GLOBAL SHORTCUT CONTAMINATION IN END-TO-END GRAPH LEARNING FOR MULTIVARIATE FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph neural networks for multivariate time-series forecasting increasingly learn graph structures end-to-end, jointly optimizing latent relational structures and forecasting models from data. Although such learned graphs are often treated as variable dependency structures, the end-to-end forecasting objective does not inherently support their structural correctness. As a result, the learned graph should be viewed as a prediction-optimized structure, which may fail to capture invariant local dependencies. Here we analyze why and when this mismatch arises. We show that, under the stated latent-component conditions, persistent global predictive components can make end-to-end graph learning favor train-specific shortcut structures, and we characterize when such shortcut-contaminated graphs are preferred. Based on this analysis, we propose a simple projection-based decoupling strategy that removes dominant global components before graph learning in an architecture-agnostic manner. Experiments on controlled synthetic benchmarks and real traffic benchmarks under time-of-day shifts show that projection can improve robustness for shortcut-prone graph learners, while its effects are architecture- and shift-dependent. Beyond out-of-distribution (OOD) robustness, our results highlight the importance of separating predictive usefulness from structural correctness when interpreting learned graphs in end-to-end forecasting models.

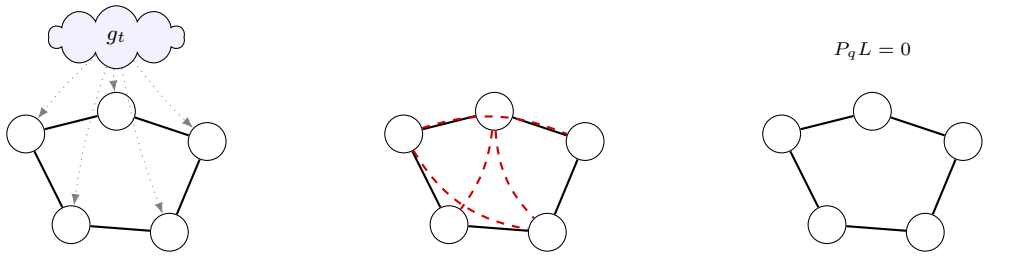
1 INTRODUCTION

Graph-based models exploit inter-variable or spatial dependencies for multivariate time-series forecasting (Yu et al., 2018; Li et al., 2018). Recent approaches learn latent, adaptive, or dynamic graphs jointly with forecasting models (Wu et al., 2019; 2020; Bai et al., 2020; Shang et al., 2021; Li et al., 2021; Satorras et al., 2022; Xu et al., 2023). The learned graph determines which variables exchange information, but its training signal rewards accurate forecasts rather than correct local dependencies. An edge can therefore be useful for prediction without representing an interaction that remains valid across environments. This distinction matters both when interpreting the learned relations and when relying on them for forecasting under distribution shift.

Consider sensors whose observations combine local dynamics with a persistent shared component. Measurements outside a sensor's local neighborhood can help estimate this temporally predictive component, making additional edges useful even without corresponding local interactions. Forecasting loss may then reward a graph that encodes the shared training component. If its loading across sensors changes, that predictive advantage can become a liability even when the local dynamics remain unchanged. Thus, good in-distribution forecasts need not reveal whether the learned graph relies on stable local information or on an environment-specific shortcut.

Related work develops sparse probabilistic graph learning (Cini et al., 2023), functional relation constraints (Li et al., 2024), and time-varying leading-indicator models (Zhao & Shen, 2024). Cini et al. (2025) provide a methodological framework for graph-based forecasting, including model design and evaluation. Our focus is complementary: *when can forecasting loss favor shortcut-augmented graphs over invariant local support, and when does this reliance cause excess test risk?* We analyze these questions under explicit latent-component assumptions and distinguish population-risk preference from conditional local optimization effects, rather than assuming that nonlinear training recovers a unique graph.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) True DGP: $A^\star$ + global Lg_t **(b) Learned:** shortcut-contaminated graph **(c) After P_q:** shortcuts reduced

Figure 1: **Conceptual illustration of shortcut contamination and projection-based mitigation.** Persistent global factors can induce train-specific shortcut edges. Projection removes the dominant shortcut source and motivates the proposed decoupling method.

To reduce this shortcut access, we introduce projection-based decoupling *before* graph learning. The rationale is to remove the shared input direction that makes extra cross-variable edges useful for estimating the global component. Under the model's orthogonality condition, exact projection removes that component while preserving the latent local signal. MeanProj targets a homogeneous shared direction; PCA estimates a dominant direction from training data when loadings are heterogeneous. We train each backbone on projected inputs and targets and complete the removed coordinate using last-observation persistence, separating learned residual prediction from a fixed rule for the removed component. Comparing the original and projected pipelines probes sensitivity to shared variation without redesigning the backbone. Approximate directions can remove local signal or leave shared variation, so neither graph recovery nor uniform robustness gains are guaranteed.

The main contributions of this study are summarized as follows:

- We identify a failure mode in end-to-end graph learning for multivariate forecasting: minimizing forecasting loss can favor train-specific shortcut graphs rather than invariant local dependency structure.
- We derive a one-step population-risk comparison, identify conditions for local shortcut-aligned descent, and characterize when changes in the global loading direction cause excess test risk.
- Based on these theoretical insights, we propose a simple architecture-agnostic projection-based diagnostic and mitigation strategy and show that it can stabilize shortcut-prone end-to-end graph learners under distribution shift, while its effects on already-stable models are shift-dependent.

2 Theoretical Analysis

Throughout this section, a graph is defined over N components of $x_t \in \mathbb{R}^N$: each node is a variable, sensor, or time series, and an edge $j \to i$ means that a node j can contribute to predicting a node i through graph-mediated aggregation. We allow directed graphs, so M need not be symmetric; undirected graphs are covered as the special symmetric case. We use a support mask $M \in \{0,1\}^{N\times N}$ with $M_{ij} = 1$ denoting $j \to i$, define $S_i(M) = \{j : M_{ij} = 1\}$ and $\bar{S}_i(M) = \{1,\dots,N\} \setminus S_i(M)$, and write $\mathrm{supp}(A) = \{(i,j) : A_{ij} \neq 0\}$ for the graph support of a real-valued matrix A.

Assume $Y = x_{t+1}$ for simplicity, and write $X_{t-T:t} := (x_{t-T},\dots,x_t)$ for the input history, where T is the look-back extent. Representative end-to-end graph-learning forecasters jointly learn a graph generator and a forecasting network from prediction loss (Wu et al., 2019; 2020; Bai et al., 2020; Shang et al., 2021). Abstracting these methods, we write

$$(\theta^\star, \phi^\star) \in \arg\min_{\theta,\phi} \mathbb{E}\left[\|Y - f_\theta(X_{t-T:t}, \mathcal{A}_\phi(X_{t-T:t}))\|_2^2\right], \tag{1}$$

where $\mathcal{A}_\phi$ is a learned graph generator and f_θ is a forecasting network that maps the input history and the generated graph to a prediction of Y. We distinguish the true local dependency support $M^\star$ from shortcut-contaminated graphs that include directions aligned with train-specific global components.

To isolate the mechanism, consider the latent decomposition

$$x_t = z_t + Lg_t, \qquad \|L\|_2 = 1, \tag{2}$$

where z_t contains invariant local interactions and Lg_t is a rank-one global component. The local component evolves according to an autoregression of order p,

$$z_{t+1} = \sum_{\tau=0}^{p-1} B_\tau^\star z_{t-\tau} + \eta_{t+1}, \tag{3}$$

where $B_\tau^\star \in \mathbb{R}^{N\times N}$ is the lag-τ local interaction matrix, $\tau \in \{0,\dots,p-1\}$ indexes the lag, and $\eta_{t+1} \in \mathbb{R}^N$ denotes the local innovation noise. The global factor follows

$$g_{t+1} = \rho g_t + \nu_{t+1}, \qquad |\rho| < 1. \tag{4}$$

The true local support mask is defined entrywise as

$$(M^\star)_{ij} = \mathbf{1}\left\{\sum_{\tau=0}^{p-1} |(B_\tau^\star)_{ij}| > 0\right\}. \tag{5}$$

Thus, local dependencies are stable across environments, whereas the global component is predictive but may be environment-specific.

Forecasting loss can prefer shortcut graphs. For a support mask M, define the population forecasting risk

$$\mathcal{J}(M) = \inf_{\{D_\tau\},\{\Theta_\tau\}} \mathbb{E}\left[\left\| x_{t+1} - \sum_{\tau=0}^{p-1} D_\tau x_{t-\tau} - \sum_{\tau=0}^{p-1} (\Theta_\tau \odot M) x_{t-\tau} \right\|_2^2\right], \tag{6}$$

where D_τ is diagonal and Θ_τ is unrestricted. The diagonal pathway captures node-wise temporal prediction, while the masked pathway captures graph-mediated cross-node aggregation. Consequently, the variables available to node i are

$$R_i(M) := \{i\} \cup S_i(M), \qquad \bar{R}_i(M) := \{1,\dots,N\} \setminus R_i(M). \tag{7}$$

These effective sets include self-information even when $M_{ii} = 0$.

Let M_L be the shortcut-augmented support mask,

$$(M_L)_{ij} = (M^\star)_{ij} \vee \mathbf{1}\{(LL^\top)_{ij} \neq 0\}, \tag{8}$$

which adds loading-aligned shortcut edges to the true local support. Under the one-step factorized history distribution

$$z_t \sim \mathcal{N}\big(0, \sigma_z^2(I - LL^\top)\big), \qquad g_t \sim \mathcal{N}(0,\sigma_g^2), \qquad z_t \perp\!\!\!\perp g_t, \tag{9}$$

we obtain the following result.

Theorem 1 (Shortcut contamination in one-step population risk)**.** *Assume $p = 1$, write $B = B_0^\star$, and let $\sigma_z^2, \sigma_g^2 > 0$. Assume that both innovations have finite second moments and conditional mean zero given (z_t, g_t). Define $b := \rho L - BL$. Then*

$$\mathcal{J}(M^\star) - \mathcal{J}(M_L) = \sum_{i:L_i\neq 0} b_i^2 \frac{\sigma_z^2\sigma_g^2 \|L_{\bar{R}_i(M^\star)}\|_2^2}{\sigma_z^2\|L_{\bar{R}_i(M^\star)}\|_2^2 + \sigma_g^2\|L_{R_i(M^\star)}\|_2^2}. \tag{10}$$

The gap is strictly positive if and only if some node i satisfies $L_i \neq 0$, $b_i \neq 0$, and $\|L_{\bar{R}_i(M^\star)}\|_2^2 > 0$, where $\|L_S\|_2^2 := \sum_{j\in S} L_j^2$.

The proof is given in Appendix A.3. The coefficient b follows from the original latent dynamics: $Bz_t + \rho L g_t = Bx_t + (\rho L - BL)g_t$. Thus, the local term cannot generally be treated as Bx_t while retaining the coefficient ρL on the factor. When $BL = 0$, $b = \rho L$, and persistence yields a strictly positive gap whenever $\rho \neq 0$ and a node with nonzero loading lacks access to some loading mass outside its effective neighborhood. Appendix A.3 gives a stationary example with nonzero local interactions. The result compares unregularized population risks in the stated linear class. It shows that prediction risk alone need not privilege the true local support, not that a unique graph is recovered or that every nonlinear training procedure selects M_L. The closed form is restricted to $p = 1$; the multi-lag alignment calculation below is a separate statement.

Shortcut alignment during optimization. To examine a possible local optimization mechanism, consider a response surrogate around a reference graph: $\Gamma_\tau^{(0)}$ is the fixed lag-specific output-response map at that reference point, and U^g is a shortcut perturbation direction in adjacency space. The associated first-order output change is

$$\phi^g = \sum_{\tau=0}^{p-1} \Gamma_\tau^{(0)} U^g x_{t-\tau}. \tag{11}$$

If g_t is a zero-mean covariance-stationary AR(1) process and $\mathbb{E}[g_{t+1} z_{t-\tau}] = 0$ for each lag, then

$$\mathbb{E}\left[\langle L g_{t+1},\ \phi^g \rangle\right] = \mathrm{Var}(g_t) \sum_{\tau=0}^{p-1} \rho^{\tau+1} L^\top \Gamma_\tau^{(0)} U^g L. \tag{12}$$

A positive right-hand side establishes alignment with the global target component, not automatically with the current forecasting residual. For the surrogate $f_\alpha = f_0 + \alpha\phi^g$,

$$\frac{d}{d\alpha} \mathbb{E}\|x_{t+1} - f_\alpha\|_2^2 \Big|_{\alpha=0} = -2\mathbb{E}\langle x_{t+1} - f_0, \phi^g \rangle. \tag{13}$$

The shortcut direction is locally loss-decreasing when this residual alignment is positive. Appendix A.5 gives the exact surrogate loss change and explains the additional condition needed to infer descent from Eq. (12); neither identity alone guarantees gradient starvation or a nonlinear training trajectory.

Shift fragility. Shortcut reliance can become harmful when the global loading direction changes. Consider the one-step populations

$$x_t^e = z_t^e + L_e g_t^e, \quad \|L_e\|_2 = 1, \quad e \in \{\mathrm{tr}, \mathrm{te}\}, \tag{14}$$

with shared $B = B_0^\star$ and ρ. In each environment, use Eq. (9) with L_e, the same positive scalar variances, and conditionally mean-zero innovations as in Theorem 1. Define

$$Q_e = L_e L_e^\top, \quad P_e = I - Q_e, \quad C_e = BP_e + \rho Q_e. \tag{15}$$

Here $\widehat{f}^{\mathrm{tr}}_{\mathrm{short}}(X_{t-T:t}) = C_{\mathrm{tr}} x_t$ is the population least-squares fit over all current variables; the oracle is $f^{\mathrm{te}}_{\mathrm{oracle}}(X_{t-T:t}) = C_{\mathrm{te}} x_t$. Both use only x_t, and the train-fitted coefficients are frozen on test data. The train fit is attainable under the shortcut mask whenever that mask represents C_{tr}, in particular when every entry of L_{tr} is nonzero. This is not an assumption that a finite-sample nonlinear backbone equals its population fit. Writing $\mathcal{R}_{\mathrm{te}}(f) := \mathbb{E}_{\mathrm{te}}\|x_{t+1}^{\mathrm{te}} - f(X_{t-T:t}^{\mathrm{te}})\|_2^2$ and $\mathcal{R}_{\mathrm{te}}^{\mathrm{oracle}} := \mathcal{R}_{\mathrm{te}}(f^{\mathrm{te}}_{\mathrm{oracle}})$, we obtain:

Theorem 2 (Shift fragility of population-fitted shortcut predictors)**.** *Under the stated one-step conditions, let* $F = (\rho I - B)(Q_{\mathrm{tr}} - Q_{\mathrm{te}})$ *and* $\Sigma_{\mathrm{te}} = \sigma_z^2 P_{\mathrm{te}} + \sigma_g^2 Q_{\mathrm{te}}$*. Then*

$$\mathcal{R}_{\mathrm{te}}(\widehat{f}^{\mathrm{tr}}_{\mathrm{short}}) - \mathcal{R}_{\mathrm{te}}^{\mathrm{oracle}} = \mathrm{tr}(F\Sigma_{\mathrm{te}}F^\top), \tag{16}$$

which is strictly positive if and only if $F \neq 0$*. In particular, if* $BL_{\mathrm{tr}} = BL_{\mathrm{te}} = 0$*, then*

$$\mathcal{R}_{\mathrm{te}}(\widehat{f}^{\mathrm{tr}}_{\mathrm{short}}) - \mathcal{R}_{\mathrm{te}}^{\mathrm{oracle}} = \rho^2(\sigma_g^2 + \sigma_z^2)\sin^2\theta, \tag{17}$$

where $\theta = \arccos|L_{\mathrm{tr}}^\top L_{\mathrm{te}}|$ *is the angle between the loading subspaces.*

Appendix A.6 gives the complete proof and the mask-representability condition. The angular formula requires the additional annihilation conditions; in general, shared local dynamics do not cancel from the risk gap. For example, $B = \rho I$ gives $F = 0$ even under a nontrivial rotation. The result is a sufficient fragility mechanism, not a guarantee that every loading shift raises risk. The specified local history covariance depends on L_e; the shared quantities are B, ρ, and the scalar variance parameters.

Projection preserves local structure while suppressing shortcut access. The analysis suggests a simple mitigation: remove the dominant global direction before graph learning. For a unit vector q, define

$$P_q = I - qq^\top, \qquad \widetilde{x}_t = P_q x_t. \tag{18}$$

Importantly, this operation is not merely generic preprocessing. Under the latent-component model, exact projection removes the global shortcut component while preserving the local component on which the invariant dependency structure is defined.

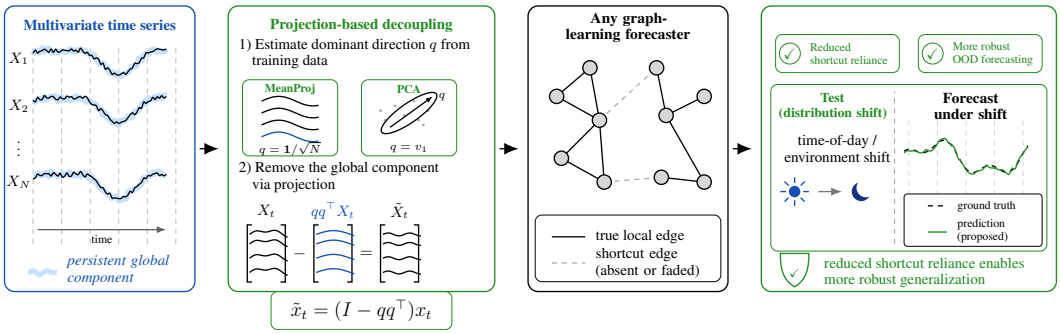


Figure 2: **Projection-based decoupling.** Given a multivariate time series, we first choose a projection direction q: a fixed spatial-mean direction for MeanProj or a top principal direction estimated from the training data for PCA. We then remove the corresponding global component before feeding the transformed sequence into an arbitrary graph-learning forecaster. The backbone is trained on the projected target, and the removed coordinate is completed causally from the last observation at evaluation. This projected-series pipeline reduces access to train-specific shortcut directions and encourages the forecaster to rely more on stable local dependencies under distribution shift.

Proposition 1 (Structure preservation under exact projection)**.** *Assume $x_t = z_t + Lg_t$, $\|L\|_2 = 1$, and $L^\top z_t = 0$ for all t. For $P_L = I - LL^\top$, we have $P_L x_t = z_t$. Consequently, exact projection removes the global component while preserving the local dynamics and support of z_t.*

The proof is given in Appendix A.2.

For an approximate projection direction q, the projected signal becomes

$$\widetilde{x}_t = P_q z_t + P_q L g_t = z_t - qq^\top z_t + P_q L g_t. \tag{19}$$

Thus, approximation has two distinct effects: residual global-component leakage, controlled by $\|P_q L\|_2$, and removal of local signal aligned with q, measured by $\|qq^\top z_t\|_2$. Even when $q = L$, failure of the sufficient orthogonality condition gives $P_L x_t = z_t - LL^\top z_t$. MeanProj and PCA are therefore practical proxies rather than guarantees of exact separation on arbitrary real data. In particular, if $q = L$, then $P_q L = 0$ and the dominant global shortcut direction is removed; the latent local component is preserved exactly only under the proposition's orthogonality condition. More generally, the projected shortcut alignment obeys

$$\left|\mathbb{E}\left[\left\langle Lg_{t+1},\ \phi_q^g \right\rangle\right]\right| \le \mathrm{Var}(g_t) \sum_{\tau=0}^{p-1} |\rho|^{\tau+1} \|\Gamma_\tau^{(0)} U^g\|_{\mathrm{op}} \|P_q L\|_2. \tag{20}$$

Exact projection has the stated preservation property under the orthogonality assumption; an approximate or fixed training projector need not preserve all local signal or remove a rotated test loading. Neither case guarantees recovery of the true graph. This motivates applying projection before graph learning.

3 Projection-based decoupling

Motivated by the preceding analysis, we propose a structure-preserving decoupling intervention that removes the dominant global shortcut direction before graph learning while preserving the latent local component in the ideal exact-projection case. The goal is to alter the representation seen by the graph learner, not to redesign representative graph-learning forecasting architectures such as Graph WaveNet, MTGNN, AGCRN, and DGCRN (Wu et al., 2019; 2020; Bai et al., 2020; Li et al., 2021).

3.1 Projected-series training and causal completion

Let $Q_q = qq^\top$ and $P_q = I - Q_q$. Projection is applied pointwise to the complete multivariate series before forecasting windows are constructed. For a history $X_{t-T:t}$ and an H-step target, define

$$\widetilde{X}_{t-T:t} := (P_q x_{t-T}, \ldots, P_q x_t), \qquad \widetilde{Y}_{t,H} := (P_q x_{t+1}, \ldots, P_q x_{t+H}).$$

The graph generator and forecasting module therefore both operate on the projected history, and each backbone retains its original architecture and forecasting-loss form in the projected coordinates:

$$\min_{\theta,\phi} \mathbb{E}\left[\sum_{h=1}^{H} \left\| P_q x_{t+h} - h_{\theta,\phi,h}(\widetilde{X}_{t-T:t}) \right\|_2^2\right], \quad h_{\theta,\phi} := f_\theta\Big(\widetilde{X}_{t-T:t}, \mathcal{A}_\phi(\widetilde{X}_{t-T:t})\Big). \tag{21}$$

For evaluation in the original coordinates, the implementation predicts the removed coordinate by causal last-observation persistence:

$$\widehat{x}_{t+h} = h_{\theta,\phi,h}(\widetilde{X}_{t-T:t}) + Q_q x_t, \qquad h = 1, \dots, H. \tag{22}$$

The term $Q_q x_t$ uses no future target and is not an inverse of the non-invertible projection. Consequently, raw-coordinate forecasting errors evaluate the complete pipeline—the learned projected-series predictor together with the fixed persistence completion—and should not be attributed exclusively to changes in the learned graph. The preprocessing is a shortcut-mitigation and diagnostic tool rather than a guarantee of true graph recovery.

3.2 Two practical variants

We instantiate the projection direction q in two simple ways.

MeanProj (spatial mean projection). The simplest choice is the static direction

$$q_{\text{mean}} := \frac{1}{\sqrt{N}}\mathbf{1}, \qquad P_{\text{mean}} := I - q_{\text{mean}} q_{\text{mean}}^\top.$$

Then

$$\widetilde{x}_t = P_{\text{mean}} x_t = x_t - \bar{x}_t \mathbf{1}, \qquad \bar{x}_t := \frac{1}{N}\mathbf{1}^\top x_t. \tag{23}$$

This variant is effective when the global factor is approximately homogeneous across nodes. MeanProj deterministically removes the normalized all-ones direction; it is not a random estimator of the latent loading L.

PCA (adaptive rank-one projection). When the global factor is not homogeneous, we estimate a dominant shared direction from the training histories. Let $\mathcal{X}_{\text{tr}}$ denote the collection of node vectors appearing in the training inputs and define

$$\widehat{\mu}_{\text{tr}} := \frac{1}{|\mathcal{X}_{\text{tr}}|} \sum_{x \in \mathcal{X}_{\text{tr}}} x, \qquad \widehat{\Sigma}_{\text{tr}} := \frac{1}{|\mathcal{X}_{\text{tr}}|} \sum_{x \in \mathcal{X}_{\text{tr}}} (x - \widehat{\mu}_{\text{tr}})(x - \widehat{\mu}_{\text{tr}})^\top.$$

We choose the leading eigenvector

$$q_{\text{PCA}} := \arg\max_{\|u\|_2 = 1} u^\top \widehat{\Sigma}_{\text{tr}} u$$

and define

$$\widetilde{x}_t = (I - q_{\text{PCA}} q_{\text{PCA}}^\top) x_t. \tag{24}$$

This data-adaptive variant targets the leading shared direction and can better approximate a heterogeneous loading than the fixed all-ones direction, but it is not assumed to recover L exactly.

Practical implementation. Both variants are model-agnostic. The theoretical L is a latent quantity in the population model and is not learned by a supervised objective; the practical method instead selects the proxy q above. PCA directions are estimated on the training split only and then fixed for validation, ID, and OOD evaluation. Inputs and targets are both projected, while the network output is trained against the projected target without an additional hard output-projection layer. We apply the same full-series convention to every backbone and compare each original pipeline with its MeanProj and PCA variants.

4 Experiments

We evaluate shortcut contamination, robustness under controlled synthetic shifts, and real time-of-day shifts. Code: Anonymous code repository.

Table 1: Results of the shortcut contamination control benchmark. Lower Test Loss and higher Graph AUROC are better. `grad_graph_ratio` tracks the gradient allocation towards graph parameters.

Model	Condition	Test Loss	Graph AUROC	`grad_graph_ratio`
AGCRN	`no_shortcut`	**0.776**	**0.594**	0.201
	`with_shortcut`	0.825	0.506	**0.858**
DGCRN	`no_shortcut`	1.200	**0.486**	0.599
	`with_shortcut`	**0.986**	0.439	0.205
GTS	`no_shortcut`	0.684	**0.531**	~ 0
	`with_shortcut`	**0.531**	0.513	~ 0
GWNet	`no_shortcut`	**0.049**	0.422	0.188
	`with_shortcut`	0.058	**0.556**	0.265
MTGNN	`no_shortcut`	**0.050**	0.517	~ 0
	`with_shortcut`	0.058	**0.581**	~ 0

4.1 Experimental setup

Synthetic OOD benchmark. We generate multivariate time series with a sparse local interaction graph and a controllable persistent global factor. The local graph is fixed across environments, while the strength of the global factor is varied only at test time. The main setting uses $N = 8$ nodes, lag order $p = 3$, input length 24, horizon 1, training sequence length 2600, and test sequence length 1400. The training environment uses factor strength 1.0, while OOD environments use strengths $\{0.0, 0.5, 1.0, 1.5, 2.0\}$.

Real-data OOD benchmark. For real data, we use two full-scale traffic benchmarks: METR-LA with 207 sensors and PEMS-BAY with 325 sensors (Li et al., 2018). For both datasets, train/validation/ID test splits are constructed from weekday off-peak periods (10:00–16:00), while OOD environments correspond to time-of-day shifts: `shift_am_rush` (weekday 7:00–9:00), `shift_pm_rush` (weekday 17:00–19:00), and `shift_weekend` (weekend daytime). All real-data results are averaged over 3 random seeds. The released traffic pipeline operates directly on the original traffic-speed values, and all real-data MSE values are reported on that scale; for reference, an MSE of 100 corresponds to an RMSE of 10 in the original units.

Compared models and metrics. We evaluate five representative end-to-end graph-learning forecasters: Graph WaveNet, MTGNN, GTS, AGCRN, and DGCRN. For each model family, we compare the original pipeline against the proposed MeanProj and PCA variants. The main metric is OOD MSE; ID MSE and additional diagnostics are reported in the appendix.

4.2 Experiment 1: controlled shortcut contamination (synthetic data)

We first construct synthetic data to explicitly control the presence of a persistent global shortcut, comparing conditions with and without it.

Results are presented in Table 1. For DGCRN and GTS, shortcuts improve test loss but degrade graph AUROC, showing that predictive gains can come at the cost of structural accuracy, consistent with Theorem 1. AGCRN shows a more severe case: both metrics worsen under shortcuts, while `grad_graph_ratio` jumps from 0.20 to 0.86, indicating a larger relative graph-parameter gradient in this condition. GTS and MTGNN show near-zero graph-gradient ratios in both conditions, whereas GWNet has ratios of 0.188 and 0.265. Overall, shortcut effects are architecture-dependent, consistent with our local theoretical regime.

4.3 Experiment 2: OOD strength sweep (synthetic data)

This strength-shift sweep complements, rather than directly tests, Theorem 2, which concerns loading rotations. In Table 4, reported gaps are near zero at $s = 1$, negative at the tested strengths below 1, and positive and increasing at the tested strengths above 1. The shortcut-graph linear baseline has a

Table 2: Projection-direction ablation at $s = 2.0$: OOD Gap MSE (lower is better). Original and PCA-PC1 are the references from this ablation suite, not the separately fitted five-backbone sweep in Table 5. PCA-PC2 removes only the second principal component.

Model	Original	MeanProj	PCA-PC1	PCA-PC2	RandomProj
MTGNN	0.270	0.185	**0.147**	0.299	0.280
GTS	0.322	**0.181**	0.185	0.305	0.245
AGCRN	0.170	0.160	**0.136**	0.173	0.182

Table 3: Compact real-data OOD summary over AM rush, PM rush, and weekend shifts, computed from the reported mean MSEs. Positive Δ denotes the unweighted mean relative reduction versus Original; "Lower" counts lower point estimates and is not a significance test.

		MeanProj		PCA	
Dataset	Model	Lower	Δ (%)	Lower	Δ (%)
METR-LA	GWNet	1/3	-1.4	1/3	-1.0
	MTGNN	3/3	50.8	3/3	50.8
	GTS	3/3	73.8	3/3	73.9
	AGCRN	3/3	17.0	3/3	15.9
	DGCRN	2/3	8.0	2/3	8.3
PEMS-BAY	GWNet	1/3	1.3	1/3	-2.1
	MTGNN	3/3	93.4	3/3	47.1
	GTS	3/3	86.4	1/3	-25.5
	AGCRN	3/3	15.1	0/3	-43.8
	DGCRN	3/3	17.6	1/3	-31.3

slightly smaller gap than the oracle-graph linear baseline at $s \in \{1.5, 2.0\}$, but this does not establish lower absolute OOD MSE or verify the population-risk comparison in Theorem 1.

When comparing the original models against PCA projection (Table 5 in Appendix), we observe substantial reductions in the OOD gap across all architectures, ranging from 17% to 49%. MeanProj achieves comparable improvements at strength 1.5 (e.g., GWNet: 0.064, AGCRN: 0.057, DGCRN: 0.068), showing that both projection strategies can substantially reduce OOD gaps in the synthetic benchmarks. **Projection-direction ablation.** To test whether the gain comes from removing the dominant shared shortcut direction rather than arbitrary denoising, we compare rank-one projections $\tilde{x}_t = (I - qq^\top)x_t$ using different fixed or training-estimated directions. MeanProj uses the normalized all-ones direction, PCA-PC1 and PCA-PC2 use the first and second principal components respectively, and RandomProj uses a fixed random unit direction. Table 2 shows that MeanProj or PCA-PC1 consistently outperforms PCA-PC2 and RandomProj, supporting that the benefit depends on removing the dominant shortcut-aligned direction.

4.4 Experiment 3: Real-world time-of-day shifts

We next test whether the synthetic findings transfer to METR-LA and PEMS-BAY under the time-of-day shifts in Section 4.1, interpreting them as robustness stress tests rather than graph-recovery evaluations because the true functional graph is unknown.

For each shift, we evaluate Original, MeanProj, and PCA as separate fixed pipelines; we never select a projection using the test split. Table 3 gives a compact cross-dataset summary. "Lower" counts shifts for which a variant has a lower mean OOD MSE than Original, and Δ is the unweighted mean relative MSE reduction across the three shifts. Complete mean $\pm$ standard-deviation values, ID errors, OOD gaps, and graph diagnostics are reported in Appendix B.

On METR-LA, both projection variants lower the OOD-MSE point estimate for MTGNN and GTS in all three shifts, with average reductions of approximately 51% and 74%, respectively. On PEMS-BAY, MeanProj lowers the point estimate for both models in all three shifts, whereas PCA is less reliable for GTS. Effects on the other backbones are smaller or direction-dependent, particularly for PCA on

PEMS-BAY. This architecture- and direction-dependence is consistent with the method's role as a conditional mitigation rather than a universal forecasting improvement.

Projection placement ablation. Tables 12 and 13 compare the full pipeline with an evaluation-input projection control trained on unprojected pairs. The control has higher OOD MSE in every reported setting and does not reproduce the gains. Because target coordinates, validation inputs, and output completion also differ, this is not an isolated test of graph learning or projection timing; Appendix B.2 specifies the protocol and run groups.

4.5 Cross-dataset takeaways and limitations

Projection is a diagnostic and mitigation tool, not a graph-identification guarantee. Each evaluated variant uses a fixed spatial-mean direction or a PCA direction estimated once from training data. Continuously varying loadings, high-rank or nonlinear shared structure, and projection selection in a new deployment domain remain outside the evaluated scope.

5 Related work

Graph neural networks are widely used for multivariate time-series forecasting, including fixed-graph, adaptive-graph, latent-graph, and dynamic-graph approaches (Yu et al., 2018; Li et al., 2018; Wu et al., 2019; 2020; Bai et al., 2020; Shang et al., 2021; Li et al., 2021; Satorras et al., 2022; Xu et al., 2023; Cao et al., 2020). While recent models increasingly learn relational structures jointly with the prediction objective, our work analyzes when this objective can make learned graphs predictively useful but structurally misleading.

Related concerns also arise in graph structure learning, including smoothness-based graph learning, sparse relational structure learning, and robust or flexible modeling of inter-series relations (Kalofolias, 2016; Cini et al., 2023; Li et al., 2024; Zhao & Shen, 2024). Rather than proposing another architectural constraint, we study an objective-level failure mode: persistent global components can make shortcut-aligned graphs favorable even when they do not represent invariant local dependencies.

Bayesian graph-learning methods instead represent uncertainty over latent or random graphs (Zhang et al., 2019). The perspectives are complementary: a graph prior can regularize which structures are preferred, whereas our analysis asks how a forecasting objective can favor a shortcut-aligned support in the presence of a persistent shared component. Posterior uncertainty alone does not guarantee invariance across environments, and combining the two perspectives remains an open direction.

Our theoretical viewpoint is connected to shortcut learning, optimization bias, simplicity bias, and gradient starvation in neural networks (Geirhos et al., 2020; Shah et al., 2020; Pezeshki et al., 2021; Scimeca et al., 2022; Hermann et al., 2024). We extend this perspective to graph-learning forecasters, showing that shortcut learning can contaminate the learned relational structure itself, not only the input features used by the predictor.

6 Conclusion

We have studied a failure mode of end-to-end graph learning for multivariate forecasting: forecasting loss can favor train-specific shortcut graphs induced by persistent global components rather than invariant local dependencies. Our one-step population analysis identifies when extra loading-aligned edges lower risk and when loading changes increase test risk; the local optimization analysis states a residual-alignment condition for descent. Synthetic and real traffic experiments show that projection-based decoupling can stabilize shortcut-prone graph learners, while its effect on already stable models is shift-dependent. An adjacency learned solely through forecasting loss should therefore be interpreted as a predictive computation graph, not automatically as a physical, causal, or environment-invariant dependency graph. Likewise, edges that change after projection provide sensitivity evidence but are not thereby certified as spurious or causal.

AI USE STATEMENT

Generative AI tools assisted with software development, reviewing experimental protocols and reported results, and drafting and editing the manuscript. They also assisted with revising mathematical statements and assumptions, re-deriving results, checking counterexamples, and drafting proofs and numerical verification code for Section 2 and Appendix A. The authors conducted the experiments and retain full responsibility for assessing all AI-assisted contributions and for the accuracy, originality, and integrity of the claims, results, text, code, and other artifacts in this submission.

REFERENCES

Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. Adaptive graph convolutional recurrent network for traffic forecasting. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020.

Defu Cao, Yujing Wang, Juanyong Duan, Ce Zhang, Xia Zhu, Congrui Huang, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, and Qi Zhang. Spectral temporal graph neural network for multivariate time-series forecasting. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 17766–17778. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/cdf6581cb7aca4b7e19ef136c6e601a5-Paper.pdf.

Andrea Cini, Daniele Zambon, and Cesare Alippi. Sparse graph learning from spatiotemporal time series. *Journal of Machine Learning Research*, 24(242):1–36, 2023.

Andrea Cini, Ivan Marisca, Daniele Zambon, and Cesare Alippi. Graph deep learning for time series forecasting. *ACM Computing Surveys*, 57(12):321:1–321:34, 2025.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, November 2020. ISSN 2522-5839. doi: 10.1038/s42256-020-00257-z. URL http://dx.doi.org/10.1038/s42256-020-00257-z.

Katherine L. Hermann, Hossein Mobahi, Thomas Fel, and Michael C. Mozer. On the foundations of shortcut learning. In *International Conference on Learning Representations (ICLR)*, 2024.

Vassilis Kalofolias. How to learn a graph from smooth signals. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pp. 920–929, 2016.

Fuxian Li, Jie Feng, Huan Yan, Guangyin Jin, Depeng Jin, and Yong Li. Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution, 2021.

Ting Li, Bing Yu, Jianguo Li, and Zhanxing Zhu. Functional relation field: A model-agnostic framework for multivariate time series forecasting. *Artificial Intelligence*, 334:104158, 2024.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations (ICLR)*, 2018.

Mohammad Pezeshki, Oumar Kaba, Yoshua Bengio, Aaron C Courville, Doina Precup, and Guillaume Lajoie. Gradient starvation: A learning proclivity in neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 1256–1272, 2021.

Victor Garcia Satorras, Syama Sundar Rangapuram, and Tim Januschowski. Multivariate time series forecasting with latent graph inference, 2022. URL https://arxiv.org/abs/2203.03423.

Luca Scimeca, Seong Joon Oh, Sanghyuk Chun, Michael Poli, and Sangdoo Yun. Which shortcut cues will dnns choose? a study from the parameter-space perspective. In *International Conference on Learning Representations (ICLR)*, 2022.

Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020.

Chao Shang, Jie Chen, and Jinbo Bi. Discrete graph structure learning for forecasting multiple time series. In *International Conference on Learning Representations (ICLR)*, 2021.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1907–1913, 2019. doi: 10.24963/ijcai.2019/264.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 753–763, 2020.

Nancy Xu, Chrysoula Kosma, and Michalis Vazirgiannis. Timegnn: Temporal dynamic graph learning for time series forecasting. In *Complex Networks*, 2023.

Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 3634–3640, 2018. doi: 10.24963/ijcai.2018/505.

Yingxue Zhang, Soumyasundar Pal, Mark Coates, and Deniz Üstebay. Bayesian graph convolutional neural networks for semi-supervised classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 5829–5836, 2019. doi: 10.1609/aaai.v33i01.33015829.

Lifan Zhao and Yanyan Shen. Rethinking channel dependence for multivariate time series forecasting: Learning from leading indicators. In *International Conference on Learning Representations (ICLR)*, 2024.

A ADDITIONAL THEORETICAL ANALYSIS

This appendix provides proofs and additional details for the theoretical analysis in Section 2. The purpose of the analysis is not to model every detail of nonlinear graph-learning forecasters, but to isolate a failure mechanism: forecasting loss can favor graph directions aligned with persistent global components rather than invariant local dependencies.

A.1 POPULATION MODEL

We use the latent-component model

$$x_t = z_t + Lg_t, \qquad \|L\|_2 = 1, \tag{25}$$

where z_t is the invariant local component and Lg_t is a rank-one global component. The local component evolves as

$$z_{t+1} = \sum_{\tau=0}^{p-1} B_\tau^\star z_{t-\tau} + \eta_{t+1}, \tag{26}$$

and the global factor follows

$$g_{t+1} = \rho g_t + \nu_{t+1}, \qquad |\rho| < 1. \tag{27}$$

The observed target is therefore

$$x_{t+1} = z_{t+1} + Lg_{t+1}. \tag{28}$$

The true local support mask is

$$(M^\star)_{ij} = \mathbf{1}\left\{ \sum_{\tau=0}^{p-1} |(B_\tau^\star)_{ij}| > 0 \right\}. \tag{29}$$

For the closed-form one-step analysis, we assume

$$z_t \sim \mathcal{N}(0, \sigma_z^2(I - LL^\top)), \qquad g_t \sim \mathcal{N}(0, \sigma_g^2), \qquad z_t \perp\!\!\!\perp g_t. \tag{30}$$

Hence

$$\Sigma_x = \mathrm{Cov}(x_t) = \sigma_z^2(I - LL^\top) + \sigma_g^2 LL^\top. \tag{31}$$

For Theorems 1 and 2, $p = 1$, $\sigma_z^2, \sigma_g^2 > 0$, and the innovations have finite second moments and satisfy

$$\mathbb{E}[\eta_{t+1} \mid z_t, g_t] = 0, \qquad \mathbb{E}[\nu_{t+1} \mid z_t, g_t] = 0. \tag{32}$$

Thus $\epsilon_{t+1} := \eta_{t+1} + L\nu_{t+1}$ has conditional mean zero given x_t. The two innovations need not be independent of each other for the risk comparisons: their combined squared-error contribution is the same for both predictors on a fixed environment. The Gaussian assumption concerns the current history distribution; it is not a claim that an arbitrary choice of B generates that stationary covariance. The one-step risk results hold for history–target populations satisfying these conditions. A stationary local process confined to $L^\perp$ additionally requires compatible dynamics and noise. A sufficient construction is $BL^\perp \subseteq L^\perp$, $\|B|_{L^\perp}\|_{\mathrm{op}} < 1$, and independent Gaussian innovations with covariance $\sigma_z^2(P_L - BP_LB^\top)$, initialized in its stationary law, together with a stationary independent factor whose innovation variance is $(1-\rho^2)\sigma_g^2$. The examples below satisfy these conditions. The optimization identities instead assume a zero-mean covariance-stationary factor and the specified cross-moment conditions at every included lag.

A.2 PROOF OF PROPOSITION 1

Since $z_t \in L^\perp$, we have $P_L z_t = z_t$. Since $P_L L = 0$, we obtain

$$P_L x_t = P_L z_t + P_L Lg_t = z_t.$$

The statement about the local dynamics and support follows immediately from the assumed dynamics of z_t.

A.3 Proof of Theorem 1

We work with $p = 1$ and $B = B_0^\star$. Throughout, $R_i(M) = \{i\} \cup S_i(M)$ includes the diagonal prediction pathway. All matrices with row i supported on $R_i(M)$ can be written as $D + \Theta \odot M$, so this access convention is exactly the risk class in Eq. (6), not a modified predictor.

The latent local term and the predictive factor coefficient. Let $P_L = I - LL^\top$. Under Eq. (30), $L^\top z_t = 0$ almost surely, and hence

$$g_t = L^\top x_t, \qquad z_t = P_L x_t. \tag{33}$$

Using the original latent dynamics gives

$$x_{t+1} = Bx_t + bg_t + \epsilon_{t+1}, \qquad b = \rho L - BL, \quad \epsilon_{t+1} = \eta_{t+1} + L\nu_{t+1}. \tag{34}$$

For any mask M containing $M^\star$ entrywise, the scalar $(Bx_t)_i$ is already measurable from $x_{R_i(M),t}$, since row i of B is supported on $S_i(M^\star)$. This step does not assume that $(Bz_t)_i$ is measurable from the same observed coordinates.

Conditional factor variance with self-information. For any nonempty coordinate set R, let $\bar{R}$ denote its complement. Its observation covariance is positive definite:

$$\Sigma_R := \mathrm{Cov}(x_{R,t}) = \sigma_z^2 I_R + (\sigma_g^2 - \sigma_z^2) L_R L_R^\top. \tag{35}$$

Joint Gaussianity and $\mathrm{Cov}(x_{R,t}, g_t) = \sigma_g^2 L_R$ give

$$v(R) := \mathrm{Var}(g_t \mid x_{R,t}) = \sigma_g^2 - \sigma_g^4 L_R^\top \Sigma_R^{-1} L_R. \tag{36}$$

The conditional mean of g_t is linear in $x_{R,t}$. By the Sherman–Morrison identity,

$$L_R^\top \Sigma_R^{-1} L_R = \frac{\|L_R\|_2^2}{\sigma_z^2 + (\sigma_g^2 - \sigma_z^2)\|L_R\|_2^2}. \tag{37}$$

Substituting $1 - \|L_R\|_2^2 = \|L_{\bar{R}}\|_2^2$ yields

$$v(R) = \frac{\sigma_z^2 \sigma_g^2 \|L_{\bar{R}}\|_2^2}{\sigma_z^2 \|L_{\bar{R}}\|_2^2 + \sigma_g^2 \|L_R\|_2^2}. \tag{38}$$

The denominator is strictly positive because both scalar variances are positive and the two loading energies sum to one.

Risk comparison. The conditional mean for node i is

$$\mathbb{E}[x_{i,t+1} \mid x_{R_i(M),t}] = (Bx_t)_i + b_i \mathbb{E}[g_t \mid x_{R_i(M),t}]. \tag{39}$$

It belongs to the allowed linear class. Since $\mathbb{E}[\epsilon_{t+1} \mid x_t] = 0$, ϵ_{t+1} is orthogonal to all square-integrable functions of x_t, including the remaining factor-estimation error. Therefore, with $\mathcal{J}_{\mathrm{irr}} := \mathbb{E}\|\epsilon_{t+1}\|_2^2$,

$$\mathcal{J}(M) = \mathcal{J}_{\mathrm{irr}} + \sum_i b_i^2 v(R_i(M)) \qquad (M \supseteq M^\star). \tag{40}$$

If $L_i \neq 0$, row i of M_L contains every j with $L_j \neq 0$, so $g_t = L^\top x_t$ is available and $v(R_i(M_L)) = 0$. If $L_i = 0$, the augmentation adds no entries in row i, so $R_i(M_L) = R_i(M^\star)$ and that row contributes zero to the difference, even if $b_i \neq 0$. Consequently,

$$\mathcal{J}(M^\star) - \mathcal{J}(M_L) = \sum_i b_i^2 \{v(R_i(M^\star)) - v(R_i(M_L))\} \tag{41}$$

$$= \sum_{i: L_i \neq 0} b_i^2 v(R_i(M^\star)). \tag{42}$$

Eq. (38) proves Eq. (10). Every summand is nonnegative; it is positive exactly when $b_i \neq 0$ and the excluded loading energy is positive. This proves the stated strict-positivity criterion. □

Persistence-driven special case and a nontrivial stationary construction. When $BL = 0$, $b = \rho L$. The special-case coefficient is then $\rho^2 L_i^2$, but the conditional variance still uses $R_i(M)$, including self-information. For $N = 2$, $B = 0$, $L = (1, 1)^\top/\sqrt{2}$, and $\sigma_z^2 = \sigma_g^2 = 1$, the gap is $\rho^2/2$, which is $1/8$ at $\rho = 1/2$. The strict-gain setting is not limited to a zero local graph. For $N = 4$, let $L = \mathbf{1}/2$ and

$$B = a \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix}, \qquad 0 < |a| < 1/2. \tag{43}$$

Here $BL = 0$, B is symmetric and preserves $L^\perp$, and each local neighborhood contains only its two-node block. With the stationary innovations specified in Appendix A.1, all assumptions hold. Each effective neighborhood excludes nonzero loading mass, so the gap is positive for $\rho \neq 0$. These are mathematical constructions, not additional forecasting experiments.

Scope. For general B, $b = \rho L - BL$ reflects the contrast between factor persistence and the local dynamics; the risk benefit is not attributable to ρ alone. In particular, $B = \rho I$ gives $b = 0$ and no benefit from adding graph edges. This proof is restricted to a one-step Gaussian history. Stacking a multi-lag history generally changes its covariance and conditional variance, so no identical multi-lag closed form is asserted.

A.4 Optimization-level shortcut alignment

We justify Eq. (12). Let U^g be a shortcut update direction in the graph matrix and define

$$\phi^g = \sum_{\tau=0}^{p-1} \Gamma_\tau^{(0)} U^g x_{t-\tau}. \tag{44}$$

Assume

$$\mathbb{E}[g_{t+1} z_{t-\tau}] = 0 \qquad \text{for all } \tau = 0, \dots, p-1. \tag{45}$$

Then

$$\mathbb{E}\left[\langle L g_{t+1},\ \phi^g \rangle\right] = \sum_{\tau=0}^{p-1} \mathbb{E}\left[g_{t+1} L^\top \Gamma_\tau^{(0)} U^g x_{t-\tau} \right] \tag{46}$$

$$= \sum_{\tau=0}^{p-1} \mathbb{E}\left[g_{t+1} L^\top \Gamma_\tau^{(0)} U^g (z_{t-\tau} + L g_{t-\tau}) \right] \tag{47}$$

$$= \sum_{\tau=0}^{p-1} L^\top \Gamma_\tau^{(0)} U^g L\, \mathbb{E}[g_{t+1} g_{t-\tau}]. \tag{48}$$

Since g_t is a zero-mean covariance-stationary AR(1) process with innovations uncorrelated with its past,

$$\mathbb{E}[g_{t+1} g_{t-\tau}] = \rho^{\tau+1} \operatorname{Var}(g_t). \tag{49}$$

Therefore,

$$\mathbb{E}\left[\langle L g_{t+1},\ \phi^g \rangle\right] = \operatorname{Var}(g_t) \sum_{\tau=0}^{p-1} \rho^{\tau+1} L^\top \Gamma_\tau^{(0)} U^g L. \tag{50}$$

A.5 Residual alignment and a conditional local descent mechanism

The identity in Appendix A.4 concerns the global target component. To connect it to optimization, fix a square-integrable reference forecast $f_0(X)$ and direction $\phi^g(X)$, and consider the local output surrogate $f_\alpha = f_0 + \alpha \phi^g$. Write

$$a_g := \mathbb{E}\langle x_{t+1} - f_0, \phi^g \rangle, \qquad v_g := \mathbb{E}\|\phi^g\|_2^2. \tag{51}$$

Direct expansion gives the exact surrogate identity

$$\mathbb{E}\|x_{t+1} - f_\alpha\|_2^2 - \mathbb{E}\|x_{t+1} - f_0\|_2^2 = -2\alpha a_g + \alpha^2 v_g. \tag{52}$$

This proves Eq. (13). If $a_g > 0$, then $v_g > 0$ by Cauchy–Schwarz, and every $0 < \alpha < 2a_g/v_g$ strictly decreases the surrogate loss. If $v_g = 0$, the direction vanishes almost surely and $a_g = 0$.

The actual residual alignment decomposes as

$$a_g = \mathbb{E}\langle Lg_{t+1}, \phi^g \rangle + \mathbb{E}\langle z_{t+1} - f_0, \phi^g \rangle. \tag{53}$$

Thus a positive global alignment is sufficient for descent only with an additional restriction on the second term, for example that it is nonnegative (or zero). A previously learned forecast can already explain the global component, making a_g zero or negative even when the target alignment is positive.

This gives a conditional local optimization mechanism: a shortcut-aligned perturbation can reduce loss when an aligned component remains in the residual. It does not establish that its decrease exceeds that of a normalized local-structure perturbation, that gradient descent selects it, or that other directions are starved over training. For a nonlinear model the output surrogate is a first-order approximation; a descent statement for the actual model additionally requires a feasible perturbation, differentiability, and sufficient control to differentiate the expected loss. The present identity is not a convergence or universal gradient-starvation theorem.

A.6 Train-fitted shortcut fragility under environment shift

We give a complete derivation for the original latent-z model, with $p = 1$ and shared $B = B_0^\star$. Each environment has unit loading L_e, $Q_e = L_e L_e^\top$, $P_e = I - Q_e$, current history covariance $\Sigma_e = \sigma_z^2 P_e + \sigma_g^2 Q_e$, and conditionally mean-zero innovations as specified in the main theorem. Both scalar variances are strictly positive. Shared local dynamics means shared B, not a shared full local covariance: the assumed local history covariance is $\sigma_z^2 P_e$.

The fitted predictor, oracle, and graph support. Since $z_t^e = P_e x_t^e$ and $g_t^e = L_e^\top x_t^e$,

$$x_{t+1}^e = C_e x_t^e + \epsilon_{t+1}^e, \qquad C_e = BP_e + \rho Q_e = B + (\rho L_e - BL_e)L_e^\top, \tag{54}$$

where $\mathbb{E}[\epsilon_{t+1}^e \mid x_t^e] = 0$. Thus $C_e x_t^e$ is the conditional-mean oracle among predictors using x_t^e, and is the unique optimal linear map since Σ_e is positive definite. We define the population-fitted train predictor as $\widehat{f}^{\mathrm{tr}}_{\mathrm{short}}(X) = C_{\mathrm{tr}} x_t$ and the test oracle as $f^{\mathrm{te}}_{\mathrm{oracle}}(X) = C_{\mathrm{te}} x_t$. The oracle here is not the separately named oracle-graph experimental baseline. For any graph mask M, C_{tr} belongs to its prediction class precisely when $(C_{\mathrm{tr}})_{ij} = 0$ for every $j \notin R_i(M)$. For M_L^{tr} defined by Eq. (8) with L_{tr}, this is guaranteed if L_{tr} has no zero entries, or if $BL_{\mathrm{tr}} = 0$; it need not hold for arbitrary sparse loadings and B. The schematic weighted graph

$$A^{\mathrm{tr}}_{\mathrm{short}} = A^\star + \lambda L_{\mathrm{tr}} L_{\mathrm{tr}}^\top, \qquad \lambda \geq 0, \tag{55}$$

only illustrates a loading-aligned augmentation. It is not the fitted coefficient matrix C_{tr}; if its support is used, mask representability must be checked separately, including possible cancellation of weighted entries.

Proposition 2 (Population rotation-gap identity)**.** *Let $H = Q_{\mathrm{tr}} - Q_{\mathrm{te}}$ and $A = \rho I - B$. For the explicitly defined predictors,*

$$\mathcal{R}_{\mathrm{te}}(\widehat{f}^{\mathrm{tr}}_{\mathrm{short}}) - \mathcal{R}^{\mathrm{oracle}}_{\mathrm{te}} = \mathrm{tr}(AH\Sigma_{\mathrm{te}} H A^\top) = \|AH\Sigma_{\mathrm{te}}^{1/2}\|_F^2. \tag{56}$$

If $BL_{\mathrm{tr}} = BL_{\mathrm{te}} = 0$, this becomes

$$\mathcal{R}_{\mathrm{te}}(\widehat{f}^{\mathrm{tr}}_{\mathrm{short}}) - \mathcal{R}^{\mathrm{oracle}}_{\mathrm{te}} = \rho^2(\sigma_g^2 + \sigma_z^2)\sin^2\theta, \tag{57}$$

where $\cos\theta = |L_{\mathrm{tr}}^\top L_{\mathrm{te}}|$.

Proof. On the test population, the oracle residual is $\epsilon_{t+1}^{\mathrm{te}}$. Its conditional mean given x_t^{te} is zero, so the cross term with $(C_{\mathrm{tr}} - C_{\mathrm{te}})x_t^{\mathrm{te}}$ vanishes. Therefore

$$\mathcal{R}_{\mathrm{te}}(\widehat{f}^{\mathrm{tr}}_{\mathrm{short}}) - \mathcal{R}^{\mathrm{oracle}}_{\mathrm{te}} = \mathbb{E}_{\mathrm{te}}\|(C_{\mathrm{tr}} - C_{\mathrm{te}})x_t^{\mathrm{te}}\|_2^2 \tag{58}$$

$$= \mathrm{tr}((C_{\mathrm{tr}} - C_{\mathrm{te}})\Sigma_{\mathrm{te}}(C_{\mathrm{tr}} - C_{\mathrm{te}})^\top). \tag{59}$$

The common local coefficient does not in general cancel from the difference: since $C_e = B + (\rho I - B)Q_e$,

$$C_{\mathrm{tr}} - C_{\mathrm{te}} = (\rho I - B)H = AH. \tag{60}$$

This proves Eq. (56) and Eq. (16). Since Σ_{te} is positive definite, the gap is positive if and only if $AH \neq 0$.

Under $BL_{\mathrm{tr}} = BL_{\mathrm{te}} = 0$, $BH = 0$ and hence $AH = \rho H$. Put $c = L_{\mathrm{tr}}^\top L_{\mathrm{te}}$. Rank-one projector algebra gives

$$\mathrm{tr}(H^2) = 2(1 - c^2), \qquad L_{\mathrm{te}}^\top H^2 L_{\mathrm{te}} = 1 - c^2. \tag{61}$$

For the second identity, $HL_{\mathrm{te}} = cL_{\mathrm{tr}} - L_{\mathrm{te}}$ has squared norm $1 - c^2$. Consequently,

$$\mathrm{tr}(H\Sigma_{\mathrm{te}}H) = \sigma_z^2 \,\mathrm{tr}(H^2 P_{\mathrm{te}}) + \sigma_g^2 \,\mathrm{tr}(H^2 Q_{\mathrm{te}}) \tag{62}$$

$$= (\sigma_z^2 + \sigma_g^2)(1 - c^2) = (\sigma_z^2 + \sigma_g^2)\sin^2\theta. \tag{63}$$

Multiplication by ρ^2 proves the angular special case and completes the proof of Theorem 2. □

Nonzero-gap and cancellation cases. The angular formula is strictly positive only when $\rho \neq 0$ and $\theta > 0$. A valid stationary construction is $B = 0$, independent $z_t^e \sim \mathcal{N}(0, \sigma_z^2 P_e)$ over time, and a stationary independent AR(1) factor with innovation variance $(1 - \rho^2)\sigma_g^2$. The nonzero local-interaction construction in Eq. (43) also exhibits both effects simultaneously. Take $L_{\mathrm{tr}} = (1, 1, 1, 1)^\top/2$ and $L_{\mathrm{te}} = (1, 1, -1, -1)^\top/2$, and unit scalar variances. Both loadings are annihilated by that same B. Theorem 1 gives a training-population support advantage $\rho^2/2$, while Theorem 2 gives test excess risk $2\rho^2$ for the frozen train fit, both strictly positive for $\rho \neq 0$. Compatible stationary innovations exist in both environments as described above. This construction preserves the distinction between a sparse invariant local graph and a globally predictive, environment-specific component. More generally, if $B = aI$ with $|a| < 1$, then $AH = (\rho - a)H$, giving

$$\mathcal{R}_{\mathrm{te}}(\widehat{f}_{\mathrm{short}}^{\mathrm{tr}}) - \mathcal{R}_{\mathrm{te}}^{\mathrm{oracle}} = (\rho - a)^2(\sigma_g^2 + \sigma_z^2)\sin^2\theta. \tag{64}$$

This case is realized by stationary local innovations with covariance $(1 - a^2)\sigma_z^2 P_e$. At $a = \rho$, both population predictors are ρx_t and the gap is zero even if the loading directions differ. Thus a rotation alone is insufficient without a predictive contrast between the global and local components. These are population identities, not empirical claims about a trained nonlinear model. The strength-only experiments remain complementary stress tests, not a validation of the angular formula. Likewise, a fixed training projector P_{tr} generally leaves $P_{\mathrm{tr}}L_{\mathrm{te}} \neq 0$ after a rotation; the theorem does not claim that train-fitted projection eliminates this gap.

A.7 Projection suppresses shortcut alignment

For a unit vector q, define

$$P_q = I - qq^\top, \qquad \widetilde{x}_t = P_q x_t. \tag{65}$$

Under the latent decomposition,

$$\widetilde{x}_t = P_q z_t + P_q L g_t. \tag{66}$$

Thus, the remaining shortcut source is controlled by $P_q L$.

Define the projected shortcut output direction

$$\phi_q^g = \sum_{\tau=0}^{p-1} \Gamma_\tau^{(0)} U^g P_q x_{t-\tau}. \tag{67}$$

Using the same calculation as in Appendix A.4,

$$\mathbb{E}\left[\left\langle Lg_{t+1},\ \phi_q^g \right\rangle\right] = \mathrm{Var}(g_t) \sum_{\tau=0}^{p-1} \rho^{\tau+1} L^\top \Gamma_\tau^{(0)} U^g P_q L. \tag{68}$$

Therefore,

$$\left|\mathbb{E}\left[\left\langle Lg_{t+1},\ \phi_q^g \right\rangle\right]\right| \leq \mathrm{Var}(g_t) \sum_{\tau=0}^{p-1} |\rho|^{\tau+1} \left| L^\top \Gamma_\tau^{(0)} U^g P_q L \right| \tag{69}$$

$$\leq \mathrm{Var}(g_t) \sum_{\tau=0}^{p-1} |\rho|^{\tau+1} \|\Gamma_\tau^{(0)} U^g\|_{\mathrm{op}} \|P_q L\|_2. \tag{70}$$

Table 4: Baseline OOD Gap MSE across all test-time shift strengths (training strength = 1.0). Bold indicates the better value between shortcut and oracle graph baselines.

	Test-time shift strength s					
Model	0.0	0.25	0.5	1.0	1.5	2.0
`shortcut_graph`	−0.036	−0.034	−0.026	0.004	**0.053**	**0.120**
`oracle_graph`	**−0.038**	−0.034	**−0.027**	0.004	0.055	0.128
`graph_learned`	−0.048	−0.044	−0.034	0.008	0.078	0.176
`mlp`	−0.051	−0.048	−0.034	0.000	0.074	0.183
`temporal_linear`	−0.050	−0.047	−0.036	0.009	0.083	0.188

If $q = L$, then $P_q L = 0$ and the alignment with the dominant global shortcut vanishes. If q only approximates L, the residual shortcut alignment is controlled by $\|P_q L\|_2$.

B Additional experimental tables

For page efficiency, the detailed synthetic, METR-LA, and PEMS-BAY result tables are collected here.

Table 5: OOD Gap MSE improvement with PCA at $s = 2.0$, from the five-backbone strength-sweep runs. The projection-direction and rank ablations in Tables 2 and 15 use a separate set of fitted runs and their own Original/PCA references.

Model	Original Gap	PCA Gap	Improvement
GWNet	0.239	**0.136**	−43%
MTGNN	0.263	**0.147**	−44%
GTS	0.260	**0.149**	−43%
AGCRN	0.165	**0.136**	−17%
DGCRN	0.272	**0.138**	−49%

B.1 PEMS-BAY detailed tables

B.2 Real-data projection placement ablation

Tables 12 and 13 compare the full train+eval pipeline with an evaluation-input projection control. Train+Eval uses projected training inputs and targets with causal output completion. In the Eval-only control, training pairs are unprojected, validation and test inputs are projected, validation targets remain unprojected, and no output completion is applied. Checkpoints are selected using the corresponding validation loss. The control therefore differs in training coordinates, validation inputs, and output handling, not solely in the timing of an otherwise identical transformation.

The METR-LA placement results come from dedicated fits, separate from the detailed benchmark runs. For its three per-seed OOD MSEs m_0, m_1, m_2, Table 13 reports $\bar{m} = (m_0 + m_1 + m_2)/3$ and $s_m = \sqrt{\sum_{k=0}^{2}(m_k - \bar{m})^2/2}$ (`ddof=1`). Each table must be interpreted within its own run group; differences between groups are not paired treatment effects.

On both datasets, the evaluated input-projection control has higher OOD MSE than Original and Train+Eval in every reported setting. It does not reproduce the full pipeline's gains. Because training coordinates, validation inputs, and output handling differ, the comparison does not separately identify the contributions of the learned graph, projected-target training, and persistence completion.

C Additional experimental details and results

This appendix provides additional experimental context that complements the main paper. In particular, we report the contamination-control benchmark that motivated the shortcut-contamination

Table 6: METR-LA (207 sensors), `shift_am_rush`. Average ± std over 3 seeds.

Model	Variant	ID MSE	OOD MSE	OOD Gap	AUROC
GWNet	Original	37.9±1.9	62.4±13.2	24.5±12.1	0.495±0.004
	MeanProj	36.8±1.7	64.9±8.9	28.2±8.2	0.495±0.004
	PCA	36.8±1.6	64.6±9.1	27.8±8.4	0.492±0.007
MTGNN	Original	89.4±9.7	100.6±7.6	11.2±3.2	0.500±0.003
	MeanProj	38.0±2.0	69.8±7.8	31.8±6.9	0.507±0.007
	PCA	37.8±1.3	70.3±7.3	32.5±7.0	0.505±0.005
GTS	Original	241.3±60.0	280.5±26.4	39.2±55.3	0.497±0.033
	MeanProj	52.6±1.3	92.5±7.9	39.8±7.1	0.479±0.017
	PCA	54.1±1.1	91.6±6.4	37.5±5.9	0.535±0.016
AGCRN	Original	42.7±3.4	72.7±14.1	30.0±11.4	0.505±0.012
	MeanProj	37.3±1.6	69.3±8.8	32.1±7.9	0.537±0.013
	PCA	37.6±1.8	70.9±9.1	33.3±8.1	0.531±0.009
DGCRN	Original	39.0±2.4	63.8±12.3	24.8±11.1	0.503±0.007
	MeanProj	36.8±1.4	65.2±7.9	28.4±7.5	0.506±0.007
	PCA	37.0±1.4	65.6±7.9	28.6±7.5	0.504±0.006

Table 7: METR-LA (207 sensors), `shift_pm_rush`. Average ± std over 3 seeds.

Model	Variant	ID MSE	OOD MSE	OOD Gap	AUROC
GWNet	Original	37.9±1.9	42.9±6.4	5.0±7.0	0.495±0.004
	MeanProj	36.8±1.7	42.5±5.8	5.7±6.5	0.495±0.004
	PCA	36.8±1.6	42.1±5.9	5.4±6.5	0.492±0.007
MTGNN	Original	89.4±9.7	90.1±14.5	0.6±13.0	0.500±0.003
	MeanProj	38.0±2.0	43.9±6.3	5.9±6.9	0.507±0.007
	PCA	37.8±1.3	43.8±6.8	6.0±7.1	0.505±0.005
GTS	Original	241.3±60.0	271.5±33.6	30.3±42.6	0.497±0.033
	MeanProj	52.6±1.3	79.9±4.2	27.2±5.1	0.479±0.017
	PCA	54.1±1.1	78.2±6.6	24.1±6.9	0.535±0.016
AGCRN	Original	42.7±3.4	51.2±3.6	8.5±6.1	0.505±0.012
	MeanProj	37.3±1.6	43.7±5.6	6.4±6.4	0.537±0.013
	PCA	37.6±1.8	43.7±5.7	6.1±6.7	0.531±0.009
DGCRN	Original	39.0±2.4	48.2±6.4	9.2±6.5	0.503±0.007
	MeanProj	36.8±1.4	43.0±6.2	6.1±6.6	0.506±0.007
	PCA	37.0±1.4	42.5±6.8	5.5±7.1	0.504±0.006

viewpoint, and we summarize the concrete data splits used in the synthetic and real-data OOD evaluations. The synthetic benchmarks are described in language consistent with the latent-component population model used in Appendix A: a sparse latent local process is generated from an invariant interaction graph, and an additive persistent global component is superimposed on the observed multivariate series.

C.1 Experimental goals

The empirical study is designed to test three questions. First, we ask whether adding a persistent global factor changes the learned graph and the optimization dynamics of end-to-end graph learners. Second, we test whether the proposed projection-based decoupling suppresses train-specific shortcut directions and improves robustness under distribution shift. Third, we examine whether, from the standpoint of prediction loss alone, a shortcut-compatible graph can become preferable to the true local graph.

Table 8: METR-LA (207 sensors), `shift_weekend`. Average ± std over 3 seeds.

Model	Variant	ID MSE	OOD MSE	OOD Gap	AUROC
GWNet	Original	37.9±1.9	29.2±4.7	−8.7±2.8	0.495±0.004
	MeanProj	36.8±1.7	29.5±7.3	−7.3±5.6	0.495±0.004
	PCA	36.8±1.6	29.6±7.2	−7.2±5.6	0.492±0.007
MTGNN	Original	89.4±9.7	106.0±8.2	16.6±3.1	0.500±0.003
	MeanProj	38.0±2.0	31.2±7.9	−6.9±5.9	0.507±0.007
	PCA	37.8±1.3	30.9±7.3	−7.0±6.1	0.505±0.005
GTS	Original	241.3±60.0	287.2±46.9	46.0±13.7	0.497±0.033
	MeanProj	52.6±1.3	46.3±8.9	−6.4±7.9	0.479±0.017
	PCA	54.1±1.1	48.1±9.2	−6.0±8.2	0.535±0.016
AGCRN	Original	42.7±3.4	44.4±4.6	1.8±1.4	0.505±0.012
	MeanProj	37.3±1.6	30.3±7.1	−7.0±5.5	0.537±0.013
	PCA	37.6±1.8	30.8±7.6	−6.8±5.8	0.531±0.009
DGCRN	Original	39.0±2.4	35.4±5.1	−3.7±2.7	0.503±0.007
	MeanProj	36.8±1.4	29.9±6.9	−6.9±5.6	0.506±0.007
	PCA	37.0±1.4	29.8±7.2	−7.2±5.8	0.504±0.006

Table 9: PEMS-BAY (325 sensors), `shift_am_rush`. Mean ± std over three seeds.

Model	Variant	ID MSE	OOD MSE	OOD Gap	AUROC
GWNet	Original	3.0±0.3	6.0±0.5	3.1±0.7	0.503±0.003
	MeanProj	2.7±0.1	6.0±0.3	3.4±0.3	0.501±0.003
	PCA	3.2±0.1	5.7±0.2	2.5±0.2	0.502±0.003
MTGNN	Original	154.4±12.7	103.6±6.3	−50.8±6.8	0.500±0.004
	MeanProj	2.8±0.1	12.9±2.3	10.2±2.3	0.500±0.005
	PCA	98.3±8.3	30.5±1.9	−67.8±6.5	0.500±0.004
GTS	Original	39.5±9.5	195.3±57.4	155.8±52.1	0.464±0.057
	MeanProj	5.1±0.1	28.2±9.5	23.0±9.4	0.501±0.046
	PCA	58.7±23.8	199.6±101.3	140.9±80.5	0.476±0.025
AGCRN	Original	3.1±0.1	14.0±2.1	10.9±2.2	0.519±0.003
	MeanProj	2.7±0.1	10.3±2.2	7.6±2.1	0.545±0.007
	PCA	5.1±0.1	15.5±4.4	10.4±4.5	0.588±0.009
DGCRN	Original	3.2±0.1	7.4±0.7	4.2±0.6	0.500±0.007
	MeanProj	2.6±0.1	5.6±0.3	3.0±0.2	0.500±0.006
	PCA	4.5±0.1	8.5±2.0	3.9±1.9	0.499±0.007

C.2 CONTAMINATION-CONTROL BENCHMARK

We first construct synthetic data with two ingredients: a sparse latent local interaction process and an optional additive persistent common factor shared across nodes. For each architecture, we compare two conditions:

$$\texttt{no_shortcut} : \alpha = 0, \qquad \texttt{with_shortcut} : \alpha = 1,$$

where α controls the strength of the persistent global component added to the observed series. We record the final test loss, graph recovery AUROC, and gradient-based optimization diagnostics such as the relative graph-gradient strength.

Table 14 shows that the effect of the persistent global factor is architecture dependent. The clearest case is DGCRN: when the shortcut is introduced, the test loss improves from 1.2005 to 0.9862, while the graph AUROC drops from 0.4859 to 0.4391. In the reported diagnostics, the graph-gradient ratio also decreases markedly in this setting. This is a within-architecture observation, not a direct measurement of the residual directional alignment in Appendix A.5 or a proof of gradient starvation.

GTS exhibits a similar qualitative pattern: the shortcut improves prediction error while slightly degrading graph recovery. By contrast, Graph WaveNet and MTGNN do not necessarily show worse

Table 10: PEMS-BAY (325 sensors), `shift_pm_rush`. Mean ± std over three seeds.

Model	Variant	ID MSE	OOD MSE	OOD Gap	AUROC
GWNet	Original	3.0±0.3	5.2±0.4	2.3±0.1	0.503±0.003
	MeanProj	2.7±0.1	5.5±0.3	2.8±0.4	0.501±0.003
	PCA	3.2±0.1	5.3±0.3	2.1±0.3	0.502±0.003
MTGNN	Original	154.4±12.7	122.9±10.1	−31.6±4.3	0.500±0.004
	MeanProj	2.8±0.1	7.4±0.6	4.6±0.6	0.500±0.005
	PCA	98.3±8.3	84.8±6.1	−13.5±2.2	0.500±0.004
GTS	Original	39.5±9.5	199.3±57.6	159.7±50.0	0.464±0.057
	MeanProj	5.1±0.1	23.2±2.0	18.1±1.9	0.501±0.046
	PCA	58.7±23.8	115.7±69.3	57.0±46.6	0.476±0.025
AGCRN	Original	3.1±0.1	6.3±0.1	3.2±0.1	0.519±0.003
	MeanProj	2.7±0.1	5.9±0.3	3.2±0.3	0.545±0.007
	PCA	5.1±0.1	6.3±0.1	1.2±0.2	0.588±0.009
DGCRN	Original	3.2±0.1	6.0±0.1	2.8±0.0	0.500±0.007
	MeanProj	2.6±0.1	5.5±0.1	2.9±0.2	0.500±0.006
	PCA	4.5±0.1	5.7±0.2	1.1±0.3	0.499±0.007

Table 11: PEMS-BAY (325 sensors), `shift_weekend`. Mean ± std over three seeds.

Model	Variant	ID MSE	OOD MSE	OOD Gap	AUROC
GWNet	Original	3.0±0.3	2.1±0.5	−0.9±0.7	0.503±0.003
	MeanProj	2.7±0.1	1.9±0.5	−0.7±0.6	0.501±0.003
	PCA	3.2±0.1	2.3±0.7	−0.9±0.8	0.502±0.003
MTGNN	Original	154.4±12.7	163.7±14.1	9.2±2.7	0.500±0.004
	MeanProj	2.8±0.1	2.0±0.6	−0.8±0.6	0.500±0.005
	PCA	98.3±8.3	98.7±9.9	0.4±1.8	0.500±0.004
GTS	Original	39.5±9.5	26.7±9.1	−12.8±0.5	0.464±0.057
	MeanProj	5.1±0.1	3.9±1.0	−1.2±1.0	0.501±0.046
	PCA	58.7±23.8	57.7±25.9	−0.9±2.3	0.476±0.025
AGCRN	Original	3.1±0.1	2.4±0.6	−0.7±0.7	0.519±0.003
	MeanProj	2.7±0.1	2.1±0.4	−0.6±0.5	0.545±0.007
	PCA	5.1±0.1	5.3±0.7	0.2±0.7	0.588±0.009
DGCRN	Original	3.2±0.1	2.5±0.8	−0.6±0.7	0.500±0.007
	MeanProj	2.6±0.1	2.0±0.5	−0.6±0.6	0.500±0.006
	PCA	4.5±0.1	4.6±0.8	0.1±0.8	0.499±0.007

final AUROC after full nonlinear training. This does not contradict the theory. The optimization result is local and conditional: a shortcut direction decreases the surrogate loss when its residual alignment is positive, but the identities do not determine the final nonlinear endpoint. In some cases the graph pathway becomes less useful; in others it may reorganize to represent the shortcut itself.

The main message of this control benchmark is therefore not that all architectures converge to an identical failure mode. Rather, it shows that adding a persistent global factor changes the reported predictive, graph-recovery, and gradient diagnostics in an architecture-dependent manner; these observations motivate, rather than fully verify, the proposed mechanism.

C.3 Synthetic OOD benchmark details

For the synthetic OOD evaluation, the local graph is fixed across environments while only the strength of the persistent global factor is changed at test time. The main benchmark uses $N = 8$ nodes, lag order $p = 3$, input length 24, prediction horizon 1, training sequence length 2600, and test sequence length 1400. The training environment always uses factor strength 1.0, whereas the OOD environments use

$$\{0.0,\ 0.5,\ 1.0,\ 1.5,\ 2.0\}.$$

Table 12: Projection placement ablation on PEMS-BAY: OOD MSE (lower is better), mean ± std over three seeds. Train+Eval uses the full MeanProj pipeline; Eval-only is the validation/test-input projection control defined above.

Model	Shift	Original	MeanProj (Train+Eval)	MeanProj (Eval-only)
MTGNN	`pm_rush`	122.9±10.1	7.4±0.6	1384.7±67.3
MTGNN	`weekend`	163.7±14.1	2.0±0.6	3252.8±139.3
GTS	`pm_rush`	199.3±57.6	23.2±2.0	2973.1±214.3
GTS	`weekend`	26.7±9.1	3.9±1.0	3960.9±196.9
AGCRN	`pm_rush`	6.3±0.1	5.9±0.3	296.3±11.7
AGCRN	`weekend`	2.4±0.6	2.1±0.4	64.8±5.4

Table 13: Projection placement ablation on METR-LA: mean OOD MSE ± sample standard deviation (`ddof=1`) over seeds $0, 1, 2$. These dedicated placement-ablation fits are separate from the detailed benchmark runs in Tables 6–8; Original and Train+Eval are not reused from those tables. The evaluation-input control is defined above.

Model	Shift	Original	MeanProj (Train+Eval)	MeanProj (Eval-only)
MTGNN	`pm_rush`	79.8±6.8	43.9±8.0	1239.2±77.8
MTGNN	`weekend`	99.5±24.3	31.3±8.8	1923.5±234.2
GTS	`pm_rush`	265.1±38.0	80.2±5.2	1862.5±550.5
GTS	`weekend`	277.8±40.9	46.2±11.0	2478.2±818.9
AGCRN	`pm_rush`	51.2±4.4	43.7±6.8	936.6±169.5
AGCRN	`weekend`	44.4±5.6	30.3±8.7	1215.2±99.1

We compare the original end-to-end graph learners against two projection-based variants: **MeanProj**, which removes the spatial mean at each time step, and **PCA**, which removes the top principal component estimated from the training data.

The synthetic construction separates the invariant local interaction graph from the added global factor. In Table 4, the shortcut-oriented linear baseline has a slightly smaller reported OOD gap than the oracle local-graph baseline at $s = 1.5$ and $s = 2.0$. This complements the distinction between predictive usefulness and structural correctness emphasized by Theorem 1, but is not a direct verification of its population-risk comparison; a smaller OOD gap alone does not establish a lower absolute OOD MSE.

Additional projection ablations. We also tested two synthetic ablations that further sharpen the mechanism claim. First, a projection-direction control compares MeanProj and PCA-PC1 against misaligned alternatives such as PCA-PC2 and RandomProj. At the high-shift setting $s = 2.0$, the aligned projections consistently achieve smaller OOD gaps, supporting that the gain comes from suppressing the dominant shared shortcut direction rather than from arbitrary denoising. Second, in Table 15, rank two gives the lowest reported gap for MTGNN and GTS, while rank one gives the lowest gap for AGCRN; rank four is higher than the best lower-rank setting for each model.

C.4 Summary of empirical implications

Taken together, the experiments support the theory at two levels. First, the contamination-control benchmark shows that persistent global factors can materially affect both graph learning and optimization signals. Second, the OOD benchmarks demonstrate that removing the dominant global direction before graph learning often improves robustness under shift, especially for shortcut-prone graph learners.

These results support the broader interpretation of the paper: the central issue is not merely that graph learning can fail in some abstract sense, but that end-to-end graph learners can systematically absorb global shortcut modes into the learned graph. Projection-based decoupling offers a simple and model-agnostic way to mitigate that failure mode.

Table 14: Final results on the shortcut-contamination control benchmark. "No" denotes `no_shortcut`, and "With" denotes `with_shortcut`.

Model	Test Loss (No)	Test Loss (With)	AUROC (No)	AUROC (With)
Graph WaveNet	0.0490	0.0579	0.4219	0.5563
MTGNN	0.0497	0.0580	0.5172	0.5813
GTS	0.6842	0.5311	0.5313	0.5125
DGCRN	1.2005	0.9862	0.4859	0.4391
AGCRN	0.7756	0.8251	0.5938	0.5063

Table 15: Projection-rank ablation at $s = 2.0$: OOD Gap MSE (lower is better). Original and PCA-rank1 are the ablation-suite references also shown in Table 2, not the separate strength-sweep results in Table 5. Rank two gives the lowest gap for MTGNN/GTS and rank one for AGCRN.

Model	Original	PCA-rank1	PCA-rank2	PCA-rank4
MTGNN	0.270	0.147	**0.140**	0.145
GTS	0.322	0.185	**0.143**	0.144
AGCRN	0.170	**0.136**	0.139	0.144

D Reproducibility Details

Data splits and preprocessing. For the synthetic benchmark, we use the configuration described in the main text: $N = 8$ nodes, lag order $p = 3$, input length 24, prediction horizon 1, training sequence length 2600, and OOD test sequence length 1400. Within the training environment, windows are split into train/validation/ID-test partitions with ratios $0.70/0.15/0.15$. The main synthetic OOD sweep uses test strengths $\{0.0, 0.5, 1.0, 1.5, 2.0\}$, and the additional projection-direction/rank ablations also include $s = 0.25$. For METR-LA and PEMS-BAY, we construct sliding windows of length 12 with one-step-ahead prediction. Training, validation, and ID test windows are sampled from weekday off-peak segments (10:00–16:00), with day-level train/validation/ID-test partitions of $0.70/0.15/0.15$. OOD windows are drawn from three held-out scenarios: weekday morning rush (7:00–9:00), weekday evening rush (17:00–19:00), and weekend daytime (10:00–16:00). The same split protocol is used for METR-LA and PEMS-BAY.

Projection and output completion. For MeanProj and PCA, the projector is applied to both the input histories and their targets. MeanProj fixes the all-ones direction, while PCA estimates its direction from the training split; both projectors are then held fixed. Each backbone is trained against the projected target with its original loss form. At evaluation, the removed coordinate from the last observed vector, $Q_q x_t$, is added to the projected-series forecast as in Eq. (22); all submitted real-data experiments use $H = 1$. The network output is not passed through an additional hard P_q layer, although loss against $P_q x_{t+1}$ penalizes output in the removed direction. Reported raw-coordinate errors therefore measure the complete projected-series-plus-persistence pipeline. The PCA direction is computed from centered training inputs only and is fixed thereafter.

Optimization, seeds, and fixed settings. All graph forecasting models are trained with Adam-based optimization through the released wrapper scripts. On the synthetic benchmark, all graph models are trained for 80 epochs. On METR-LA and PEMS-BAY, all graph-model runs use 15 epochs per seed and shift. The default batch size is 64 for Graph WaveNet, MTGNN, GTS, and AGCRN, and 32 for DGCRN. The learning rate is 10^{-3} for Graph WaveNet, MTGNN, AGCRN, and DGCRN, and 5×10^{-3} for GTS; MTGNN additionally uses weight decay 10^{-4}. Unless otherwise noted, synthetic experiments use seed 42. All real-data results reported in the main paper are averaged over three seeds $\{0, 1, 2\}$. For each architecture, the optimizer and model hyperparameters are held fixed across Original, MeanProj, and PCA variants. Except for the explicitly separate Eval-only control, the projection variants use the full projected-series training and causal completion defined in Section 3.1.

Compute and runtime. All experiments are executed in a single-device setting via the `-device` flag exposed by every wrapper; no distributed or multi-GPU training is required by our pipeline. In practice, the synthetic sweeps are substantially cheaper than the full three-seed real-data benchmarks,

while the full METR-LA and PEMS-BAY evaluations dominate total wall-clock time because each seed independently trains every model-shift pair. Because wall-clock timing depends strongly on hardware and on the underlying third-party implementation, we release fixed seeds, preprocessing scripts, and command-line configurations rather than treating machine-specific runtime as a scientific claim.

E BROADER IMPACTS AND LIMITATIONS

Our motivating application is traffic forecasting under distribution shift. Improving robustness in this setting can have positive downstream effects for congestion monitoring, incident response, and resource allocation, especially when systems are deployed across recurring regime changes such as rush-hour transitions or weekend behavior. More generally, the paper illustrates a simple way to reduce vulnerability to train-specific shared components in multivariate forecasting systems that include a learned relational component.

At the same time, forecasting systems in transportation can affect operational decisions, and incorrect predictions can create real costs. Overconfident deployment could misallocate roadway interventions, distort travel-time estimates, or propagate errors into downstream control and planning systems. For that reason, we view the proposed projection step as a robustness aid rather than a replacement for domain validation, fallback heuristics, or human oversight in safety-relevant settings.

The paper also has clear scientific limitations. Our method does *not* guarantee recovery of the true causal graph or even the true physical road network. Its goal is narrower: to reduce shortcut contamination from dominant shared components before graph learning. Likewise, the risk identities are for specified one-step population models, and the optimization result is a conditional local statement, not a global identifiability or training-convergence theorem. Empirically, the gain is strongest for shortcut-prone graph learners and is not uniform across all architectures or shifts. Finally, our real-data evaluation focuses on traffic time-of-day shifts; broader deployment claims would require testing on additional domains, sensor failures, seasonal changes, and intervention-driven nonstationarities.

We do not claim a calibrated one-number rule for deciding when projection is beneficial. Graph sparsity alone is insufficient, and `grad_graph_ratio` is a matched-condition, within-architecture diagnostic rather than a universal threshold. Candidate evidence includes instability of the dominant component across pre-specified chronological validation environments, learned-edge instability across regimes or seeds, and a reproducible worst-environment gain with an acceptable in-distribution cost. These criteria remain to be calibrated without access to the deployment environment.

Several extensions are left for future work. A causal rolling PCA direction could track a time-varying loading, but may be noisy or remove transient local signal; we have not evaluated it. A fixed physical road graph can be retained by a compatible backbone, but cannot restore feature variation already removed by an approximate projector. Higher-rank or nonlinear shared structure and applications such as energy demand, environmental sensing, or physiological forecasting are plausible settings for further study, not evaluated claims of the present work.

F ASSETS AND LICENSES

Our experiments combine public datasets, public reference implementations, and lightweight wrapper code in this repository. The real-data benchmarks are METR-LA and PEMS-BAY. METR-LA is accessed through the public DCRNN data release, and PEMS-BAY is accessed through the public Zenodo release used by prior traffic-forecasting work. The raw datasets remain subject to the terms, licenses, and redistribution policies specified by their original maintainers and hosting platforms; our repository provides preprocessing scripts and experiment wrappers, but does not relicense the raw data.

For model implementations, we rely on the public reference repositories for Graph WaveNet, MTGNN, GTS, AGCRN, and DGCRN (via the Traffic-Benchmark codebase), together with our own experiment wrappers and preprocessing pipeline. At the time of writing, the repository-level licenses are MIT for Graph WaveNet, MTGNN, AGCRN, and Traffic-Benchmark, and Apache License 2.0 for GTS. Users

who redistribute code or trained artifacts should consult the corresponding upstream repositories and preserve the required notices.

All additional code written for this paper—including synthetic-data generation, real-data preprocessing, multiseed runners, projection controls, and evaluation scripts—is contained in this repository and should be cited and reused together with the upstream licenses and dataset terms listed above.