synthetic control is uniquely structured. Each problem instance is a panel of time-series trajectories (Figure 1). Rather than extrapolating a single series, the goal is to predict the future time-steps of one trajectory (the target) from the trajectories of others (the donors). Therefore, the core inductive task is comparison across trajectories within a set. The predominant methods for synthetic control are linear, expressing the target as a weighted sum of donors. At the same time, recent work has shown that large models trained on synthetic data can be highly effective in structured, sample-limited regimes such as tabular data and time-series forecasting, suggesting potential opportunities for use in synthetic control.

In this work, we conduct a large-scale empirical study of traditional and foundation model approaches on over 300K simulated panels, spanning fully linear to fully nonlinear state-space dynamics, varying input noise regimes, and a range of panel sizes and ranks. We compare three representative foundation models (TabPFN, TabPFN-TS, and Chronos) against standard SC baselines (Robust SC, Lasso, and Sim-

plex), and identify areas of substantial promise for foundation models on this task. In aggregate, TabPFN achieves the lowest mean and median error across our testbed, and foundation models more broadly excel in challenging regimes: they outperform traditional methods when latent dynamics are nonlinear and when low-rank assumptions do not hold, while exhibiting similar noise robustness to SC baselines. At the same time, linear methods remain competitive or superior in low-rank and near-linear settings, and provide a reliable baseline across the full range of conditions we study. Together, these results suggest that foundation models pretrained on synthetic data are a promising direction for synthetic control in challenging regimes. We release our benchmarks and analysis to support future work[1], and provide a detailed discussion of opportunities for further study of these methods.

2. Related Work

Synthetic control. The synthetic control (SC) framework of (Abadie et al., 2010) estimates a counterfactual for a treated unit by reweighting donor units to match the pre-intervention outcome trajectory. SC has since become a standard tool in applied econometrics and social sciences (Abadie, 2021), with wide-ranging applications including in evaluating disaster response (Heersink et al., 2017), public health (Pieters et al., 2016), immigration enforcement (Bohn et al., 2014), and the efficacy of nonprofit programs (Abadie, 2021). Method development in SC has traditionally focused on improving the robustness of linear methods, including pre-processing steps such as RSC and weight regularizations such as Lasso and Simplex. More recent work has working to relax the assumption that donor-target relationships are constant over time (Rho et al., 2026), which exhibits better performance than linear methods when strong temporal trends exist.

Foundation models for structured data. Recent foundation models for time-series forecasting and tabular data have been developed by training large transformers on broad corpora of real and synthetic series. TabPFN (Hollmann et al., 2023) pretrains a transformer on millions of synthetic tabular datasets sampled from a structural prior, and at inference time produces predictions for a new dataset in a single forward pass. TabPFN-TS (Hoo et al., 2025) leverages TabPFN-v2 for forecasting problems by combining it with lightweight time-features, achieving strong performance on forecasting tasks without any additional training. (Das et al., 2024) introduce a decoder-only patched-attention model whose zero-shot accuracy approaches that of supervised baselines across a range of public datasets, and Chronos (Ansari et al., 2025) extend this paradigm to multivariate and covariate-informed forecasting via a group attention mechanism trained on synthetically structured multivariate data. A concurrent line of work (Illick et al., 2026) explores appraoches for adapting time-series foundation models for SC in a limited set of synthetic and real-world regimes, and proposes input representations tailored to the SC setting. We provide a complementary view at substantially larger scale, systematically varying linearity, noise structure, and panel rank across over 300K simulated panels.

[1] anonymized

Parameter	Values
$\alpha = \beta$ (linearity)	$\{0,\ 0.25,\ 0.5,\ 0.75,\ 1\}$
Transition matrix	ortho (ρ=1); rescaled ($\rho \in \{0.8, 1\}$)
Loadings (H, C)	Dir_5, Gauss
Initial state (x_0)	Dir_5, Gauss
Noise covariance	diag; Wishart-$(d{+}1)$; Wishart-$10d$
Noise level	low, high
Latent dim. d	$\{2,\ 5,\ 10,\ 20\}$
Donors n	$\{5,\ 10,\ 20,\ 50\}$
Pre-period T_{pre}	$\{5,\ 10,\ 20,\ 50\}$
Fixed: $T_{\text{post}}{=}T_{\text{pre}}$, burn-in=5, no normalization.	

Table 1. Data generation parameter sweep. Each of the 16,200 parameter combinations is sampled 20 times, yielding 324,000 panels.

2.1. Data Generation Process

We sample tasks from a nonlinear state-space model with d-dimensional latent state $x_t \in \mathbb{R}^d$ and $(n+1)$-dimensional observation $y_t \in \mathbb{R}^{n+1}$, where index 0 denotes the target unit and indices $1, \ldots, n$ denote donors. Each task draws fresh parameters (A, B, H, C, Q, R, x_0) from the priors below:

$$x_t = (1-\alpha)\, A\, x_{t-1} + \alpha\, \tanh\big(B\, x_{t-1}\big) + q_{t-1}, \quad (1)$$

$$y_t = (1-\beta)\, H\, x_t + \beta\, \tanh\big(C\, x_t\big) + r_t, \quad (2)$$

where $q_{t-1} \sim \mathcal{N}(\mathbf{0},\ Q)$ and $r_t \sim \mathcal{N}(\mathbf{0},\ R)$.

We generate a large sweep of hyperparameters, summarized in Table 1 (see Appendix A.1 for full details). In total, we consider over 16K parameter combinations, with 20 independent samples for each, yielding 324K total panels of synthetic data.

3. Methods

3.1. Predictors

We compare traditional synthetic control methods, which express the target as a linear combination of donors, with foundation models pretrained on large corpora of synthetic or real data.

Traditional baselines. Simplex (Abadie et al., 2010) restricts donor weights to the unit simplex, requiring weights

Table 2. We find that foundation models excel in nonlinear and high-rank data regimes, while linear methods outperform them in linear and low-rank data regimes. Shown is the median MSE compared to clean target values $h \leq 5$. The best baseline per row in **bold**; 95% CIs in grey.

	RSC	Lasso	Simplex	TabPFN	TabPFN-TS	Chronos
Linearity coefficient ($\alpha = \beta$)						
0.00	1.53 (1.50, 1.56)	1.59 (1.55, 1.63)	**1.08** (1.06, 1.10)	1.18 (1.16, 1.21)	1.30 (1.27, 1.32)	1.43 (1.41, 1.46)
0.25	1.04 (1.02, 1.05)	1.02 (0.99, 1.04)	**0.60** (0.59, 0.61)	0.60 (0.59, 0.62)	0.68 (0.67, 0.69)	0.67 (0.66, 0.69)
0.50	0.90 (0.89, 0.92)	0.82 (0.79, 0.84)	0.48 (0.47, 0.49)	**0.43** (0.42, 0.44)	0.49 (0.48, 0.50)	0.47 (0.46, 0.48)
0.75	0.84 (0.83, 0.86)	0.73 (0.71, 0.75)	0.43 (0.43, 0.44)	**0.36** (0.36, 0.37)	0.42 (0.41, 0.42)	0.38 (0.38, 0.39)
1.00	0.87 (0.85, 0.88)	0.78 (0.76, 0.80)	0.48 (0.47, 0.49)	**0.41** (0.40, 0.41)	0.46 (0.46, 0.47)	0.43 (0.42, 0.44)
Relative rank $d/\min(n, T)$						
0.04	**0.26** (0.24, 0.28)	0.63 (0.53, 0.77)	0.27 (0.25, 0.29)	0.34 (0.33, 0.36)	0.35 (0.33, 0.36)	0.33 (0.31, 0.34)
0.1	0.43 (0.42, 0.44)	0.69 (0.65, 0.74)	0.37 (0.36, 0.38)	**0.35** (0.34, 0.36)	0.37 (0.36, 0.38)	0.41 (0.39, 0.41)
0.2	0.89 (0.87, 0.91)	0.83 (0.80, 0.86)	0.49 (0.48, 0.50)	**0.48** (0.47, 0.49)	0.55 (0.54, 0.56)	0.55 (0.54, 0.56)
0.25	0.56 (0.55, 0.58)	0.77 (0.72, 0.81)	0.43 (0.42, 0.45)	**0.38** (0.37, 0.39)	0.43 (0.42, 0.45)	0.42 (0.41, 0.44)
0.4	1.48 (1.45, 1.51)	1.00 (0.98, 1.03)	**0.64** (0.62, 0.65)	0.64 (0.62, 0.65)	0.75 (0.74, 0.77)	0.73 (0.72, 0.75)
0.5	0.96 (0.94, 0.98)	0.92 (0.90, 0.95)	0.56 (0.55, 0.57)	**0.49** (0.48, 0.50)	0.54 (0.54, 0.55)	0.52 (0.51, 0.53)
1	1.39 (1.36, 1.40)	1.08 (1.06, 1.10)	0.68 (0.67, 0.69)	**0.57** (0.56, 0.58)	0.66 (0.65, 0.67)	0.60 (0.60, 0.61)

to be non-negative and sum to one. Lasso fits unconstrained weights with an ℓ_1 penalty, inducing sparsity over donors. Robust SC (RSC) (Amjad et al., 2018) first denoises the donor matrix via singular-value thresholding, then fits unconstrained weights on the resulting low-rank approximation.

Foundation models. TabPFN (Hollmann et al., 2023) is a transformer pretrained on synthetic tabular regression tasks that performs in-context prediction without gradient updates. We apply it to SC by treating each donor unit as a feature and each time step as a row, with the target trajectory as the regression target. TabPFN-TS (Hoo et al., 2025) adapts TabPFN to time-series forecasting by augmenting each row with temporal features (e.g., position, seasonality encodings). Chronos (Ansari et al., 2025) is a transformer pretrained on a large corpus of real and synthetic time series; we apply it in the panel setting by passing donor trajectories as known future covariates alongside the target's pre-intervention history.

3.2. Evaluation

All methods are fit on pre-intervention time points (T_{pre}), and applied on the next T_{post} steps for the target unit. For each task, we generate T pre-intervention time steps (used as context) and an additional T_{post} post-intervention time steps (used for evaluation), where $T_{\text{post}} \leq T$. We evaluate predictions at each post-intervention horizon $h = 1, \ldots, T_{\text{post}}$, allowing us to measure how prediction quality degrades as the forecast horizon extends further past the intervention. We compare predictions against two ground-truth targets: Noisy observation ($y_t = (1-\beta)Hx_t + \beta \tanh(Cx_t) + r_t$), and the Noiseless signal ($y_t^* = (1-\beta)Hx_t + \beta \tanh(Cx_t)$). We evaluate predictions at $T_{post} = [1..5]$.

4. Results

We compare foundation models and traditional methods on our 300K synthetic panel testbed and find both areas of substantial promise for foundation models and regimes where linear methods remain the stronger choice. Table 2 summarizes our results, broken down by linearity coefficient and relative panel rank. Across all settings, TabPFN achieves the strongest mean and median performance overall, while Simplex remains the most reliable linear baseline. In the following sections, we characterize this tradeoff in detail. We first study how each method behaves as latent dynamics shift from linear to nonlinear (Section 4.1), then examine robustness under varying noise magnitudes and correlation structures (Section 4.2), and finally evaluate scaling with respect to panel rank (Section 4.3). Appendix B and B.6 contain full ablation results, plots broken down by additional axes, and all results replicated against noisy targets.

4.1. Performance Across Linearity Regimes

Figure 2 shows results as a function of the linearity coefficient, where $\alpha = \beta = 0$ reproduces a fully linear state-space model and $\alpha = \beta = 1$ produces the fully nonlinear model in Equation 2. Fully linear settings yield larger and more skewed errors across all methods, and the $\tanh$ nonlinearity appears to provide implicit regularization at higher α, β. TabPFN achieves the lowest median MSE at all configurations with $\alpha = \beta \geq 0.50$, while Simplex matches or outperforms TabPFN in fully linear and near-linear settings.

4.2. Robustness to Noise

We evaluate each method under two noise magnitudes (low and high) and two correlation structures, one fully ran-

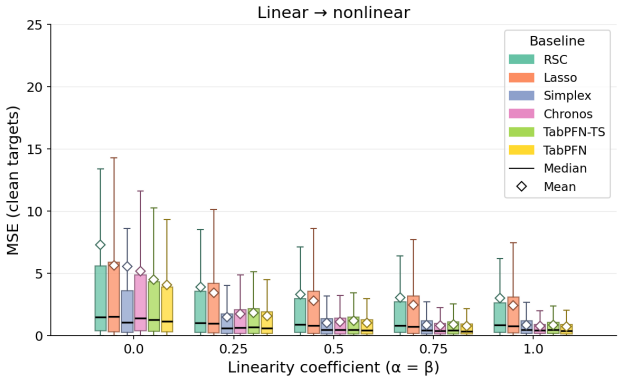


Figure 2. Testbed results according to linearity coefficients. $\alpha = \beta = 0$ reproduces a fully linear state-space model, and $\alpha = \beta = 1$ produces the fully nonlinear state-space model described in Eq. 2.

Table 3. Noise summary (median MSE, clean targets, $h \leq 5$). Best baseline per row in **bold**. 95% CIs are reported in Tables 11 and 12.

Noise	RSC	Lasso	Simplex	TabPFN	TabPFN-TS	Chronos
Low / Random	0.29	0.21	0.18	**0.17**	0.20	0.20
Low / Correlated	0.30	0.21	**0.18**	0.18	0.20	0.20
High / Random	2.74	4.16	1.46	**1.23**	1.49	1.37
High / Correlated	3.04	3.79	1.49	**1.44**	1.64	1.51

dom and one correlated with the latent state. Both correlation structures have the same expected magnitude (see Appendix 1). Table 3 reports median MSE across these four settings. Noise magnitude scales the absolute error of all methods, while correlation structure has comparatively little effect. TabPFN achieves the lowest or near-lowest median MSE in all four settings, with Simplex closely matching its performance. Foundation models thus exhibit noise robustness similar to the strongest traditional baselines.

4.3. Scaling with Panel Rank

Synthetic control typically assumes approximately low-rank structure in the panel matrix. Figure 3 shows results as a function of relative rank, defined as the panel's rank divided by the minimum of the sample and time dimensions. Robust SC and Simplex are most competitive at the lowest relative ranks (0.04 and 0.4), while TabPFN outperforms traditional methods at relative ranks of 0.1, 0.2, 0.25, 0.5, and 1.0, with the gap widening as rank increases. Foundation models thus offer substantive gains in high-rank regimes where classical low-rank assumptions are violated.

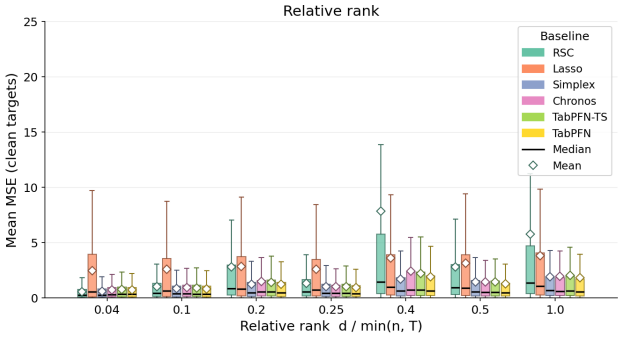


Figure 3. Testbed results as a function of relative rank, defined as the panel's rank divided by the minimum of the sample and time dimensions.

5. Discussion

5.1. Limitations

Our evaluation is conducted entirely on synthetic panels generated from state-space models. While this design lets us systematically vary structural assumptions such as linearity, noise, and rank, real-world panels may follow structures not well represented by our simulated data. We highlight opportunities to extend our evaluation below.

5.2. Future Work

Adapting foundation models to synthetic control. The foundation models we evaluate were pretrained for generic tabular or time-series prediction, not for the specific structure of synthetic control. The strong performance we observe despite this mismatch suggests both the promise of large-scale synthetic pretraining and opportunities to adapt or design models tailored to the synthetic control problem.

Benchmarking improvements and practitioner guidance. Synthetic control evaluations have historically relied on a small number of canonical case studies, such as Proposition 99 and German reunification, where the counterfactual is unobservable and rigorous comparison across methods is difficult. To support more systematic comparison, we release our testbed of over 300K simulated panels, which to our knowledge is the largest evaluation set available for SC. We hope this resource lowers the barrier to broad-coverage method comparison and encourages further extensions of our work, particularly toward semi-synthetic benchmarks built from real panels. Our results enable richer comparison of the data regimes in which different predictors are effective, and where each breaks down. Building tooling and guidance for practitioners to study when each regime applies in practice remains an important direction for future work.

References

Abadie, A. Using Synthetic Controls: Feasibility, Data Requirements, and Methodological Aspects. *Journal of Economic Literature*, 59(2):391–425, June 2021. ISSN 0022-0515. doi: 10.1257/jel.20191450. URL https://pubs.aeaweb.org/doi/10.1257/jel.20191450.

Abadie, A., Diamond, A., and Hainmueller, J. Synthetic Control Methods for Comparative Case Studies: Estimating the Effect of California's Tobacco Control Program. *Journal of the American Statistical Association*, 105(490):493–505, June 2010. ISSN 0162-1459, 1537-274X. doi: 10.1198/jasa.2009.ap08746. URL http://www.tandfonline.com/doi/abs/10.1198/jasa.2009.ap08746.

Amjad, M., Shah, D., and Shen, D. Robust Synthetic Control. *Journal of Machine Learning Research*, 19(22):1–51, 2018. ISSN 1533-7928. URL http://jmlr.org/papers/v19/17-777.html.

Ansari, A. F., Shchur, O., Küken, J., Auer, A., Han, B., Mercado, P., Rangapuram, S. S., Shen, H., Stella, L., Zhang, X., Goswami, M., Kapoor, S., Maddix, D. C., Guerron, P., Hu, T., Yin, J., Erickson, N., Desai, P. M., Wang, H., Rangwala, H., Karypis, G., Wang, Y., and Bohlke-Schneider, M. Chronos-2: From Univariate to Universal Forecasting, October 2025. URL http://arxiv.org/abs/2510.15821. arXiv:2510.15821 [cs].

Bohn, S., Lofstrom, M., and Raphael, S. Did the 2007 Legal Arizona Workers Act Reduce the State's Unauthorized Immigrant Population? *The Review of Economics and Statistics*, 96(2):258–269, May 2014. ISSN 0034-6535. doi: 10.1162/REST_a_00429. URL https://doi.org/10.1162/REST_a_00429.

Das, A., Kong, W., Sen, R., and Zhou, Y. A decoder-only foundation model for time-series forecasting, April 2024. URL http://arxiv.org/abs/2310.10688. arXiv:2310.10688 [cs].

Heersink, B., Peterson, B. D., and Jenkins, J. A. Disasters and Elections: Estimating the Net Effect of Damage and Relief in Historical Perspective. *Political Analysis*, 25(2):260–268, April 2017. ISSN 1047-1987, 1476-4989. doi: 10.1017/pan.2017.7. URL https://www.cambridge.org/core/journals/political-analysis/article/disasters-and-elections-estimating-the-net-effect-of-damage-and-relief-in-historical-perspe1E944D5BAC20A1D2F86E8419BF265C4F.

Hollmann, N., Müller, S., Eggensperger, K., and Hutter, F. TabPFN: A Transformer That Solves Small Tabular Classification Problems in a Second, September 2023. URL http://arxiv.org/abs/2207.01848. arXiv:2207.01848 [cs].

Hoo, S. B., Müller, S., Salinas, D., and Hutter, F. From Tables to Time: How TabPFN-v2 Outperforms Specialized Time Series Forecasting Models, May 2025. URL http://arxiv.org/abs/2501.02945. arXiv:2501.02945 [cs] version: 3.

Illick, C., Rho, S., and Misra, V. Causal Inference with Time Series Foundation Models. April 2026. URL https://openreview.net/forum?id=pqWJ6BXKS0.

Pieters, H., Curzi, D., Olper, A., and Swinnen, J. Effect of democratic reforms on child mortality: a synthetic control analysis. *The Lancet Global Health*, 4(9):e627–e632, September 2016. ISSN 2214-109X. doi: 10.1016/S2214-109X(16)30104-8. URL https://www.thelancet.com/journals/langlo/article/PIIS2214-109X(16)30104-8/fulltext.

Rho, S., Illick, C., Narasipura, S., Abadie, A., Hsu, D., and Misra, V. Time-Aware Synthetic Control, January 2026. URL http://arxiv.org/abs/2601.03099. arXiv:2601.03099 [cs] version: 1.

Table 4. Data generation parameter sweep: swept dimensions and fixed parameters. A total of 16,200 parameter combinations, each with 20 independent samples generated. We perform a complete grid except for under-specified samples, where rank $d > min(n, T)$, resulting in a total of 324,000 SC panels evaluated.

Parameter	Values / range	Description
Swept (23,040 parameter combinations)		
$\alpha = \beta$	$\{0,\ 0.25,\ 0.5,\ 0.75,\ 1\}$	Linear/nonlinear mixing coefficient.
Transition Matrix	$\{(\text{ortho}, \rho{=}1),\ (\text{rescaled}, \rho{=}1),\ (\text{rescaled}, \rho{=}0.8)\}$	Transition matrix construction.
Loading Distribution	$\{\text{Dir}_5, \text{Gauss}\}$	Observation matrix rows.
Noise Variant	$\{\text{diag}, \text{wishart-}\nu{=}d{+}1, \text{wishart-}\nu{=}10d\}$	Noise covariance structure.
Noise Level	{low, high}	Noise variance band.
x_0 Initial State Distribution	$\{\text{Dir}_5, \text{Gauss}\}$	Initial latent state distribution.
d (latent dim.)	$\{2,\ 5,\ 10,\ 20\}$	Latent state dimension.
n (donors)	$\{5,\ 10,\ 20,\ 50\}$	Number of donor units.
T_{pre}	$\{5,\ 10,\ 20,\ 50\}$	Pre-intervention period length.
Fixed across all parameter combinations		
T_{post}	T_{pre}	Post-intervention horizon.
burn-in	5 latent steps	Burn-in length.
normalization	none	Donor-matrix normalization.
samples per parameter combination	20	Panels generated for each parameter combination.

A. Appendix

A.1. Parameter Sweep for Data Generation

We provide a more comprehensive description of our data generation process in Table 4. For the transition matrix, *orthogonal* refers to drawing a random Gaussian matrix $G \sim \mathcal{N}(0,\ I)^{d\times d}$ and taking the Q factor of its QR decomposition; this places every eigenvalue exactly on the unit circle ($\rho = 1$), giving volume-preserving rotational dynamics. *Rescaled* instead draws G and divides by its spectral radius before scaling by the target $\rho \in \{1.0,\ 0.8\}$; eigenvalues are not constrained to the unit circle, so the same ρ can produce a mixture of contracting, expanding, and oscillating modes. For the loading distribution, Dir_5 draws each row of H and C from a symmetric Dirichlet with concentration $\kappa = 5$ (rows lie on the simplex with moderate concentration), while Gauss draws each entry iid $\mathcal{N}(0,\ 1)$ (signed and unconstrained). The noise variants control the structure of Q and R: *diag* draws independent per-feature variances; *wishart-*$\nu{=}d{+}1$ and *wishart-*$\nu{=}10d$ draw correlated covariances around the same diagonal mean, with off-diagonal correlation scaling as $\sim 1/\sqrt{\nu}$ (so $\nu = d + 1$ gives strongly correlated noise and $\nu = 10d$ near-diagonal noise). The noise bin selects the variance band: low $= [0, 1]$, high $= [4, 5]$, with per-feature variances drawn iid uniformly within the chosen band.

B. Per-axis performance

We report mean and median MSE on $h \leq 5$ extrapolation across the three axes that drive most of the qualitative differences in our sweep: linearity, loading distribution, and relative rank. For each axis we report results on clean and noisy targets separately. CIs are 95% closed-form (mean: CLT; median: order-statistics binomial). The best baseline per row is shown in **bold**.

B.1. Linearity ($\alpha = \beta$)

In Tables 5 and 6 we report performance by linearity coefficient $\alpha = \beta$, where $\alpha = 0$ is fully linear and $\alpha = 1$ is fully nonlinear. We find that **Simplex** performs the best in the mostly-linear regime ($\alpha \in \{0.25, 0.5\}$),, while **TabPFN** performs better in nonlinear regimes ($\alpha \in \{0.75, 1.0\}$). The pattern is consistent across mean and median and across clean and noisy targets.

Table 5. Linearity coefficient ($\alpha = \beta$) – MSE (clean targets, $h \leq 5$). Each axis value spans two rows: Mean (top) and Median (bottom). Best per row in **bold**; 95% CIs in grey.

α	Stat	RSC	Lasso	Simplex	TabPFN	TabPFN-TS	Chronos
0.00	Mean	12.04 (11.26, 12.81)	8.00 (7.68, 8.32)	12.29 (10.79, 13.78)	**5.66** (5.46, 5.86)	6.14 (5.96, 6.32)	6.89 (6.70, 7.08)
	Median	1.53 (1.50, 1.56)	1.59 (1.55, 1.63)	**1.08** (1.06, 1.10)	1.18 (1.16, 1.21)	1.30 (1.27, 1.32)	1.43 (1.41, 1.46)
0.25	Mean	6.33 (6.01, 6.64)	4.42 (4.30, 4.53)	**1.70** (1.68, 1.73)	1.88 (1.84, 1.91)	2.13 (2.10, 2.17)	2.05 (2.02, 2.08)
	Median	1.04 (1.02, 1.05)	1.02 (0.99, 1.04)	**0.60** (0.59, 0.61)	0.60 (0.59, 0.62)	0.68 (0.67, 0.69)	0.67 (0.66, 0.69)
0.50	Mean	5.56 (5.23, 5.88)	3.48 (3.40, 3.56)	**1.13** (1.11, 1.14)	1.16 (1.14, 1.18)	1.34 (1.32, 1.35)	1.27 (1.25, 1.28)
	Median	0.90 (0.89, 0.92)	0.82 (0.79, 0.84)	0.48 (0.47, 0.49)	**0.43** (0.42, 0.44)	0.49 (0.48, 0.50)	0.47 (0.46, 0.48)
0.75	Mean	4.96 (4.71, 5.21)	3.01 (2.94, 3.09)	0.91 (0.90, 0.92)	**0.86** (0.84, 0.87)	1.02 (1.00, 1.03)	0.93 (0.92, 0.94)
	Median	0.84 (0.83, 0.86)	0.73 (0.71, 0.75)	0.43 (0.43, 0.44)	**0.36** (0.36, 0.37)	0.42 (0.41, 0.42)	0.38 (0.38, 0.39)
1.00	Mean	4.65 (4.47, 4.83)	2.95 (2.89, 3.01)	0.90 (0.89, 0.91)	**0.82** (0.81, 0.83)	0.98 (0.97, 1.00)	0.89 (0.88, 0.91)
	Median	0.87 (0.85, 0.88)	0.78 (0.76, 0.80)	0.48 (0.47, 0.49)	**0.41** (0.40, 0.41)	0.46 (0.46, 0.47)	0.43 (0.42, 0.44)

Table 6. Linearity coefficient ($\alpha = \beta$) – MSE (noisy targets, $h \leq 5$). Each axis value spans two rows: Mean (top) and Median (bottom). Best per row in **bold**; 95% CIs in grey.

α	Stat	RSC	Lasso	Simplex	TabPFN	TabPFN-TS	Chronos
0.00	Mean	14.13 (13.37, 14.90)	9.84 (9.52, 10.17)	14.59 (13.10, 16.07)	**7.81** (7.61, 8.01)	8.35 (8.17, 8.53)	9.17 (8.97, 9.37)
	Median	3.15 (3.09, 3.21)	**2.49** (2.43, 2.55)	2.68 (2.63, 2.74)	2.52 (2.47, 2.57)	2.71 (2.66, 2.76)	3.00 (2.93, 3.06)
0.25	Mean	8.34 (8.02, 8.66)	6.18 (6.06, 6.31)	**3.92** (3.88, 3.97)	4.01 (3.96, 4.06)	4.35 (4.30, 4.40)	4.27 (4.23, 4.32)
	Median	2.33 (2.28, 2.37)	1.83 (1.79, 1.87)	1.72 (1.69, 1.76)	**1.71** (1.67, 1.74)	1.85 (1.81, 1.89)	1.84 (1.80, 1.88)
0.50	Mean	7.50 (7.18, 7.83)	5.21 (5.12, 5.30)	3.31 (3.28, 3.34)	**3.27** (3.24, 3.31)	3.54 (3.51, 3.58)	3.48 (3.45, 3.52)
	Median	2.12 (2.07, 2.16)	1.64 (1.60, 1.68)	1.54 (1.51, 1.57)	**1.50** (1.47, 1.53)	1.63 (1.60, 1.66)	1.62 (1.59, 1.65)
0.75	Mean	6.86 (6.61, 7.12)	4.73 (4.64, 4.81)	3.06 (3.04, 3.09)	**2.96** (2.93, 2.99)	3.22 (3.19, 3.25)	3.14 (3.11, 3.17)
	Median	2.03 (1.99, 2.08)	1.56 (1.53, 1.59)	1.46 (1.43, 1.48)	**1.41** (1.39, 1.44)	1.54 (1.51, 1.56)	1.52 (1.49, 1.55)
1.00	Mean	6.56 (6.38, 6.75)	4.65 (4.58, 4.73)	3.06 (3.03, 3.08)	**2.92** (2.90, 2.95)	3.19 (3.15, 3.22)	3.10 (3.08, 3.13)
	Median	2.06 (2.03, 2.10)	1.60 (1.57, 1.64)	1.49 (1.46, 1.51)	**1.43** (1.41, 1.46)	1.57 (1.54, 1.60)	1.54 (1.52, 1.57)

B.2. Loading Distribution

In Tables 7 and 8 we report performance by loading distribution. **Simplex** performs the best on Dirichlet loadings, where the data-generating weights live close to the simplex, while **TabPFN** dominates on Gaussian loadings, which produce dense, sign-mixed weights.

B.3. Relative Rank ($d/\min(n, T)$)

In Tables 9 and 10 we report performance by relative rank $d/\min(n, T)$. **RSC** performs the best at the deeply over-determined corner ($d/\min(n, T) = 0.04$), where its additional denoising provides benefits, while **TabPFN** dominates at every higher relative rank $d/\min(n, T) = 1$.

Table 7. Loading distribution – MSE (clean targets, $h \leq 5$). Each axis value spans two rows: Mean (top) and Median (bottom). Best per row in **bold**; 95% CIs in grey.

Loading	Stat	RSC	Lasso	Simplex	TabPFN	TabPFN-TS	Chronos
Dir(5)	Mean	5.13 (4.86, 5.41)	3.19 (3.14, 3.25)	**0.84** (0.83, 0.85)	1.16 (1.14, 1.17)	1.29 (1.27, 1.30)	1.42 (1.40, 1.44)
	Median	0.79 (0.78, 0.80)	0.74 (0.73, 0.76)	**0.35** (0.35, 0.36)	0.39 (0.38, 0.39)	0.44 (0.44, 0.45)	0.44 (0.43, 0.44)
Gaussian	Mean	8.28 (8.01, 8.54)	5.55 (5.42, 5.69)	5.92 (5.33, 6.52)	**2.99** (2.91, 3.07)	3.36 (3.29, 3.43)	3.39 (3.32, 3.47)
	Median	1.25 (1.24, 1.26)	1.18 (1.16, 1.20)	0.84 (0.83, 0.85)	**0.66** (0.65, 0.66)	0.75 (0.74, 0.76)	0.70 (0.69, 0.70)

Table 8. Loading distribution – MSE (noisy targets, $h \leq 5$). Each axis value spans two rows: Mean (top) and Median (bottom). Best per row in **bold**; 95% CIs in grey.

Loading	Stat	RSC	Lasso	Simplex	TabPFN	TabPFN-TS	Chronos
Dir(5)	Mean	7.11 (6.84, 7.38)	4.93 (4.87, 4.99)	**3.00** (2.98, 3.01)	3.26 (3.23, 3.29)	3.50 (3.48, 3.52)	3.66 (3.64, 3.69)
	Median	2.04 (2.01, 2.07)	1.55 (1.53, 1.57)	**1.38** (1.36, 1.39)	1.46 (1.44, 1.47)	1.58 (1.56, 1.60)	1.64 (1.61, 1.66)
Gaussian	Mean	10.25 (9.99, 10.52)	7.32 (7.18, 7.45)	8.18 (7.58, 8.78)	**5.13** (5.05, 5.21)	5.56 (5.49, 5.64)	5.61 (5.53, 5.69)
	Median	2.58 (2.55, 2.61)	2.06 (2.03, 2.08)	2.11 (2.08, 2.13)	**1.89** (1.86, 1.91)	2.04 (2.02, 2.07)	2.04 (2.01, 2.06)

B.4. Noise (magnitude and structure)

In Tables 11 and 12 we report performance by noise magnitude (low/high) and noise structure (random for diagonal R, correlated for Wishart R pooling both concentration variants).

Table 9. Relative rank $d/\min(n,T)$ – MSE (clean targets, $h \leq 5$). Each axis value spans two rows: Mean (top) and Median (bottom). Best per row in **bold**; 95% CIs in grey.

$d/\min(n,T)$	Stat	RSC	Lasso	Simplex	TabPFN	TabPFN-TS	Chronos
0.04	Mean	**0.61** (0.59, 0.63)	2.69 (2.60, 2.79)	10.50 (4.73, 16.27)	0.82 (0.79, 0.86)	0.82 (0.79, 0.85)	1.21 (0.94, 1.48)
	Median	**0.26** (0.24, 0.28)	0.63 (0.53, 0.77)	0.27 (0.25, 0.29)	0.34 (0.33, 0.36)	0.35 (0.33, 0.36)	0.33 (0.31, 0.34)
0.1	Mean	1.14 (1.12, 1.16)	2.89 (2.83, 2.95)	3.11 (2.25, 3.96)	**0.97** (0.94, 1.00)	1.03 (1.01, 1.05)	1.25 (1.19, 1.30)
	Median	0.43 (0.42, 0.44)	0.69 (0.65, 0.74)	0.37 (0.36, 0.38)	**0.35** (0.34, 0.36)	0.37 (0.36, 0.38)	0.41 (0.39, 0.41)
0.2	Mean	3.60 (3.48, 3.72)	3.48 (3.38, 3.58)	3.57 (2.52, 4.63)	**1.59** (1.51, 1.67)	1.78 (1.72, 1.85)	2.02 (1.94, 2.10)
	Median	0.89 (0.87, 0.91)	0.83 (0.80, 0.86)	0.49 (0.48, 0.50)	**0.48** (0.47, 0.49)	0.55 (0.54, 0.56)	0.55 (0.54, 0.56)
0.25	Mean	1.44 (1.41, 1.47)	2.96 (2.88, 3.03)	1.81 (1.61, 2.02)	**1.07** (1.04, 1.10)	1.21 (1.17, 1.25)	1.28 (1.23, 1.32)
	Median	0.56 (0.55, 0.58)	0.77 (0.72, 0.81)	0.43 (0.42, 0.45)	**0.38** (0.37, 0.39)	0.43 (0.42, 0.45)	0.42 (0.41, 0.44)
0.4	Mean	13.37 (12.49, 14.25)	5.31 (5.04, 5.58)	3.35 (2.98, 3.72)	**2.76** (2.61, 2.90)	3.13 (3.00, 3.26)	3.53 (3.38, 3.69)
	Median	1.48 (1.45, 1.51)	1.00 (0.98, 1.03)	**0.64** (0.62, 0.65)	0.64 (0.62, 0.65)	0.75 (0.74, 0.77)	0.73 (0.72, 0.75)
0.5	Mean	3.61 (3.51, 3.72)	3.88 (3.80, 3.97)	2.49 (2.31, 2.68)	**1.64** (1.59, 1.68)	1.91 (1.86, 1.96)	1.96 (1.90, 2.02)
	Median	0.96 (0.94, 0.98)	0.92 (0.90, 0.95)	0.56 (0.55, 0.57)	**0.49** (0.48, 0.50)	0.54 (0.54, 0.55)	0.52 (0.51, 0.53)
1	Mean	8.99 (8.68, 9.30)	5.28 (5.12, 5.43)	3.71 (3.10, 4.32)	**2.72** (2.63, 2.80)	2.99 (2.92, 3.07)	2.81 (2.74, 2.88)
	Median	1.39 (1.36, 1.40)	1.08 (1.06, 1.10)	0.68 (0.67, 0.69)	**0.57** (0.56, 0.58)	0.66 (0.65, 0.67)	0.60 (0.60, 0.61)

Table 10. Relative rank $d/\min(n,T)$ – MSE (noisy targets, $h \leq 5$). Each axis value spans two rows: Mean (top) and Median (bottom). Best per row in **bold**; 95% CIs in grey.

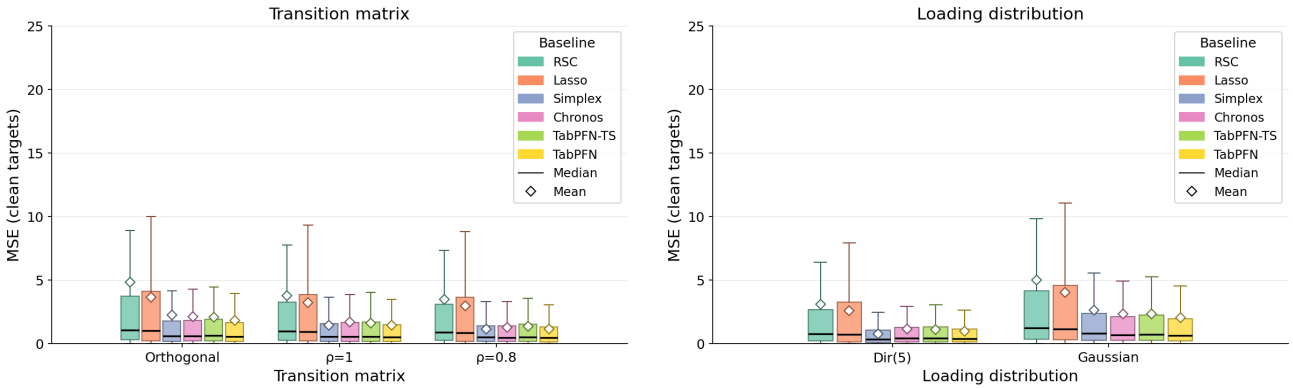
$d/\min(n,T)$	Stat	RSC	Lasso	Simplex	TabPFN	TabPFN-TS	Chronos
0.04	Mean	2.91 (2.83, 2.99)	3.91 (3.78, 4.04)	12.67 (6.91, 18.42)	**2.77** (2.69, 2.85)	2.84 (2.76, 2.92)	3.48 (3.19, 3.75)
	Median	1.44 (1.35, 1.53)	1.36 (1.28, 1.45)	1.37 (1.28, 1.45)	**1.32** (1.25, 1.41)	1.36 (1.29, 1.45)	1.49 (1.42, 1.57)
0.1	Mean	3.28 (3.23, 3.33)	4.43 (4.35, 4.51)	5.34 (4.47, 6.22)	**3.01** (2.96, 3.05)	3.16 (3.12, 3.21)	3.54 (3.47, 3.61)
	Median	1.56 (1.51, 1.61)	1.41 (1.36, 1.46)	1.50 (1.45, 1.55)	**1.39** (1.34, 1.43)	1.48 (1.44, 1.53)	1.59 (1.55, 1.64)
0.2	Mean	5.52 (5.39, 5.64)	5.14 (5.03, 5.25)	5.73 (4.67, 6.79)	**3.65** (3.56, 3.75)	3.94 (3.86, 4.02)	4.24 (4.15, 4.34)
	Median	2.09 (2.04, 2.15)	1.64 (1.60, 1.68)	1.57 (1.53, 1.61)	**1.52** (1.48, 1.56)	1.64 (1.60, 1.68)	1.73 (1.69, 1.77)
0.25	Mean	3.54 (3.48, 3.60)	4.45 (4.35, 4.54)	3.94 (3.73, 4.15)	**3.09** (3.04, 3.15)	3.31 (3.25, 3.37)	3.46 (3.40, 3.53)
	Median	1.67 (1.62, 1.74)	**1.42** (1.38, 1.45)	1.52 (1.47, 1.57)	1.44 (1.40, 1.49)	1.54 (1.49, 1.59)	1.57 (1.51, 1.63)
0.4	Mean	15.26 (14.39, 16.12)	7.26 (6.99, 7.53)	5.68 (5.31, 6.06)	**5.00** (4.85, 5.14)	5.44 (5.31, 5.58)	5.82 (5.66, 5.98)
	Median	2.81 (2.76, 2.87)	1.97 (1.93, 2.02)	**1.80** (1.76, 1.83)	1.88 (1.84, 1.91)	2.06 (2.02, 2.10)	2.08 (2.04, 2.13)
0.5	Mean	5.59 (5.48, 5.70)	5.57 (5.47, 5.67)	4.66 (4.47, 4.85)	**3.71** (3.66, 3.77)	4.10 (4.04, 4.16)	4.12 (4.05, 4.19)
	Median	2.17 (2.12, 2.21)	1.72 (1.69, 1.76)	1.73 (1.70, 1.77)	**1.59** (1.57, 1.63)	1.75 (1.72, 1.79)	1.74 (1.70, 1.77)
1	Mean	10.94 (10.62, 11.25)	7.14 (6.98, 7.30)	5.90 (5.29, 6.50)	**4.88** (4.79, 4.97)	5.23 (5.15, 5.31)	5.04 (4.96, 5.11)
	Median	2.67 (2.63, 2.71)	2.04 (2.00, 2.07)	1.84 (1.81, 1.86)	**1.80** (1.77, 1.83)	1.94 (1.91, 1.97)	1.93 (1.90, 1.96)

Table 11. Noise (magnitude / structure) – MSE (clean targets, $h \leq 5$). Each row spans two lines: Mean (top) and Median (bottom). Best per row in **bold**; 95% CIs in grey.

Noise (mag / struct)	Stat	RSC	Lasso	Simplex	TabPFN	TabPFN-TS	Chronos
Low / Random	Mean	1.05 (1.02, 1.09)	0.47 (0.46, 0.48)	0.55 (0.53, 0.57)	**0.44** (0.42, 0.46)	0.49 (0.47, 0.50)	0.51 (0.50, 0.53)
	Median	0.29 (0.29, 0.30)	0.21 (0.21, 0.21)	0.18 (0.18, 0.18)	**0.17** (0.17, 0.17)	0.20 (0.19, 0.20)	0.20 (0.20, 0.20)
Low / Correlated	Mean	0.94 (0.93, 0.96)	0.46 (0.45, 0.47)	0.60 (0.55, 0.65)	**0.43** (0.42, 0.44)	0.48 (0.47, 0.49)	0.52 (0.51, 0.53)
	Median	0.30 (0.30, 0.31)	0.21 (0.21, 0.21)	**0.18** (0.17, 0.18)	0.18 (0.18, 0.18)	0.20 (0.20, 0.20)	0.20 (0.20, 0.20)
High / Random	Mean	13.26 (12.64, 13.88)	8.93 (8.68, 9.18)	5.78 (5.08, 6.49)	**3.71** (3.57, 3.86)	4.12 (4.00, 4.24)	4.23 (4.10, 4.36)
	Median	2.74 (2.70, 2.78)	4.16 (4.11, 4.21)	1.46 (1.45, 1.48)	**1.23** (1.22, 1.25)	1.49 (1.47, 1.50)	1.37 (1.35, 1.38)
High / Correlated	Mean	12.02 (11.54, 12.49)	7.95 (7.78, 8.13)	6.39 (5.57, 7.21)	**3.72** (3.62, 3.81)	4.18 (4.09, 4.27)	4.33 (4.23, 4.42)
	Median	3.04 (3.00, 3.07)	3.79 (3.76, 3.82)	1.49 (1.48, 1.50)	**1.44** (1.43, 1.45)	1.64 (1.62, 1.65)	1.51 (1.50, 1.52)

Table 12. Noise (magnitude / structure) – MSE (noisy targets, $h \leq 5$). Each row spans two lines: Mean (top) and Median (bottom). Best per row in **bold**; 95% CIs in grey.

Noise (mag / struct)	Stat	RSC	Lasso	Simplex	TabPFN	TabPFN-TS	Chronos
Low / Random	Mean	1.55 (1.52, 1.59)	0.97 (0.95, 0.98)	1.05 (1.03, 1.07)	**0.94** (0.92, 0.96)	0.98 (0.97, 1.00)	1.01 (0.99, 1.02)
	Median	0.75 (0.74, 0.76)	0.62 (0.62, 0.63)	0.60 (0.60, 0.61)	**0.59** (0.58, 0.59)	0.62 (0.61, 0.63)	0.63 (0.62, 0.64)
Low / Correlated	Mean	1.30 (1.28, 1.32)	0.85 (0.84, 0.86)	1.02 (0.97, 1.07)	**0.82** (0.81, 0.83)	0.90 (0.89, 0.91)	0.94 (0.93, 0.96)
	Median	0.63 (0.63, 0.64)	0.53 (0.52, 0.53)	0.52 (0.52, 0.53)	**0.49** (0.49, 0.50)	0.55 (0.54, 0.55)	0.56 (0.56, 0.57)
High / Random	Mean	17.79 (17.17, 18.41)	13.35 (13.09, 13.60)	10.31 (9.59, 11.02)	**8.18** (8.03, 8.33)	8.60 (8.47, 8.73)	8.71 (8.57, 8.85)
	Median	7.60 (7.53, 7.67)	8.65 (8.57, 8.72)	5.82 (5.77, 5.86)	**5.66** (5.62, 5.71)	6.01 (5.96, 6.05)	5.87 (5.82, 5.91)
High / Correlated	Mean	15.07 (14.60, 15.54)	10.37 (10.19, 10.54)	10.07 (9.25, 10.88)	**7.21** (7.11, 7.31)	7.89 (7.80, 7.99)	8.10 (8.00, 8.20)
	Median	6.27 (6.22, 6.31)	5.86 (5.82, 5.91)	4.84 (4.81, 4.87)	**4.71** (4.68, 4.74)	5.16 (5.13, 5.20)	5.19 (5.16, 5.22)



(a) Transition kind *(b)* Loading distribution

Figure 4. Transition and loading ablations.

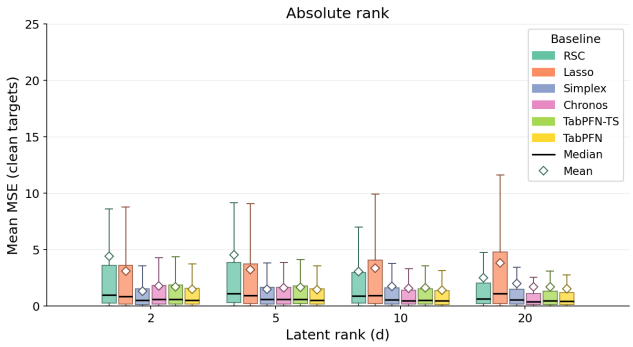
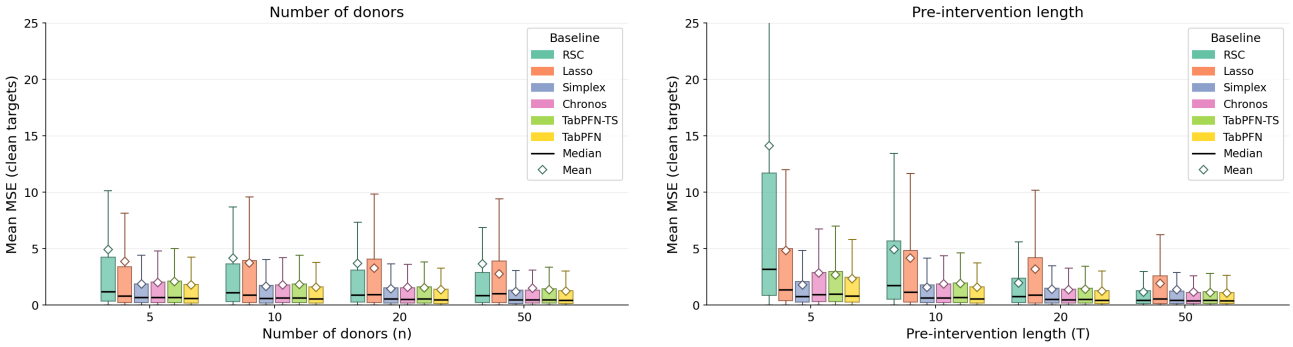


Figure 5. Absolute rank (d).

B.5. Additional Axis Plots (Clean Targets)

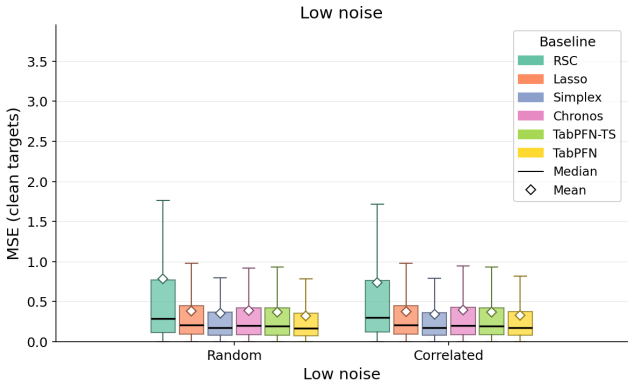
We provide further breakdowns of our results according to transition matrix structure and latent loading distribution in Figure 4. We find that performance is consistent under parameterizations of the transition matrix. We find that when the latent loading distribution is sparse (Dirichlet distribution with $\kappa = 5$), simplex achieves the best performance, and when the latent loading distribution is dense (Gaussian), the foundation models achieve better performance.

We plot results according to absolute rank in Figure 5, and by the number of donors and time steps in Figure 6. We find that in the lowest absolute rank setting (2), simplex performs the best, and foundation models outperform linear methods in high rank settings. We find that method ranking is consistent under increasing number of donors, and more varied under increasing time steps.

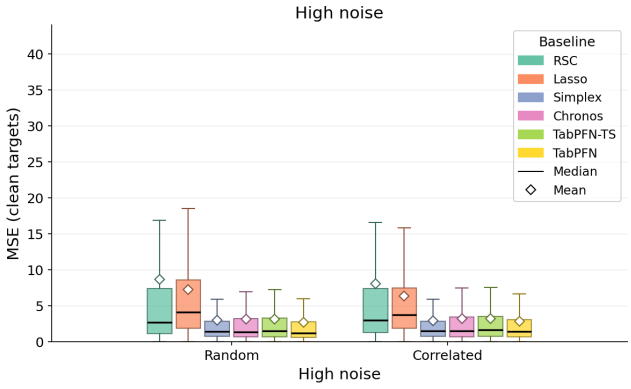


(a) n_{donors} *(b)* T_{pre}

Figure 6. Donor count and pre-period length ablations.



(a) Low noise



(b) High noise

Figure 7

B.6. Plots Compared to Noisy Targets

Below, we provide replications of our results when compared to noisy targets rather than the clean underlying signal.

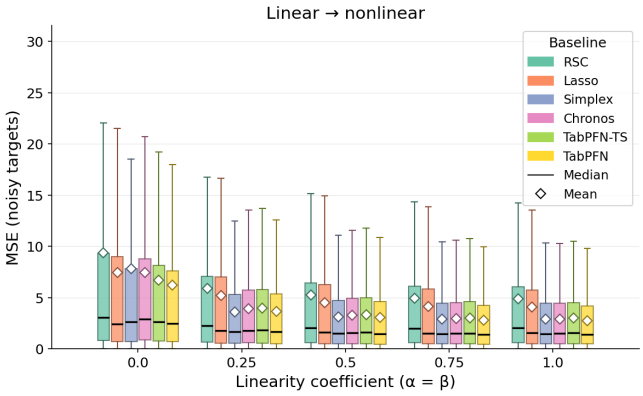
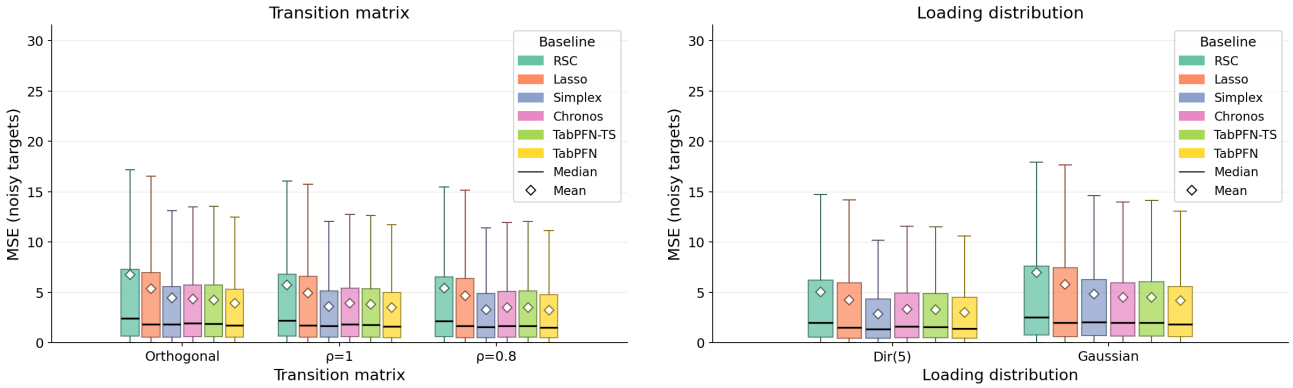


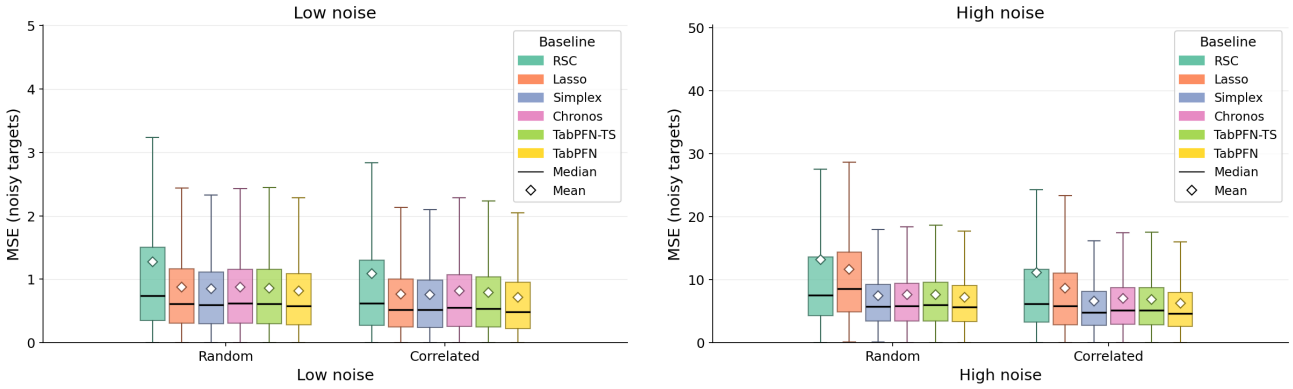
Figure 8. Linearity (α).



(a) Transition kind

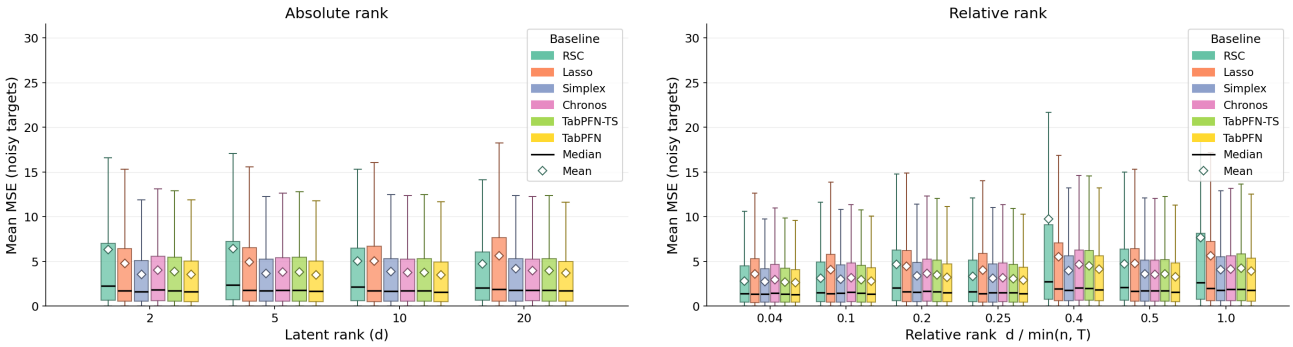
(b) Loading distribution

Figure 9. Transition and loading ablations.



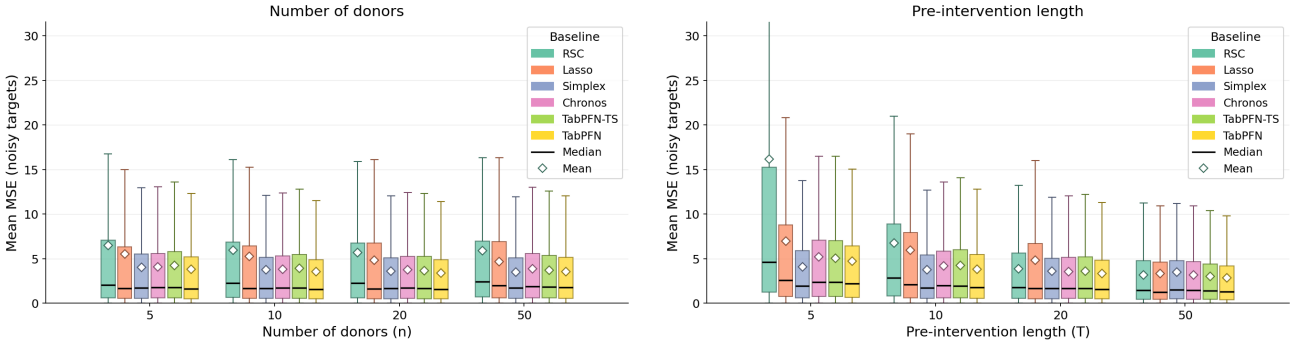
(a) Low noise *(b)* High noise

Figure 10. Noise-structure ablations.



(a) Absolute rank (d) *(b)* Relative rank

Figure 11. Rank ablations.



(a) n_{donors} *(b)* T_{pre}

Figure 12. Donor count and pre-period length ablations.