RAP: RECONSTRUCTION-AUGMENTED PREDICTORS FOR ROBUST TIME-SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-term time-series forecasting (LTSF) supports applications in energy, traffic, and healthcare, yet state-of-the-art models remain brittle when inputs are corrupted by sensor or preprocessing noise. To resolve this problem, we introduce a *reconstruction-augmented predictor* (RAP) that improves robustness *without* corrupting or filtering training inputs by coupling the standard forecasting loss with a self-reconstruction objective on the history window. A single encoder feeds two linear heads—reconstruction and prediction—so the shared representation is explicitly regularized to preserve fine-grained, multi-scale structure. For small input noise, we relate the predictor's Jacobian energy to noisy forecasting risk and bound the residual Jacobian of a fitted full-sequence map under explicit distributional and curvature assumptions. Controlled ablations measure the effect of reconstruction on the fitted predictor. Across representative LTSF benchmarks in high-dimensional datasets, RAP remains competitive with strong baselines in clean forecasting and degrades more gracefully under synthetic corruptions, and its advantage grows with the length of the observed context. Our measurement confirms that the reconstruction term lowers the predictor's Jacobian energy, and controlled comparisons against Gaussian-noise training, low-pass filtering and an explicit Jacobian penalty on the same backbone isolate the effect of the objective.

1 INTRODUCTION

Time-series data play a central role across power systems (Susto et al., 2018), finance (Sezer et al., 2020), and healthcare (Kaushik et al., 2020; Morid et al., 2023). Building on the success of deep learning in numerous areas such as image generation and classification, deep learning approaches have recently achieved strong performance on time-series tasks (Lim & Zohren, 2021; Torres et al., 2021; Lara-Benítez et al., 2021). Within this landscape, long-term time-series forecasting (LTSF)—predicting a horizon H steps ahead from a history window of length T—has emerged as a major focus due to its practical importance and modeling challenges such as nonstationarity (Liu et al., 2022c), multi-scale dynamics (Franceschi et al., 2019; Tonekaboni et al., 2021; Yang & Hong, 2022; Yue et al., 2022; Liu et al., 2022b), and covariate shift (Ulyanov et al., 2016; Kim et al., 2021).

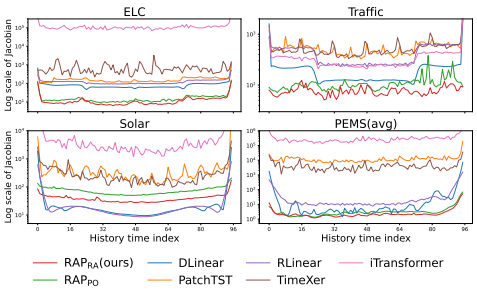


Figure 1: Log of the summed Jacobian energy w.r.t. each history step, averaged over prediction length 96, 192, 336, 720 on our main datasets. Lower curves indicate reduced input-noise sensitivity (greater robustness). We use the Hutchinson estimator for calculating the Jacobian by following Algorithm 3.

Despite these advances, state-of-the-art forecasters are often brittle when observations are corrupted by noise. Notably, Figure 3 shows that transformer-based SOTA models are easily deteriorated by small perturbations. Corruptions arise naturally in real sensors and pipelines (e.g., quantization, jitter, missingness, clock drift) and may also be adversarial or unmodeled (e.g., abrupt spikes, regime changes). Common robustness strategies include (i) augmenting the training set with Gaussian noise (Tashiro et al., 2021; Rasul et al., 2021; Kollovieh et al., 2023) and (ii) suppressing high-frequency components via filtering (Zhou et al., 2022; Wu

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.

et al., 2023; Piao et al., 2024). In high-frequency regimes, both approaches can discard predictive fine-grained, multi-scale structure, thereby degrading accuracy on complex signals (cf. Section 2). Gaussian noise training induces a smoothing bias in the population predictor, and frequency-domain filtering can remove informative transients.

To address these issues, we propose a *reconstruction-augmented predictor* (RAP) that enhances robustness *without* augmenting the training data. Concretely, our model produces both a forecast and a self-reconstruction term from the same shared representation (cf. Eq. 3). The reconstruction loss supervises the history coordinates of the joint output. We analyze how the residual of a fitted full-sequence map relates to its Jacobian, and measure the resulting predictor sensitivity $\mathbb{E}_{\mathbf{x}_{1:T}} \|J_{\text{pred}}(\mathbf{x}_{1:T})\|_F^2$ in controlled comparisons (Figure 1).

Complementing this objective, we adopt a *dual-axis Transformer* (Vaswani et al., 2017) that alternates attention along temporal and feature (channel) axes to capture cross-time and cross-variable dependencies efficiently. The decoder generates the full target sequence conditioned on the history window, inspired by DiffWave-style sequence generation (Kong et al., 2020); this design naturally accommodates both prediction-only training and reconstruction-augmented training under a single architecture.

On the evaluated benchmarks, RAP is competitive in clean forecasting and often degrades more slowly than standard baselines as test-time synthetic corruption levels increases. We further analyze failure modes of augmentation-based robustness (Section 2), detail the model architecture, training and inference process (Section 5), analyze small-noise sensitivity under explicit regularity conditions (Section 4), and present extensive experiments and ablations (Section 7.2 and 7.3).

To summarize, our contributions are:

- We identify and theoretically characterize failure modes of augmentation- and filtering-based robustness for time-series forecasting: Gaussian corruption training induces a smoothing bias that can deteriorate predictive high-frequency structure, and low-pass filtering suppresses temporal components that can remain predictive (cf. Section 2).
- We introduce *RAP*, a no-augmentation robustness regularizer and reconstruction-augmented training objective that jointly forecasts the future and reconstructs the history within a same-shape architecture; under explicit regularity conditions, a fitted full-sequence residual yields an upper bound on the predictor's Jacobian sensitivity.
- Across diverse, complex datasets, our approach remains competitive in clean forecasting and degrades more slowly under corruption than both standard forecasters and robustness-oriented objectives on the same backbone.
- We show that the objective transfers to other backbones when the reconstruction and prediction paths share the output map, that its robustness benefit increases with context length up to $C = 720$.

2 Motivation: Failure modes of previous robustness techniques

There have been three types of machine learning methods for robustness: directly regularizing the Jacobian norm technique(Hoffman et al., 2019), inserting a Gaussian on input data (Tashiro et al., 2021; Rasul et al., 2021; Kollovieh et al., 2023), and removing high-frequency noise, e.g., Fourier transform (Zhou et al., 2022; Wu et al., 2023; Piao et al., 2024). Direct Jacobian regularization is costly for large forecasters, contradicting our main goal of a simple, adaptable robust objective. It also penalises $\|J_{\text{pred}}\|$ with no reference point, and can therefore suppress sensitivity that is genuinely predictive; the objective we propose instead bounds the predictor's sensitivity *relative to* that of the Bayes forecasting map (Theorem 4.3). On the same backbone, the explicit penalty is both slower (42.1 vs. 26.6 ms/iter) and less robust than our objective (Table 4 in Appendix D.1). We therefore focus on the latter two: Gaussian perturbation of the history window and low-pass conditioning. We theoretically characterize the following failure mode:

Gaussian noise injection training and low-pass filtering for robust time-series prediction can deteriorate predictive high-frequency structure in the original history data.

Setup. We study multi-step univariate time-series forecasting under an unknown distribution μ over history–future pairs (X, Y). Let $X \in \mathbb{R}^d$ with density p_X and Bayes regressor is $m(x) := \mathbb{E}[Y \mid X = x]$. Assume p_X, m in d-dimensional Schwartz space $\mathcal{S}(\mathbb{R}^d)$ with $p_X > 0$ everywhere.[1] We evaluate clean-test risk $R_0(f) := \mathbb{E}[(f(X) - Y)^2]$ (expectation w.r.t. p_X), and write the score $s(x) := \nabla \log p_X(x)$. Denote by $\varphi_\sigma(x) = (2\pi\sigma^2)^{-d/2} \exp(-\|x\|^2/(2\sigma^2))$ and by

$$(e^{t\Delta} g)(x) = \big(g * \phi_t\big)(x), \quad \phi_t(x) = (4\pi t)^{-d/2} \exp\Big(-\|x\|^2/(4t)\Big), \quad t > 0.$$

2.1 Gaussian-noise training on $\mathbb{R}^d$

We train with corrupted inputs $U := X + Z$, $Z \sim \mathcal{N}(0, \sigma^2 I_d)$ independent of (X, Y), by minimizing $\mathcal{L}_\sigma(f) = \mathbb{E}[(f(U) - Y)^2]$ over a rich function class.

Proposition 2.1. *The population minimizer is the conditional expectation*

$$f_\sigma(u) = \mathbb{E}[Y \mid U = u] = \mathbb{E}[m(X) \mid U = u] = \frac{\big((m\, p_X) * \varphi_\sigma\big)(u)}{\big(p_X * \varphi_\sigma\big)(u)}.$$

Moreover, as $\sigma \downarrow 0$,

$$f_\sigma(x) = m(x) + \frac{\sigma^2}{2}\big(\Delta m(x) + 2\nabla m(x) \cdot s(x)\big) + O_K(\sigma^4), \tag{1}$$

uniformly on every compact $K \subset \mathbb{R}^d$. Assume in addition that the heat-ratio expansion remainder is $O(\sigma^4)$ in $L^2(p_X)$. Consequently,

$$\Delta R_0^{\text{noise}}(\sigma) = \frac{\sigma^4}{4}\, \mathbb{E}\Big[\big(\Delta m + 2\nabla m \cdot s\big)^2\Big] + O(\sigma^6). \tag{2}$$

By Plancherel on $\mathbb{R}^d$, $\|\Delta m\|^2_{L^2(dx)} = \int \|\xi\|^4 |\widehat{m}(\xi)|^2\, d\xi$, so the leading term in equation 2 accentuates high frequencies. Thus, Gaussian-noise training changes the population target from m to a heat-smoothed predictor, and the clean-risk gap grows with predictive high-frequency energy. If the score s is bounded on the support of X, then

$$\mathbb{E}\big[(\Delta m + 2\nabla m \cdot s)^2\big] \;\geq\; \tfrac{1}{2}\, \mathbb{E}[(\Delta m)^2] - 2\, \|s\|^2_\infty\, \mathbb{E}[\|\nabla m\|^2],$$

so HF-dominant m makes $\Delta R_0^{\text{noise}}(\sigma)$ large even for small σ.

2.2 Low-pass conditioning on $\mathbb{R}^d$

Let $P_{\leq \omega_c}$ denote a fixed low-pass projection applied to the history window. In the time-series setting, this projection acts along the temporal frequency axis of the observed history. The model only receives $U := P_{\leq \omega_c} X$. The population optimal predictor is $f^{\text{LPF}}(x) := \mathbb{E}[m(X) \mid U = P_{\leq \omega_c} x]$.

Proposition 2.2. *For any p_X,*

$$R_0\big(f^{\text{LPF}}\big) - R_0(m) = \mathbb{E}[\mathit{Var}(m(X) \mid P_{\leq \omega_c} X)] = \inf_h\, \mathbb{E}\Big[\big(m(X) - h(P_{\leq \omega_c} X)\big)^2\Big].$$

where the infimum is over all measurable $h : \mathbb{R}^d \to \mathbb{R}$.

In images, mild manipulation of high frequencies often preserves perceptual quality; in LTSF, however, fine, phase-sensitive structure at high frequencies is often *predictive* and the above propositions formalize this. Thus, in high-frequency regimes, these tactics can deteriorate predictive microstructure in the original history data, making them unreliable generic robustness fixes for time series and motivating our reconstruction-augmented approach. We conduct several verification experiments by constructing toy datasets and the results are in Appendix A.1, where we also show that the reconstruction objective introduced below recovers the predictive high-frequency structure removed by low-pass training (Table 5).

[1] The Schwartz assumption is a standard sufficient condition for the heat-kernel expansion used below.

3 PRELIMINARIES

Inspired by failure modes of previous robustness techniques, we propose a *reconstruction-augmented predictor* (RAP) that seeks to improve robustness without modifying the training inputs.

3.1 PROBLEM STATEMENT

Setting and notation. We study multi-step univariate time-series forecasting under an unknown distribution μ over history–future pairs $(X, Y) \in \mathbb{R}^T \times \mathbb{R}^H$.[2] Here X collects the most recent T samples in a lookback window, and Y denotes the next H samples to be predicted. We write $\mathbb{E}[\cdot]$ for expectation with respect to the relevant variables (explicitly indicated when needed), $\|\cdot\|$ for the Euclidean norm, $\|\cdot\|_{\mathrm{op}}$ for the spectral (operator) norm, and $\|\cdot\|_F$ for the Frobenius norm.

A single encoder $\Phi_\theta : \mathbb{R}^T \to \mathbb{R}^D$ produces a shared representation $z = \Phi_\theta(x)$, on which we place two linear heads:

$$f_{\mathrm{rec}}(x) = W_c\,\Phi_\theta(x) \in \mathbb{R}^T, \quad f_{\mathrm{pred}}(x) = W_p\,\Phi_\theta(x) \in \mathbb{R}^H,$$

where $W_c \in \mathbb{R}^{T\times D}$ and $W_p \in \mathbb{R}^{H\times D}$. The *reconstruction* head f_{rec} reconstructs the input history X and serves as a structure-preserving self-supervised signal, while the *prediction* head f_{pred} forecasts Y using the same representation. For later use, we denote the Jacobian with respect to the input:

$$J_\Phi(x) := \nabla_x \Phi_\theta(x) \in \mathbb{R}^{D\times T}, \quad J_{\mathrm{rec}}(x) := \nabla_x f_{\mathrm{rec}}(x), \quad J_{\mathrm{pred}}(x) := \nabla_x f_{\mathrm{pred}}(x).$$

In our implementation, we take W_c and W_p as fixed selection matrices that split the network output into $[\hat{x}_{1:T}, \hat{y}_{1:H}]$; thus, the only learned parameters are those of the encoder.

Training objectives. Conventional forecasters ignore f_{rec} and optimize only the prediction loss

$$\mathcal{L}_{\mathrm{pred}}(\theta, W_p) := \mathbb{E}_{(X,Y)\sim\mu}\,\|f_{\mathrm{pred}}(X) - Y\|^2.$$

In our reconstruction-augmented formulation, we add a reconstruction term and minimize the composite objective

$$\mathcal{L}_{\mathrm{rec}}(\theta, W_c) := \mathbb{E}_X\left\|f_{\mathrm{rec}}(X) - X\right\|^2,$$
$$\mathcal{L}_{\mathrm{PO}}(\theta, W_p) := \mathcal{L}_{\mathrm{pred}}(\theta, W_p), \qquad \mathcal{L}_{\mathrm{RA}}(\theta, W_c, W_p) := \mathcal{L}_{\mathrm{pred}}(\theta, W_p) + \mathcal{L}_{\mathrm{rec}}(\theta, W_c).$$

This regularizer encourages the encoder to retain fine-grained, multi-scale information in X that is predictive for Y, without requiring data augmentation. Specifically, we refer to the two settings as prediction-only ($\mathrm{RAP}_{\mathrm{PO}}$) and reconstruction-augmented ($\mathrm{RAP}_{\mathrm{RA}}$).

For convenience, we instantiate a shared latent of size $D = T + H$ and choose $W_c = [\mathbf{I}_T, \mathbf{0}]$ and $W_p = [\mathbf{0}, \mathbf{I}_H]$ so that the model can generate the concatenated sequence $[f_{\mathrm{rec}}(x),\ f_{\mathrm{pred}}(x)] \approx [X, Y]$. Therefore, it is straightforward that the $\mathcal{L}_{\mathrm{RA}}$ is represented by

$$\mathcal{L}_{\mathrm{RA}}(\theta) = \mathbb{E}_{X,Y}[\,\|\underbrace{\hat{Y} - Y}_{\text{prediction}}\|^2 + \underbrace{\|\hat{X} - X}_{\text{reconstruction}}\|^2], \tag{3}$$

where $\hat{X}, \hat{Y}$ are history, prediction part of $\Phi_\theta(X)$. This output interface enables a controlled comparison under fixed model capacity; the analysis applies to the concatenated outputs of a two-head predictor under the regularity assumptions in Section 4. At inference, we report forecasts from f_{pred}. Because the model output is the concatenation $[\hat{x}_{1:T}, \hat{y}_{1:H}]$ and $\mathcal{L}_{\mathrm{RA}}$ is the squared error on this full sequence:

$$\mathcal{L}_{\mathrm{RA}} = \|[x_{1:T}, y_{1:H}] - [\hat{x}_{1:T}, \hat{y}_{1:H}]\|^2 \tag{4}$$

the reconstruction and prediction terms share unit weight by construction. No additional hyperparameters are introduced.

[2] We point out that the univariate arguments can be extended directly to the multivariate setting: replace derivatives by Jacobian and scalar inner products by trace forms. We refer to Section 4.

3.2 Robust prediction under noisy inputs

To capture robustness to sensor, quantization, or preprocessing noise at deployment, we evaluate the *noisy risk*

$$\mathcal{R}_\sigma(f_{\text{pred}}) := \mathbb{E}_{X,Y,\varepsilon} \left\| f_{\text{pred}}(X + \sigma\varepsilon) - Y \right\|^2, \quad \varepsilon \sim \mathcal{N}(0, I) \text{ independent of } (X, Y),$$

and compare it to the clean risk $\mathcal{R}_0(f_{\text{pred}}) = L_{\text{pred}}$. Here σ controls the noise level and I denotes the identity.

Assuming differentiability and square-integrability of the Jacobian, a first-order Taylor expansion around X yields, for small $\sigma > 0$,

$$\mathbb{E}_{X,Y,\varepsilon} \| f_{\text{pred}}(X + \sigma\varepsilon) - Y \|^2 \simeq \mathbb{E}_{X,Y} \| f_{\text{pred}}(X) - Y \|^2 + \sigma^2 \, \mathbb{E}_X \| J_{\text{pred}}(X) \|_F^2.$$

Following standard small-noise analysis, we define the *noise sensitivity*

$$S := \mathbb{E}_X \| J_{\text{pred}}(X) \|_F^2, \tag{5}$$

which serves as a first-order proxy for the increase $\mathcal{R}_\sigma - \mathcal{R}_0$ when σ is small. Intuitively, S measures how strongly the predictor amplifies infinitesimal, isotropic perturbations of the input; lower S indicates a more stable (robust) mapping. Section 4 gives a conditional upper bound on S in terms of the fitted full-sequence residual and the Bayes-map sensitivity.

4 Small-noise robustness via Jacobian energy

In this section, we analyze small additive input noise using standard regularity assumptions for Taylor and finite-difference arguments. We work directly with the full output map of the model. Let

$$F(x) := [f_{\text{rec}}(x), f_{\text{pred}}(x)] \in \mathbb{R}^{d+k}, \quad G(x) := [x, m(x)] \in \mathbb{R}^{d+k},$$

where $m(x) := \mathbb{E}[Y \mid X = x]$ denotes the Bayes regressor. This formulation matches our architecture, which predicts the full sequence under a single forward pass.

Assumption 4.1. The predictor $f_{\text{pred}} \in C^3(\mathbb{R}^d; \mathbb{R}^k)$ has bounded second- and third-order derivatives, denoted by H_f and M_f. The input distribution admits a positive C^1 density with bounded score $\|\nabla \log p_X\|_\infty \le L_p$.[3] The Bayes map m has finite Jacobian energy, and the residual map $r := F - G$ is C^2 with bounded curvature $H_r := \sup_x \|\nabla^2 r(x)\|_{\text{op}} < \infty$.

The smoothness part of Assumption 4.1 gives the small-noise expansion.

Theorem 4.2. *Let $\varepsilon \sim \mathcal{N}(0, \mathbf{I})$ be independent of (X, Y) and define $\mathcal{R}_\sigma(f_{\text{pred}}) := \mathbb{E}_{X,Y,\varepsilon} \left\| f_{\text{pred}}(X + \sigma\varepsilon) - Y \right\|^2$ and $\mathcal{R}_0(f_{\text{pred}}) := \mathcal{L}_{\text{pred}}$. Under Assumption 4.1,*

$$\mathcal{R}_\sigma(f_{\text{pred}}) = \mathcal{R}_0(f_{\text{pred}}) + \sigma^2 \, \mathbb{E} \| J_{\text{pred}}(X) \|_F^2 + \sigma^2 \, \mathbb{E} \, \langle f_{\text{pred}}(X) - Y, \Delta f_{\text{pred}}(X) \rangle + O(\sigma^3) \tag{6}$$

where Δ is the componentwise Laplacian. In particular, for constants $c_{d,k}, C$ depending on dimension and derivative bounds,

$$\mathcal{R}_\sigma(f_{\text{pred}}) - \mathcal{R}_0(f_{\text{pred}}) \le \sigma^2 \, \mathbb{E} \| J_{\text{pred}}(X) \|_F^2 + c_{d,k} \, \sigma^2 \, H_f \sqrt{\mathcal{L}_{\text{pred}}} + C \, \sigma^3.$$

By Theorem 4.2, when the model is well-trained, $L_{\text{pred}} = \mathbb{E}\|f_{\text{pred}}(X) - Y\|^2$ becomes small, the Laplacian inner-product term in equation 6 is negligible compared to the Jacobian energy. Thus the *noise sensitivity* $S = \mathbb{E}\|J_{\text{pred}}(X)\|_F^2$ governs the dominant σ^2-order increase in risk, and reducing S directly improves small-noise robustness.

Since $m(X) = \mathbb{E}[Y \mid X]$, the prediction loss decomposes as

$$\mathbb{E}\|f_{\text{pred}}(X) - Y\|^2 = \mathbb{E}\|f_{\text{pred}}(X) - m(X)\|^2 + \mathbb{E}\|Y - m(X)\|^2.$$

Therefore,

$$\begin{aligned} \mathbb{E}\|F(X) - G(X)\|^2 &= \mathbb{E}\|f_{\text{rec}}(X) - X\|^2 + \mathbb{E}\|f_{\text{pred}}(X) - m(X)\|^2 \\ &= L_{\text{RA}} - \mathbb{E}\|Y - m(X)\|^2 \le L_{\text{RA}}. \end{aligned}$$

The last term is the irreducible Bayes noise and is independent of the predictor. Hence, $\mathcal{L}_{\text{RA}}$ controls $\mathbb{E}\|F(X) - G(X)\|^2$, which lets us bound the residual Jacobian through the following theorem:

[3] Score boundedness or regularity is common in score-based and diffusion generative-model analyses, where $\nabla \log p$ is the modeled object; see Hyvärinen (2005); Song et al. (2021); Tashiro et al. (2021).

Theorem 4.3. *Under Assumption 4.1, for any $0 < t \le t_{\max}$, there exist constants $c_d, C_d > 0$, depending on $d, L_p, t_{\max}$, such that*

$$\mathbb{E}\|J_F(X) - J_G(X)\|_F^2 \le \frac{c_d}{t^2}\,\mathbb{E}\|F(X) - G(X)\|^2 + C_d H_r^2 t^2,$$

with the bounded-score factor absorbed into c_d. Therefore, the sensitivity becomes:

$$S := \mathbb{E}\|J_{\text{pred}}(X)\|_F^2 \le 2\,\mathbb{E}\|J_m(X)\|_F^2 + \frac{2c_d}{t^2} L_{\text{RA}} + 2C_d H_r^2 t^2.$$

If $t_\star := \left(\frac{c_d L_{\text{RA}}}{C_d H_r^2}\right)^{1/4} \le t_{\max}$, *then* $S \le 2\,\mathbb{E}\|J_m(X)\|_F^2 + 4(c_d C_d)^{1/2} H_r\, L_{\text{RA}}^{1/2}$.

Thus, reducing $\mathcal{L}_{\text{RA}}$ controls the predictor-dependent full-sequence error relative to $[X, m(X)]$ up to Bayes noise, and controls the leading Jacobian sensitivity term. Combined with Theorem 4.2, this reduces the leading noisy-risk increase up to the intrinsic sensitivity of the Bayes map. The effect of adding reconstruction on the fitted forecast is evaluated through controlled ablations in Section 7.3.

Intuition. The reconstruction term supplies supervision on every history coordinate through parameters shared with forecasting. Theorem 4.3 characterizes the Jacobian of the fitted joint map under its regularity conditions. The change caused by adding reconstruction depends on the learned forecast: in our controlled RAP ablation, the measured Jacobian energy decreases on ECL ($1.500 \to 1.240$) and Traffic ($3{,}434 \to 3{,}207$; Appendix D.6).

Scope and multivariate extension. For clarity, several theoretical statements in Section 4 are presented for the univariate case. The arguments extend directly to the multivariate setting: replace derivatives by Jacobian and scalar inner products by trace forms. Under channel-correlated Gaussian input noise $\varepsilon \sim \mathcal{N}(0, \Sigma)$ with $\Sigma \succeq 0$, the small-noise expansion holds with the sensitivity term weighted by Σ, i.e., $\mathbb{E}\,\mathrm{tr}\big(J_{\text{pred}}(X)\,\Sigma\,J_{\text{pred}}(X)^\top\big)$.

5 RAP: Reconstruction-Augmented Predictors

In Section 4, we showed that the reconstruction-augmented full-sequence objective controls the dominant small-noise term of the predictor up to the intrinsic sensitivity of the target forecasting map. Here, we detail how we instantiate the model and how we train and evaluate it for both accuracy and robustness (see Figure 2 for an overview).

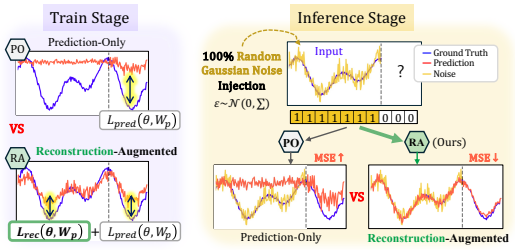


Figure 2: Overall visualization of the training and inference process

Given a history $\mathbf{x}_{1:T}$ and target $\mathbf{y}_{1:H}$, RAP produces a forecast $\hat{\mathbf{y}}_{1:H}$ and a self-reconstruction $\hat{\mathbf{x}}_{1:T}$ from a shared representation. We train the model by using Equation 3. Again, we refer to the two settings as prediction-only (RAP_{PO}) and reconstruction-augmented (RAP_{RA}). Note that both share the *same* architecture and generate the full sequence; they differ only in whether the reconstruction term is optimized (cf. Appendix E). Neither variant sees corrupted inputs during training. By default, we use MSE for both terms and optimize with Adam and a cosine learning-rate scheduler.

At inference time, we evaluate both standard LTSF accuracy and robustness under input noise. For robustness, we inject i.i.d. Gaussian noise into the history window, $\mathbf{x}_{1:T} \mapsto \mathbf{x}_{1:T} + \sigma\varepsilon$ with $\varepsilon \sim \mathcal{N}(0, I)$, and report the MSE/MAE as σ increases. We compare RAP_{RA}, RAP_{PO}, and representative SOTA baselines under identical settings in Section 7.3. This protocol directly probes the small-noise sensitivity analyzed in Section 4 while holding the forecasting setup fixed. We describe detailed training and inference processes in Algorithm 2 and 1 in Appendix F. In Section 7.3, we additionally sweep σ to moderate/large values as an empirical stress test. Results in this non-perturbative regime should be interpreted as empirical evidence beyond the asymptotic guarantee.

6 Related Work

In this section, we summarize representative approaches that we partly include as baselines.

For Decomposition-centric models, Autoformer (Wu et al., 2021) introduces a progressive decomposition block and an auto-correlation mechanism that replaces point-wise self-attention, improving efficiency while explicitly disentangling trend and seasonal components. This decomposition perspective motivated follow-ups such as FEDformer (Zhou et al., 2022), DLinear (Zeng et al., 2023), which further explore how separating low- and high-frequency behavior can stabilize long-horizon prediction. RLinear (Li et al., 2023) further highlights the role of normalization, combining linear forecasting with Reversible Instance Normalization (RevIN) (Kim et al., 2021) to stabilize training and improve generalization across datasets.

TimesNet (Wu et al., 2023) leverages the Fourier transform to decompose signals into frequency components with varying periods, strengthening modeling of multi-scale temporal variation. Beyond forecasting, TimesNet is proposed as a task-general backbone for imputation, anomaly detection, and classification, and its official codebase offers a standardized evaluation protocol and unified baselines.

Next, segmentation and patch tokens are applied to Crossformer (Zhang & Yan, 2023) and PatchTST (Nie et al., 2023) segment long sequences into subseries-level patches that act as input tokens for Transformers. iTransformer (Liu et al., 2024) utilized cross-variate attention at scale, taking an "inverted" view that embeds time points into variate tokens while shifting attention from temporal correlations to cross-variate correlations.

Table 1: Full results of the long-term multivariate forecasting task (MSE / MAE). Ranking notation: lower is better; 1st=**bold**, 2nd=underline, 3rd=*emph*. Seed variability is reported in Appendix D.7.

Dataset	Horizon	**RAP$_{\text{RA}}$(Ours)**		TimeXer		iTrans.		RLinear		PatchTST		Cross.		TimesNet		DLinear		Auto.	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ECL	96	*0.149*	*0.247*	**0.140**	0.242	0.148	**0.240**	0.201	0.281	0.195	0.285	0.219	0.314	0.168	0.272	0.197	0.282	0.201	0.317
	192	*0.173*	*0.274*	**0.157**	0.256	0.162	**0.253**	0.201	0.283	0.199	0.289	0.231	0.322	0.184	0.289	0.196	0.285	0.222	0.334
	336	*0.191*	*0.290*	**0.176**	0.275	0.178	**0.269**	0.215	0.298	0.215	0.305	0.246	0.337	0.198	0.300	0.209	0.301	0.231	0.338
	720	0.227	0.321	**0.211**	**0.306**	*0.225*	0.317	0.257	0.331	0.256	0.337	0.280	0.363	0.220	*0.320*	0.245	0.333	0.254	0.361
	Avg	*0.185*	*0.283*	**0.171**	**0.270**	0.178	0.270	0.219	0.298	0.216	0.304	0.244	0.334	0.192	0.295	0.212	0.300	0.227	0.338
Traffic	96	**0.312**	0.340	*0.428*	0.271	0.395	**0.268**	0.649	0.389	0.462	*0.295*	0.522	0.290	0.593	0.321	0.650	0.396	0.613	0.388
	192	**0.334**	0.345	*0.448*	0.282	0.417	**0.276**	0.601	0.366	0.466	*0.296*	0.530	0.293	0.617	0.336	0.598	0.370	0.616	0.382
	336	**0.385**	0.379	*0.473*	0.289	0.433	**0.283**	0.609	0.369	0.482	*0.304*	0.558	0.305	0.629	0.336	0.605	0.373	0.622	0.337
	720	**0.421**	0.393	0.516	0.307	0.467	**0.302**	0.647	0.387	*0.514*	*0.322*	0.589	0.328	0.640	0.350	0.645	0.394	0.660	0.408
	Avg	**0.363**	0.364	*0.466*	0.287	0.428	**0.282**	0.626	0.378	0.481	*0.304*	0.550	*0.304*	0.620	0.336	0.625	0.383	0.628	0.379
Solar	96	**0.145**	**0.168**	*0.207*	*0.259*	0.203	0.237	0.322	0.339	0.234	0.286	0.310	0.331	0.250	0.292	0.290	0.378	0.884	0.711
	192	**0.153**	**0.180**	*0.237*	*0.292*	0.233	0.261	0.359	0.356	0.267	0.310	0.734	0.725	0.296	0.318	0.320	0.398	0.834	0.692
	336	**0.210**	**0.217**	0.248	*0.293*	0.248	0.273	0.397	*0.369*	*0.290*	0.315	0.750	0.735	0.319	0.330	0.353	0.415	0.941	0.723
	720	0.348	0.292	0.258	*0.298*	**0.249**	**0.275**	0.397	0.356	*0.289*	0.317	0.769	0.765	0.338	0.337	0.356	0.413	0.882	0.717
	Avg	**0.214**	**0.214**	*0.238*	*0.286*	0.233	0.262	0.369	0.356	0.270	0.307	0.641	0.639	0.301	0.319	0.330	0.401	0.885	0.711
PEMS03	12	**0.067**	**0.170**	0.067	*0.178*	*0.070*	0.176	0.114	0.233	0.078	0.185	0.090	0.203	0.085	0.192	0.105	0.220	0.272	0.385
	24	**0.079**	**0.186**	0.088	0.204	*0.097*	*0.209*	0.193	0.309	0.121	0.230	0.121	0.240	0.118	0.223	0.183	0.296	0.334	0.440
	48	**0.122**	**0.222**	0.140	0.253	0.164	0.275	0.329	0.421	0.213	0.306	0.202	0.317	*0.155*	*0.260*	0.318	0.407	1.032	0.782
	96	**0.131**	**0.238**	0.451	0.501	0.264	*0.360*	0.455	0.513	0.364	0.410	*0.262*	0.367	0.228	0.317	0.451	0.507	1.031	0.796
	Avg	**0.100**	**0.204**	0.186	0.284	*0.149*	*0.255*	0.273	0.369	0.194	0.283	0.169	0.282	0.146	0.248	0.264	0.358	0.667	0.601
PEMS04	12	**0.074**	**0.180**	0.074	0.185	0.089	*0.194*	0.135	0.256	0.099	0.206	0.098	0.218	*0.087*	0.195	0.115	0.228	0.424	0.491
	24	**0.084**	**0.193**	0.088	0.202	*0.124*	*0.232*	0.212	0.325	0.165	0.271	0.131	0.256	0.103	0.215	0.189	0.299	0.459	0.509
	48	**0.097**	**0.210**	0.114	0.231	0.195	0.296	0.342	0.424	0.310	0.383	0.205	0.326	0.136	0.250	0.322	0.405	0.646	0.610
	96	**0.110**	**0.225**	0.162	0.283	0.479	0.497	0.443	0.496	0.527	0.515	0.402	0.457	*0.190*	*0.303*	0.428	0.483	0.912	0.748
	Avg	**0.091**	**0.202**	0.109	0.225	0.222	0.305	0.283	0.375	0.275	0.344	0.209	0.314	*0.129*	*0.241*	0.264	0.354	0.610	0.590
PEMS07	12	**0.059**	**0.156**	0.059	0.163	*0.068*	*0.168*	0.108	0.231	0.080	0.189	0.094	0.200	0.082	*0.181*	0.100	0.215	0.199	0.336
	24	**0.071**	**0.171**	0.077	0.182	*0.096*	*0.200*	0.202	0.318	0.131	0.238	0.139	0.247	0.101	0.204	0.189	0.302	0.323	0.420
	48	**0.088**	**0.189**	0.117	0.227	0.185	0.289	0.389	0.448	0.259	0.339	0.311	0.369	*0.134*	*0.238*	0.377	0.439	0.390	0.470
	96	**0.106**	**0.202**	0.159	0.266	0.214	0.311	0.584	0.547	0.429	0.445	0.396	0.442	*0.181*	*0.279*	0.580	0.539	0.554	0.578
	Avg	**0.081**	**0.179**	0.103	0.209	0.141	0.242	0.321	0.386	0.225	0.303	0.235	0.315	*0.124*	*0.226*	0.311	0.374	0.367	0.451

7 Experiments

To evaluate the proposed RAP model, we conducted two experiments. The first experiment is a general Long-term Time Series Forecasting (LTSF) [4] experiment that evaluates the predictive ability of

[4]Our baseline selection follows the Time-Series-Library repository: `https://github.com/thuml/Time-Series-Library`. Baseline implementations and recommended settings follow that protocol.

the model. In contrast, the second experiment is a robustness experiment that evaluates the model's resilience to noise. We also attach an ablation study about model architecture in Appendix D.2.

7.1 Experimental Setup

Following previous works, we use Mean Squared Error (MSE) and Mean Absolute Error (MAE) metrics to assess the performance of the LTSF models. All the experiments are implemented in PyTorch (Paszke et al., 2019), and conducted on a single Quadro RTX 6000 GPU. For the general LTSF experiment, we follow the forecasting protocol proposed by TimesNet (Wu et al., 2023) with a lookback length $T = 96$ and prediction length $H = \{96, 192, 336, 720\}$. For the robustness experiment, we evaluate noise scales $\sigma = \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7\}$.

We conduct extensive experiments on six real-world multivariate time series datasets, including Solar-energy (Lai et al., 2018), Electricity (ECL), Traffic, (Wu et al., 2021), and Performance Measurement System (PEMS) 3, 4, 7 (Chen et al., 2001) introduced in (Liu et al., 2022a). We focus on datasets that have a high number of channels and possess more complex temporal features for evaluating the robustness of the proposed approach and these regimes expose the largest robustness gaps. For baselines, we select the recent state-of-the-art LTSF models, namely Autoformer (Wu et al., 2021), DLinear (Zeng et al., 2023), TimesNet (Wu et al., 2023), Crossformer (Zhang & Yan, 2023), PatchTST (Nie et al., 2023), RLinear (Li et al., 2023), iTransformer (Liu et al., 2024), and TimeXer (Wang et al., 2024). We describe the dataset and baseline details in Appendix B.

7.2 General LTSF Experiment

In the general LTSF experiment, we evaluate the regular forecasting abilities of the prediction models. For a fair comparison, we use unified dataset settings for all models: a lookback length $T = 96$, prediction lengths $H = \{12, 24, 48, 96\}$ for the PEMS dataset, and $T = 96$, $H = \{96, 192, 336, 720\}$ for all other datasets. For RAP, we optionally enable RevIN depending on the dataset (Kim et al., 2021).

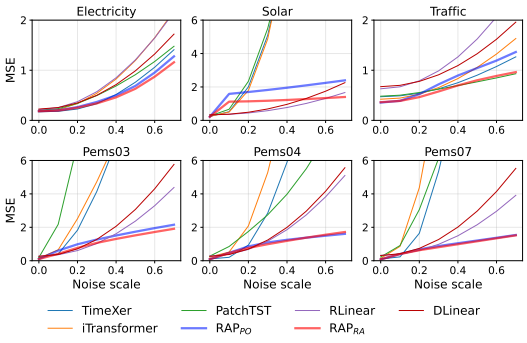


Figure 3: Average MSE vs. noise scale on six datasets. For ECL/Solar/Traffic, results are averaged over $H = \{96, 192, 336, 720\}$; for PEMS, over $H = \{12, 24, 48, 96\}$.

The clean-data experiment asks whether the auxiliary reconstruction objective preserves standard forecasting quality. Table 1 shows that RAP remains competitive with strong LTSF baselines and is often among the top performers, especially on Traffic, Solar, and the PEMS datasets. We therefore interpret the clean-setting result as compatibility evidence: reconstruction-augmented training improves robustness without requiring a systematic sacrifice in standard forecasting accuracy. When combined with the robustness gains shown in Figure 3, RAP is competitive in clean forecasting and achieves the strongest robustness profile on most high-dimensional benchmarks. The reconstruction-augmented variant offers a favorable accuracy–robustness trade-off while preserving comparable clean performance.

7.3 Experiments about Robustness

In the robustness experiment, we evaluate how well the prediction models adapt to data corrupted with random Gaussian noise. Specifically, for each noise scale σ, we sample $\varepsilon \sim \mathcal{N}(0, \mathbf{I})$ and evaluate the corrupted history $\widetilde{X} = X + \sigma\varepsilon$, simulating real-world measurement errors and sensor disturbances. We report the MSE/MAE with ground truth as σ increases. All models are trained on a pure dataset and tested under the same noise-injected conditions, with the lookback and prediction settings identical to those used in the general LTSF experiment for a fair comparison.

Figure 3 shows that RAP typically degrades more gracefully than strong baselines as the corruption level increases, with the clearest margins on the more challenging datasets. Although some baselines, such as DLinear and RLinear on the Solar dataset, achieve more robust predictions than other

Table 2: Effect of context length on Traffic (MSE). *Top:* clean (σ=0) MSE on DLinear and iTransformer. *Bottom:* relative MSE improvement (%) from adding reconstruction at σ=0.7.

	C=96	192	336	720
DLinear (clean MSE)	0.450	0.338	0.304	**0.283**
iTransformer (clean MSE)	**0.422**	**0.396**	**0.387**	0.379
Reconstruction gain (%), DLinear, σ=0.7	7.2	7.1	7.1	7.1
Reconstruction gain (%), iTransformer, σ=0.7	20.0	21.9	26.3	26.7

SOTA models on specific datasets, they cannot generate good performance on the other datasets. $\mathrm{RAP}_{\mathrm{RA}}$ maintains the lowest or near-lowest noisy MSE on most datasets and horizons, with the clearest margins on Traffic and PEMS. Moreover, we report other noise-type cases, including spike and quantization, in Table 6, 7 of Appendix D.4. These additional results show that the robustness trend also appears under spike and quantization corruptions. This indicates that RAP is not only effective in clean scenarios but also robust to uncertainty and noise in practical forecasting environments. We also provide exact MSE/MAE results in Table 10 of Appendix D.

When comparing RAP variants, the performance difference between $\mathrm{RAP}_{\mathrm{PO}}$ and $\mathrm{RAP}_{\mathrm{RA}}$ is relatively small on simpler datasets, while on more complex datasets such as Solar or Traffic, the gap becomes noticeably larger. Nevertheless, even the simpler RAP prediction-only variant surpasses other state-of-the-art models under noisy conditions, suggesting that the architecture of RAP itself inherently provides robustness to perturbations.

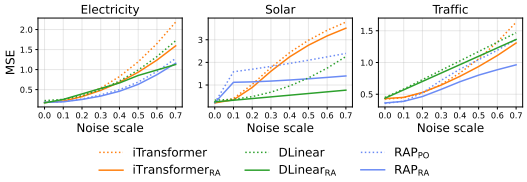


Figure 4: Average MSE vs. noise scale on RAP, iTransformer, and DLinear. Results are averaged over prediction lengths $H = \{96, 192, 336, 720\}$. Note that log-normalization is applied to iTransformer to accommodate its large error magnitude.

7.4 Effect of context length

The experiments above follow the standard protocol with context $C = 96$. We additionally test whether context length C changes (i) the ranking between backbones on clean data and (ii) the robustness gain from reconstruction. We train DLinear and iTransformer, with and without the reconstruction term, on ECL and Traffic with $C \in \{96, 192, 336, 720\}$. We corrupt the test-time history with Gaussian noise $\sigma \in [0, 0.7]$. Full results (ECL and Traffic, all σ) are in Appendix D.9. As seen in Table 2, on Traffic, DLinear and iTransformer swap ranks as C grows: at $C = 96$, iTransformer has lower clean MSE than DLinear (0.422 vs. 0.450); at $C = 720$, DLinear has lower clean MSE than iTransformer (0.283 vs. 0.379). For iTransformer, the relative MSE reduction from adding reconstruction at $\sigma = 0.7$ increases with C: 20.0% at $C = 96$, 21.9% at $C = 192$, 26.3% at $C = 336$, and 26.7% at $C = 720$. For DLinear, the corresponding reduction stays near 7.1% across all four values of C.

8 Conclusion

We introduced *reconstruction-augmented predictors* (RAP) for robust long-term time-series forecasting. Our analysis explains a mechanism: in the small-noise regime, under explicit assumptions, reconstruction-augmented full-sequence training controls the predictor's Jacobian sensitivity relative to that of the Bayes forecasting map, and direct measurement confirms that the objective lowers it. Empirically, RAP remains competitive on clean benchmarks, degrades more gracefully under Gaussian, spike and quantization corruption than both standard forecasters and robustness-oriented objectives on a shared backbone, and benefits more from longer context. Note that we fixed the backbone as a DiffWave-inspired dual-axis architecture to hold the computational budget constant across $\mathrm{RAP}_{\mathrm{PO}}$ and $\mathrm{RAP}_{\mathrm{RA}}$, isolating the effect of the reconstruction objective. Finding the optimal architecture for this objective is left for future work.

AI USE STATEMENT

We used generative AI tools solely to refine the grammar and phrasing of the manuscript. The authors reviewed all AI assisted revisions to ensure that the technical content and arguments remained unchanged.

REPRODUCIBILITY STATEMENT

We provide proofs for all theoretical results (Theorem 4.2, 4.3, Proposition 2.1, 2.2) and the details regarding regularity assumptions in Appendix H. Dataset details and baseline settings are described in Appendix B, and the training and inference stages, including hyperparameters, are outlined in Appendix F. The construction of the synthetic toy dataset is detailed in Appendix A.1, and information regarding seed variability can be found in Appendix D.7.

REFERENCES

Taha Aksu, Gerald Woo, Juncheng Liu, Xu Liu, Chenghao Liu, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. Gift-eval: A benchmark for general time series forecasting model evaluation. *arXiv preprint arXiv:2410.10393*, 2024.

Abdul Fatir Ansari, Lorenzo Stella, Ali Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Bernie Wang. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=gerNCVqqtR. Expert Certification.

Chao Chen, Karl Petty, Alexander Skabardonis, Pravin Varaiya, and Zhanfeng Jia. Freeway performance measurement system: mining loop detector data. *Transportation research record*, 1748, 2001.

Jean-Yves Franceschi, Aymeric Dieuleveut, and Martin Jaggi. Unsupervised scalable representation learning for multivariate time series. *Advances in neural information processing systems*, 32, 2019.

Judy Hoffman, Daniel A. Roberts, and Sho Yaida. Robust learning with jacobian regularization, 2019. URL https://arxiv.org/abs/1908.02729.

Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6:695–709, 2005.

Shruti Kaushik, Abhinav Choudhury, Pankaj Kumar Sheron, Nataraj Dasgupta, Sayee Natarajan, Larry A Pickett, and Varun Dutt. Ai in healthcare: time-series forecasting using statistical, neural, and ensemble architectures. *Frontiers in big data*, 3:4, 2020.

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*, 2021.

Marcel Kollovieh, Abdul Fatir Ansari, Michael Bohlke-Schneider, Jasper Zschiegner, Hao Wang, and Yuyang Bernie Wang. Predict, refine, synthesize: Self-guiding diffusion models for probabilistic time series forecasting. *Advances in Neural Information Processing Systems*, 2023.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*, 2020.

Dilfira Kudrat, Zongxia Xie, Yanru Sun, Tianyu Jia, and Qinghua Hu. Patch-wise structural loss for time series forecasting. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning*

Research, pp. 31841–31859. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/kudrat25a.html.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, 2018.

Pedro Lara-Benítez, Manuel Carranza-García, and José C Riquelme. An experimental review on deep learning architectures for time series forecasting. *International journal of neural systems*, 31(03):2130001, 2021.

Zhe Li, Shiyi Qi, Yiduo Li, and Zenglin Xu. Revisiting long-term time series forecasting: An investigation on linear mapping. *arXiv preprint arXiv:2305.10721*, 2023.

Bryan Lim and Stefan Zohren. Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A*, 379(2194):20200209, 2021.

Minhao Liu, Ailing Zeng, Muxi Chen, Zhijian Xu, Qiuxia Lai, Lingna Ma, and Qiang Xu. Scinet: Time series modeling and forecasting with sample convolution and interaction. *Advances in Neural Information Processing Systems*, 2022a.

Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X Liu, and Schahram Dustdar. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International Conference on Learning Representations*, 2022b.

Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. *Advances in neural information processing systems*, 35: 9881–9893, 2022c.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024.

Mohammad Amin Morid, Olivia R Liu Sheng, and Joseph Dunbar. Time series prediction using deep learning methods in healthcare. *ACM Transactions on Management Information Systems*, 2023.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*, 2023.

Adam Paszke, Sam Gross, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Xihao Piao, Zheng Chen, Taichi Murayama, Yasuko Matsubara, and Yasushi Sakurai. Fredformer: Frequency debiased transformer for time series forecasting. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, 2024.

Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *International conference on machine learning*. PMLR, 2021.

Omer Berat Sezer, Mehmet Ugur Gudelek, and Ahmet Murat Ozbayoglu. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied soft computing*, 90:106181, 2020.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

Gian Antonio Susto, Angelo Cenedese, and Matteo Terzi. Time-series classification methods: Review and applications to power systems data. *Big data application in power systems*, 2018.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in neural information processing systems*, 2021.

Sana Tonekaboni, Danny Eytan, and Anna Goldenberg. Unsupervised representation learning for time series with temporal neighborhood coding. *arXiv preprint arXiv:2106.00750*, 2021.

José F Torres, Dalil Hadjout, Abderrazak Sebaa, Francisco Martínez-Álvarez, and Alicia Troncoso. Deep learning for time series forecasting: a survey. *Big data*, 9(1):3–21, 2021.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Hao Wang, Licheng Pan, Zhichao Chen, Xu Chen, Qingyang Dai, Lei Wang, Haoxuan Li, and Zhouchen Lin. Time-o1: Time-series forecasting needs transformed label alignment. *Advances in Neural Information Processing Systems*, 38:8660–8690, 2026.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Guo Qin, Haoran Zhang, Yong Liu, Yunzhong Qiu, Jianmin Wang, and Mingsheng Long. Timexer: Empowering transformers for time series forecasting with exogenous variables. *Advances in Neural Information Processing Systems*, 37:469–498, 2024.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 53140–53164. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/woo24a.html.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34:22419–22430, 2021.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *International Conference on Learning Representations*, 2023.

Ling Yang and Shenda Hong. Unsupervised time-series representation learning with iterative bilinear temporal-spectral fusion. In *International conference on machine learning*. PMLR, 2022.

Yuxuan Yang, Dalin Zhang, Yuxuan Liang, Hua Lu, Gang Chen, and Huan Li. Not all data are good labels: On the self-supervised labeling for time series forecasting, 2025.

Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. Ts2vec: Towards universal representation of time series. In *Proceedings of the AAAI conference on artificial intelligence*, 2022.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, 2023.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The eleventh international conference on learning representations*, 2023.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International Conference on Machine Learning*, pp. 27268–27286. PMLR, 2022.

A EXPERIMENTS ON TOY DATASET

A.1 TOY EXPERIMENTAL SETUP

We generate each time series as a sum of components with different spectral characteristics:

$$x(t) = \underbrace{\mathrm{HF}(t)}_{\text{high-frequency}} + \underbrace{\mathrm{LF}(t)}_{\text{low-frequency}} + \underbrace{\mathrm{Trend}(t)}_{\text{random walk + drift}} + \underbrace{\xi(t)}_{\text{noise}}, \qquad y(t) = \mathrm{HF}(t). \tag{7}$$

The model receives a history window formed from $x(t)$ and is trained to predict the future values of the HF component $y(t)$. This setup ensures that removing or distorting high-frequency content directly destroys the information needed for forecasting.

According to equation 7, the HF term is the prediction target and is constructed as a high-frequency signal. We sample $\omega \sim \mathcal{U}[0.6\pi, 0.9\pi]$, $A_0 \sim \mathcal{U}[a_{\mathrm{hf}}^{\min}, a_{\mathrm{hf}}^{\max}]$, and $\phi_0 \sim \mathcal{U}[0, 2\pi]$. We introduce non-stationarity via i.i.d. amplitude and phase increments $\Delta A(i) \sim \mathcal{N}(0, \sigma_A^2)$ and $\Delta\phi(i) \sim \mathcal{N}(0, \sigma_\phi^2)$, defining random walks

$$\widetilde{A}(t) = A_0 + \sum_{i=1}^{t} \Delta A(i), \qquad \Phi(t) = \sum_{i=1}^{t} \Delta\phi(i).$$

We then clip amplitude $A(t) = \min(\max(\widetilde{A}(t), a_{\mathrm{hf}}^{\min}), a_{\mathrm{hf}}^{\max})$ to keep a valid range and define

$$\mathrm{HF}(t) = A(t)\sin\big(\omega t + \phi_0 + \Phi(t)\big).$$

The LF term serves as a low-frequency signal that interferes with recovering $\mathrm{HF}(t)$. We sample $K \sim \mathrm{Unif}\{k_{\mathrm{lf}}^{\min}, \ldots, k_{\mathrm{lf}}^{\max}\}$ and, for each $k = 1, \ldots, K$, sample $a_k \sim \mathcal{U}[a_{\mathrm{lf}}^{\min}, a_{\mathrm{lf}}^{\max}]$, $\omega_k \sim \mathcal{U}[0.05\pi, 0.2\pi]$, and $\varphi_k \sim \mathcal{U}[0, 2\pi]$, then define

$$\mathrm{LF}(t) = \sum_{k=1}^{K} a_k \sin\big(\omega_k t + \varphi_k\big).$$

The Trend term consists of a linear drift and a random walk process. Let $\eta(i) \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1)$ and define $R(t) = \sum_{i=0}^{t} \eta(i)$ with mean $\bar{R} = \frac{1}{N}\sum_{t=0}^{N-1} R(t)$. We sample a drift coefficient β and define

$$\mathrm{Trend}(t) = \beta t + \gamma\big(R(t) - \bar{R}\big). \tag{8}$$

Finally, we define measurement noise $\xi(t) \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_{\mathrm{meas}}^2)$. In our experiments, we set $a_{\mathrm{hf}}^{\min} = 0.5$, $a_{\mathrm{hf}}^{\max} = 1.5$, $\sigma_A = 0.02$, $\sigma_\phi = 0.05$, $k_{\mathrm{lf}}^{\min} = 1$, $k_{\mathrm{lf}}^{\max} = 3$, $a_{\mathrm{lf}}^{\min} = 0.1$, $a_{\mathrm{lf}}^{\max} = 0.5$, and $\gamma = 0.02$. We sample $\beta \sim \mathcal{U}[-0.001, 0.001]$ and use $\xi(t) \sim \mathcal{N}(0, \sigma_{\mathrm{meas}}^2)$ with $\sigma_{\mathrm{meas}} = 0.05$. The total dataset consists of $12,000$ time steps and is split chronologically into train (70%), validation (20%), and test (10%) sets.

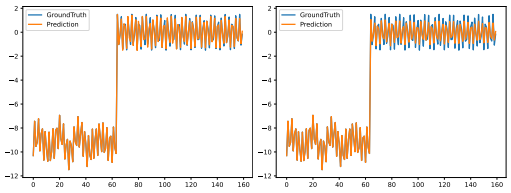
The forecasting model, DLinear, was configured with a look-back window of 64 and a prediction horizon of 96. Training was performed using the Adam optimizer with a learning rate of 10^{-3} and a batch size of 256 for 20 epochs. All evaluations were conducted on the clean test set to assess the model's ability to recover the original signal from corrupted inputs.

For Gaussian noise injection, we perturb the input during training as $\tilde{x}(t) = x(t) + z(t)$, where $z(t) \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_{\mathrm{abs}}^2)$. To ensure the noise magnitude is proportional to the original signal, we set $\sigma_{\mathrm{abs}} = \sigma_{\mathrm{rel}}\,\sigma_{\mathrm{ref}}$ with $\sigma_{\mathrm{ref}} = \mathrm{std}(x_{\mathrm{train}})$. In our experiments, we varied the relative noise σ_{rel} in the range $[0.0, 1.0]$ to observe the forecasting performance (see Figure 5a).

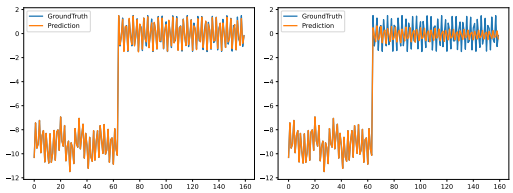
For low-pass filtering, we apply frequency masking in the Real Fast Fourier Transform (rFFT) domain during training as $\tilde{x} = \mathcal{F}^{-1}\big(M(\rho) \odot \mathcal{F}(x)\big)$, where $\mathcal{F}$ denotes rFFT,

Table 3: Detailed dataset descriptions.

Dataset	Dim	Dataset Size	Frequency
Solar	137	(36792,5271,10540)	10 min
Elec.	321	(18317,2633,5261)	Hourly
Traffic	862	(12185,1757,3509)	Hourly
PEMS03	358	(15617,5135,5135)	5 min
PEMS04	307	(10172,3375,3375)	5 min
PEMS07	883	(16911,5622,5622)	5 min



(a) Gaussian noise perturbation: Comparisons between clean input ($\sigma = 0.0$, left) and noisy input ($\sigma = 1.0$, right) demonstrate that the model misses the high-frequency target as noise increases.



(b) Low-pass filtering: Comparisons between Low-pass Ratios of 0.268 (left) and 0.272 (right) also show that filtering out the critical frequency band prevents accurate forecasting.

Figure 5: Visualization of prediction results under robustness interventions.

$\odot$ is element-wise multiplication, and $M(\rho) \in \{0,1\}^K$ keeps the lowest $\lfloor \rho K \rfloor$ frequency bins. We parameterize the keep ratio as $\rho = \rho_{\max} - s(\rho_{\max} - \rho_{\min})$ and define the Low-pass Ratio as $1 - \rho$. In our experiments, we set $(\rho_{\max}, \rho_{\min}) = (1.0, 0.125)$, and swept s in the range $[0.306, 0.311]$. According to the equation above, this corresponds to a Low-pass Ratio range of approximately 0.268 to 0.272, a critical regime where we observe a sharp error increase (see Figure 5b).

A.2 Experimental Results

To empirically validate the failure modes suggested by the above propositions, we construct a controlled toy forecasting task in which the *predictive target lies entirely in the high-frequency (HF) regime*. We train a representative forecaster, DLinear (Zeng et al., 2023), and report MSE/MAE.

Robustness interventions. We evaluate the above two commonly used robustness practices. For Gaussian-noise training, we inject Gaussian perturbations into the history window during training. For low-pass conditioning, the forecaster observes the temporally filtered history under the corresponding low-pass protocol. We sweep the noise standard deviation and the low-pass strength, and measure how forecasting error changes as these interventions become stronger. We sweep the noise standard deviation and the low-pass strength (cutoff/ratio), and measure how forecasting error changes as these interventions become stronger.

As shown in Figure 6, both Gaussian-noise training and low-pass conditioning increase MSE/MAE as their intervention strength grows. This is consistent with our analysis: when the predictive target lies in the high-frequency regime, generic robustness techniques can deteriorate predictive microstructure in the original history data. The toy experiment isolates this mechanism by making the HF component the forecasting target. Finally, to complement the toy study with real data, we report an objective-level ablation on DLinear comparing Gaussian-noise training, low-pass filtering, an explicit Jacobian penalty, and our reconstruction-augmented objective in Appendix D.1.

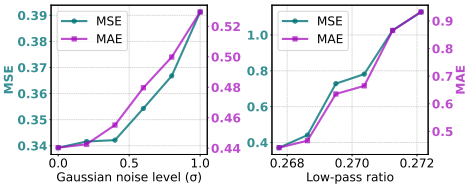


Figure 6: Toy dataset results (DLinear). MSE/MAE as Gaussian perturbation strength (left) and low-pass filtering strength (right) increase.

High-frequency structure. We figure out whether reconstruction remove useful high-frequency structure. We train two DLinear models with the low-pass training that breaks the forecast in Figures 5–6 ($\rho \approx 0.731$), with and without $\mathcal{L}_{\text{rec}}$, and add Gaussian noise only at test time. The reconstruction-augmented model is better at every noise level (Table 5). The key result is at $\sigma = 0$: it halves the clean error (4.031 vs. 2.074), so the reconstruction term recovers predictive structure removed by low-pass training, and this gap stays roughly constant across noise levels. This matches Theorem 4.3, which pulls J_{pred} toward J_m, not toward zero.

Robustness-oriented objectives. On the same DLinear backbone, changing only the loss, reconstruction gives the lowest MSE at every noise level on Solar (Table 4). Gaussian-noise training

becomes worse than no intervention from $\sigma = 0.2$ on (Proposition 2.1), and low-pass filtering is the least accurate on clean data (Proposition 2.2); the explicit Jacobian penalty improves less and is $1.6\times$ slower to train (Appendix D.1). A finer noise grid (Appendix D.8) shows that the apparent knees near $\sigma = 0.1$ in Figure 3 reflect the crossing (Traffic) or early separation (Solar) of two smooth curves, consistent with $\mathcal{R}_\sigma \approx \mathcal{R}_0 + \sigma^2 S$.

$\mathrm{RAP}_{\mathrm{PO}}$ and $\mathrm{RAP}_{\mathrm{RA}}$ share the same architecture, parameters, and inference costs, differing only in whether $\mathcal{L}_{\mathrm{rec}}$ is activated during training. This enables a controlled comparison without confounding variables arising from differences in model capacity. To validate this objective function across architectures, we applied it to iTransformer and DLinear on the Electricity, Solar, and Traffic datasets, keeping all hyperparameters fixed and extending the output to the full sequence. Performance on noisy inputs improved across the Solar and Traffic datasets, as well as at high noise levels on the Electricity dataset (Figure 4).

Table 4: Objective-level ablation on DLinear under test-time Gaussian input noise (Solar). Results are averaged over prediction lengths $H = \{96, 192, 336, 720\}$.

Dataset	Model	0.0		0.1		0.2		0.3		0.4		0.5		0.6		0.7	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Solar	DLinear	0.246	0.279	0.394	0.418	1.177	0.780	2.878	1.273	5.583	1.800	9.245	2.334	13.815	2.867	19.261	3.398
	$\mathrm{DLinear}_{\mathrm{GN}}$	0.270	0.289	0.371	0.372	1.768	0.876	4.990	1.587	10.022	2.344	16.664	3.116	24.825	3.894	34.474	4.675
	$\mathrm{DLinear}_{\mathrm{LPF}}$	0.296	0.314	0.471	0.453	1.307	0.795	2.754	1.195	4.785	1.606	7.384	2.016	10.541	2.424	14.256	2.831
	$\mathrm{DLinear}_{\mathrm{JacReg}}$	0.245	0.278	0.373	0.410	1.060	0.749	2.519	1.204	4.795	1.686	7.849	2.174	11.644	2.659	16.158	3.143
	DLinear$_{\mathrm{RA}}$	0.244	0.276	0.351	0.401	0.943	0.717	2.159	1.134	4.006	1.572	6.452	2.013	9.472	2.451	13.055	2.887

B Main experimental setup

We conduct extensive experiments on six real-world multivariate time series datasets, including Solar-energy (Lai et al., 2018), Electricity (ECL), Traffic, (Wu et al., 2021), and Performance Measurement System (PEMS) 3, 4, 7 (Chen et al., 2001) introduced in (Liu et al., 2022a). We focus on datasets that have a high number of channels and possess more complex temporal features for evaluating the robustness of the proposed approach. Table 3 summarizes details of statistics of the datasets.

- Solar-energy[5]: contains the solar power production records from 137 PV plants in 2006.
- Electricity (ECL)[6]: describes the hourly electricity consumption data of 321 clients from 2012 to 2014.
- Traffic[7]: records hourly road occupancy rates measured by 862 sensors on San Francisco Bay area freeways from January 2015 to December 2016.
- Performance Measurement System (PEMS)[8] contains the public traffic network data in California collected by 5-minute windows.

For baselines, we select recent LTSF models [9], namely Autoformer (Wu et al., 2021), DLinear (Zeng et al., 2023), TimesNet (Wu et al., 2023), Crossformer (Zhang & Yan, 2023), PatchTST (Nie et al., 2023), RLinear (Li et al., 2023), iTransformer (Liu et al., 2024), and TimeXer (Wang et al., 2024). We describe the details in Section 6.

C Extended related work

In addition to the architectures discussed in Section 6, time-series foundation models such as TimesFM (das), Chronos (Ansari et al., 2024), and Moirai (Woo et al., 2024) are evaluated on broader benchmarks like GIFT-Eval (Aksu et al., 2024). They enhance prediction accuracy on clean

[5] https://www.nrel.gov/grid/solar-power-data.html

[6] https://archive.ics.uci.edu/dataset/321/electricityloaddiagrams20112014

[7] https://pems.dot.ca.gov

[8] https://pems.dot.ca.gov

[9] Our baseline selection stems from the following github repository : https://github.com/thuml/Time-Series-Library

Table 5: MSE/MAE of PO (prediction-only) and RA (reconstruction-augmented) on the toy dataset under test-time Gaussian noise. Models are trained with LPF augmentation (keep ratio $\rho \approx 0.731$, i.e., $s = 0.307$). Lower is better; best results in **bold**.

Noise σ	0.0		0.1		0.2		0.3		0.4		0.5		0.6		0.7	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
PO	4.031	1.912	4.034	1.909	4.110	1.912	4.411	1.947	5.006	2.023	5.869	2.135	6.928	2.275	8.136	2.444
RA	**2.074**	**1.330**	**2.103**	**1.327**	**2.245**	**1.324**	**2.570**	**1.337**	**3.099**	**1.396**	**3.801**	**1.529**	**4.640**	**1.716**	**5.697**	**1.941**

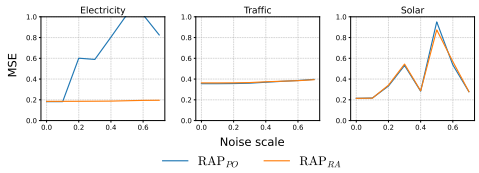


Figure 7: Robustness to Quantization Noise.

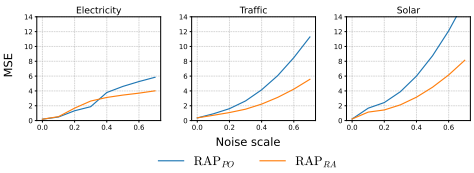


Figure 8: Robustness to Spike Noise.

Table 6: Detailed results of Quantization Robustness.

Dataset	Model	0.0		0.1		0.2		0.3		0.4		0.5		0.6		0.7	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Electricity	**RAP**$_{\text{PO}}$	0.182	0.281	0.182	0.281	0.601	0.556	0.589	0.546	0.804	0.689	1.023	0.834	1.020	0.833	0.825	0.706
	RAP$_{\text{RA}}$	0.185	0.283	0.185	0.283	0.186	0.284	0.187	0.285	0.188	0.287	0.191	0.292	0.195	0.296	0.196	0.299
Solar	**RAP**$_{\text{PO}}$	0.214	0.215	0.217	0.219	0.333	0.378	0.531	0.460	0.283	0.360	0.951	0.669	0.537	0.476	0.278	0.256
	RAP$_{\text{RA}}$	0.214	0.214	0.216	0.218	0.341	0.378	0.545	0.482	0.286	0.358	0.874	0.683	0.569	0.479	0.278	0.255
Traffic	**RAP**$_{\text{PO}}$	0.355	0.352	0.355	0.352	0.357	0.353	0.360	0.356	0.369	0.362	0.378	0.369	0.386	0.376	0.397	0.385
	RAP$_{\text{RA}}$	0.363	0.364	0.363	0.364	0.364	0.365	0.366	0.367	0.374	0.372	0.379	0.377	0.384	0.383	0.394	0.390

Table 7: Detailed results of Spike Noise Robustness.

Dataset	Model	0.0		0.1		0.2		0.3		0.4		0.5		0.6		0.7	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Electricity	**RAP**$_{\text{PO}}$	0.182	0.281	0.481	0.526	1.321	0.927	1.873	1.079	3.783	1.613	4.608	1.786	5.264	1.907	5.842	1.996
	RAP$_{\text{RA}}$	0.185	0.283	0.526	0.554	1.676	1.037	2.634	1.301	3.119	1.399	3.439	1.447	3.699	1.477	4.013	1.506
Solar	**RAP**$_{\text{PO}}$	0.214	0.215	1.672	1.036	2.430	1.377	3.896	1.825	6.004	2.305	8.747	2.801	12.109	3.305	16.094	3.818
	RAP$_{\text{RA}}$	0.214	0.214	1.143	0.813	1.433	0.976	2.122	1.227	3.156	1.511	4.501	1.808	6.155	2.114	8.134	2.427
Traffic	**RAP**$_{\text{PO}}$	0.355	0.352	0.913	0.666	1.605	0.937	2.648	1.206	4.150	1.504	6.066	1.815	8.489	2.137	11.280	2.463
	RAP$_{\text{RA}}$	0.363	0.364	0.713	0.588	1.081	0.754	1.535	0.908	2.222	1.093	3.129	1.294	4.239	1.502	5.566	1.718

data through inductive bias, tokenization, and scale. Since these models generally do not directly address the issue of sensitivity to corruption in the observed history, RAP—a learning objective designed to be integrated with the backbone—serves to complement them.

Meanwhile, related studies addressing robustness can be categorized based on *where* they intervene. (1) *Data-side* methods modify the model's training data. These include techniques such as Gaussian input perturbation or spectral filtering applied to probabilistic and diffusion-based forecasting models. The former replaces the target with a heat-smoothed prediction (Proposition 2.1), while the latter incurs an irreducible penalty due to the removal of high-frequency components that are useful for forecasting (Proposition 2.2). Both effects are observed in the experiments using real-world data presented in Table 4. (2) *Target-side* methods act under clean inputs. For instance, SCAM (Yang et al., 2025) corrects corrupted prediction targets, patch-wise (Kudrat et al., 2025) structural losses align the structures of predictions and targets, and Time-o1 (Wang et al., 2026) decorrelates label sequences. Since these methods do not directly control the sensitivity to corrupted histories defined in Theorem 4.2, they can be used in conjunction with RAP. Unlike SCAM, we assume that historical data is clean during training and corruption occurs only at deployment. Appendix G demonstrates that extending SCAM's reconstruction approach to historical data still makes it hard to bridge this performance gap. (3) *Map-side* methods such as Jacobian regularization (Hoffman et al., 2019) penalize input–output sensitivity explicitly, at the cost of extra backward passes and with no reference point. RAP also acts on the map, but indirectly, without a test-time stage, and relative to the Bayes map rather than toward zero.

D ADDITIONAL EXPERIMENTAL RESULTS

D.1 OBJECTIVE-LEVEL ABLATION ON DLINEAR

As discussed in Section 2, common robustness strategies are Gaussian-noise training, low-pass filtering, and explicit Jacobian regularization. We compare five objectives on the same DLinear backbone: prediction-only (PO), Gaussian-noise injection (GN), low-pass filtering with keep ratio 8% (LPF), an explicit Jacobian penalty (JACREG), and ours (RA). All variants share hyperparameters and output length $T+H$; only the loss changes. We use Solar, which has the richest high-frequency structure, and the protocol of Section 7.3. In Table 4, RA has the lowest MSE at every noise level and the best clean result. LPF is the worst on clean data and beats RA only in MAE at $\sigma \geq 0.6$ (Proposition 2.2); GN is worse than PO from $\sigma = 0.2$ on (Proposition 2.1); JACREG improves on PO less than RA and is slower to train (42.1 vs. 26.6 ms/iter).

D.2 ABLATION STUDY

Table 8 presents a grid search over encoder depth (Layers {1,2,3}) and channel width (Channels {8,16,32,64}) on the Electricity and Solar datasets, reporting MSE (lower is better).

Table 8: Ablation on depth (layers) and width (channels). Metric is MSE↓.

Dataset	Layers	Channels			
		8	16	32	64
Electricity	1	1.145	1.126	1.112	1.103
	2	0.217	0.188	0.163	**0.149**
	3	0.217	0.177	0.159	<u>0.155</u>
Solar	1	0.239	0.186	0.176	0.171
	2	0.161	0.145	0.132	<u>0.125</u>
	3	0.159	0.136	0.127	**0.122**

(i) At a fixed depth, increasing width yields an almost monotonic reduction in error on both datasets (e.g., for Layers=1 on Electricity, 1.145 → 1.103 as Channels 8 → 64; on Solar, 0.239 → 0.171). (ii) Increasing depth from 1 to 2 provides a substantial gain, while moving from 2 to 3 shows diminishing returns; on Electricity, depth 3 is slightly better than depth 2 at the same width, suggesting mild overfitting.

Overall, once a moderate depth and width are reached, further parameter growth brings limited benefit. This highlights a clear performance–efficiency trade-off with respect to model size (parameters and compute).

D.3 COMPLEXITY ANALYSIS

We benchmark on the TRAFFIC dataset (862 variates) with prediction length = 96. Figure 9 plots training time (x), MSE (y), and memory footprint (bubble size).

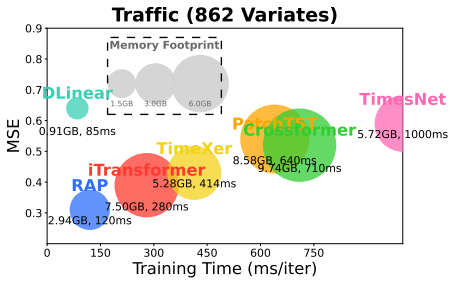


Figure 9: Model efficiency comparison under Traffic dataset. Because of capacity, we report TimesNet's training time as 1000 ms/iter, which takes 2004 ms/iter.

Our model **RAP** attains the best accuracy while using markedly less memory. Concretely, RAP reaches MSE ≈ 0.31 at 120ms/iter with a footprint of 2.94GB, whereas heavier Transformer variants—e.g., iTransformer (0.39, 280ms, 7.5GB), PatchTST (0.54, 640ms, 8.58GB), and Crossformer (0.52, 710ms, 9.74GB). A light linear baseline (DLinear) is fast (85ms, 0.91GB) but much less accurate (0.64).

Why RAP is efficient. During training, RAP performs *feature subsampling*: at each step, we activate only a subset of channels while preserving the temporal receptive field. This reduces both activation and attention costs roughly linearly in the sampled-channel ratio, yielding an efficient, smaller training footprint without sacrificing representation quality. In inference, the full feature set is used, so accuracy benefits from all channels while the training-time savings remain. Overall, RAP offers a favorable *accuracy–efficiency* trade-off on multivariate long-horizon forecasting.

Table 9: Detailed results of Figure 4. Results are averaged over prediction lengths $H = \{96, 192, 336, 720\}$.

Dataset	Model	0.0		0.1		0.2		0.3		0.4		0.5		0.6		0.7	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Electricity	DLinear	0.225	0.319	0.256	0.354	0.347	0.430	0.500	0.525	0.714	0.631	0.988	0.744	1.324	0.860	1.721	0.979
	$\text{DLinear}_{\text{RA}}$	0.168	0.266	0.305	0.412	0.442	0.508	0.579	0.587	0.716	0.655	0.853	0.717	0.990	0.774	1.127	0.826
	iTransformer	0.180	0.270	0.214	0.317	0.335	0.429	0.540	0.563	0.825	0.704	1.194	0.851	1.648	1.001	2.186	1.153
	$\text{iTransformer}_{\text{RA}}$	0.176	0.266	0.214	0.316	0.325	0.419	0.482	0.528	0.684	0.638	0.935	0.752	1.237	0.868	1.593	0.986
	$\textbf{RAP}_{\text{PO}}$	0.185	0.283	0.200	0.302	0.265	0.367	0.367	0.450	0.499	0.538	0.688	0.645	0.954	0.771	1.277	0.902
	$\textbf{RAP}_{\text{RA}}$	0.182	0.281	0.195	0.299	0.254	0.358	0.338	0.427	0.459	0.513	0.631	0.616	0.872	0.736	1.159	0.856
Solar	DLinear	0.330	0.401	0.369	0.434	0.487	0.508	0.684	0.604	0.959	0.714	1.313	0.831	1.746	0.953	2.257	1.078
	$\text{DLinear}_{\text{RA}}$	0.253	0.315	0.327	0.393	0.401	0.445	0.476	0.488	0.550	0.527	0.624	0.562	0.698	0.594	0.772	0.624
	iTransformer	0.239	0.262	0.500	0.429	1.792	0.867	4.868	1.526	10.400	2.328	18.631	3.191	29.488	4.074	42.827	4.959
	$\text{iTransformer}_{\text{RA}}$	0.233	0.258	0.413	0.411	1.440	0.841	3.997	1.490	8.447	2.225	14.760	2.976	22.803	3.720	32.477	4.455
	$\textbf{RAP}_{\text{PO}}$	0.214	0.215	1.585	0.974	1.698	1.050	1.821	1.118	1.955	1.182	2.096	1.245	2.244	1.306	2.396	1.364
	$\textbf{RAP}_{\text{RA}}$	0.214	0.214	1.120	0.787	1.139	0.815	1.174	0.843	1.221	0.873	1.275	0.902	1.336	0.932	1.400	0.962
Traffic	DLinear	0.450	0.326	0.582	0.467	0.730	0.559	0.878	0.651	1.025	0.719	1.173	0.786	1.320	0.843	1.468	0.899
	$\text{DLinear}_{\text{RA}}$	0.437	0.301	0.569	0.457	0.702	0.552	0.834	0.628	0.966	0.695	1.098	0.755	1.230	0.810	1.363	0.860
	iTransformer	0.422	0.282	0.445	0.319	0.522	0.398	0.654	0.495	0.834	0.598	1.057	0.704	1.323	0.811	1.633	0.919
	$\text{iTransformer}_{\text{RA}}$	0.431	0.289	0.451	0.320	0.528	0.392	0.644	0.474	0.782	0.557	0.936	0.638	1.110	0.718	1.307	0.798
	$\textbf{RAP}_{\text{PO}}$	0.355	0.352	0.390	0.380	0.523	0.465	0.719	0.569	0.895	0.657	1.044	0.728	1.193	0.793	1.364	0.858
	$\textbf{RAP}_{\text{RA}}$	0.363	0.364	0.388	0.387	0.464	0.446	0.577	0.516	0.697	0.579	0.801	0.630	0.886	0.670	0.963	0.705

D.4 Full results of noise robustness experiments

In addition to the Gaussian input-noise analysis, we further evaluate RAP under two distinct corruption types: **Quantization Noise** and **Spike Noise**. Quantization noise simulates low-precision transmission or compression artifacts, while spike noise mimics sudden sensor anomalies or outliers. The spike-noise results show clear gains from reconstruction, and the quantization results show dataset-dependent behavior, with strong gains on Electricity and comparable behavior on Solar and Traffic.

Figure 7 and Figure 8 illustrate the MSE trends of RAP_{PO} and RAP_{RA} under varying noise scales, averaged over prediction horizons $H = \{96, 192, 336, 720\}$. Detailed numerical results are provided in the tables below the figures. We use the following noise injection protocol:

- **Gaussian noise.** For a noise scale $\sigma > 0$, we corrupt the selected time indices by $\tilde{\mathbf{x}}_t^{\text{hist}} = \mathbf{x}_t^{\text{hist}} + \sigma\,\boldsymbol{\epsilon}_t, \;\; \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
- **Spike noise.** To model impulsive perturbations, we add Laplace noise: $\tilde{\mathbf{x}}_t^{\text{hist}} = \mathbf{x}_t^{\text{hist}} + \boldsymbol{\xi}_t, \;\; \boldsymbol{\xi}_t \sim \text{Laplace}(\mathbf{0}, \sigma\mathbf{I})$. where $\sigma > 0$ is the spike scale.
- **Quantization.** For a quantization step $\sigma > 0$, we quantize by rounding to the nearest multiple of δ: $\tilde{\mathbf{x}}_t^{\text{hist}} = \sigma \cdot \text{round}\big(\mathbf{x}_t^{\text{hist}}/\sigma\big)$.

Table 10: MSE / MAE vs noise scale. Rows: Model; Columns: Noise scales. We report averaged results across prediction length $\{12, 24, 48, 96\}$ on PEMS and $\{96, 192, 336, 720\}$ on the others.

Dataset	Model	0.0		0.1		0.2		0.3		0.4		0.5		0.6		0.7	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
PEMS03	DLinear	0.264	0.358	0.376	0.463	0.713	0.644	1.274	0.848	2.060	1.064	3.070	1.286	4.305	1.512	5.765	1.740
	RLinear	0.273	0.369	0.356	0.451	0.608	0.594	1.027	0.756	1.615	0.930	2.370	1.109	3.293	1.293	4.384	1.480
	PatchTST	0.194	0.283	2.176	0.936	7.003	1.769	13.863	2.529	23.097	3.285	34.818	4.044	49.077	4.806	65.891	5.571
	iTransformer	0.149	0.255	0.653	0.590	2.516	1.169	4.729	1.630	7.215	2.038	10.140	2.434	13.585	2.831	17.586	3.231
	TimeXer	0.187	0.284	0.404	0.420	1.828	0.915	4.142	1.407	7.065	1.864	10.546	2.293	14.500	2.701	18.985	3.101
	RAP$_{\mathrm{PO}}$	0.100	0.202	0.606	0.562	0.984	0.733	1.266	0.845	1.509	0.932	1.737	1.006	1.951	1.070	2.155	1.127
	RAP$_{\mathrm{RA}}$	0.099	0.203	0.412	0.474	0.770	0.664	1.067	0.789	1.305	0.874	1.518	0.943	1.720	1.002	1.912	1.054
PEMS04	DLinear	0.264	0.354	0.372	0.456	0.697	0.632	1.238	0.831	1.995	1.042	2.969	1.258	4.159	1.479	5.565	1.702
	RLinear	0.283	0.375	0.382	0.465	0.677	0.629	1.168	0.820	1.857	1.023	2.741	1.234	3.823	1.450	5.100	1.669
	PatchTST	0.275	0.344	0.851	0.568	1.750	0.833	2.746	1.082	3.980	1.338	5.507	1.603	7.349	1.875	9.513	2.153
	iTransformer	0.222	0.305	0.520	0.537	2.084	1.105	5.276	1.738	10.028	2.364	16.321	2.990	24.099	3.611	33.293	4.225
	TimeXer	0.109	0.225	0.204	0.300	0.935	0.629	2.847	1.105	5.890	1.610	9.867	2.101	14.889	2.592	20.949	3.082
	RAP$_{\mathrm{PO}}$	0.090	0.199	0.472	0.501	0.857	0.691	1.094	0.792	1.252	0.854	1.380	0.897	1.496	0.932	1.607	0.963
	RAP$_{\mathrm{RA}}$	0.091	0.202	0.461	0.499	0.765	0.659	0.988	0.753	1.189	0.828	1.379	0.893	1.556	0.948	1.719	0.995
PEMS07	DLinear	0.312	0.374	0.418	0.474	0.737	0.645	1.269	0.839	2.014	1.043	2.972	1.255	4.143	1.470	5.526	1.688
	RLinear	0.321	0.386	0.395	0.458	0.614	0.585	0.981	0.731	1.494	0.886	2.154	1.046	2.960	1.211	3.913	1.379
	PatchTST	0.225	0.303	0.908	0.601	3.060	1.157	6.126	1.674	10.099	2.170	15.027	2.661	20.946	3.150	27.878	3.640
	iTransformer	0.141	0.242	0.859	0.626	4.368	1.427	10.531	2.267	19.128	3.101	30.060	3.925	43.323	4.746	58.926	5.566
	TimeXer	0.103	0.210	0.230	0.314	1.640	0.760	5.349	1.314	10.319	1.809	15.992	2.245	22.180	2.642	28.780	3.010
	RAP$_{\mathrm{PO}}$	0.081	0.179	0.400	0.463	0.683	0.610	0.879	0.697	1.047	0.761	1.210	0.816	1.373	0.865	1.541	0.911
	RAP$_{\mathrm{RA}}$	0.081	0.180	0.398	0.470	0.629	0.601	0.809	0.683	0.986	0.753	1.165	0.816	1.342	0.873	1.517	0.925
ECL	DLinear	0.225	0.319	0.256	0.354	0.347	0.430	0.500	0.525	0.714	0.631	0.988	0.744	1.324	0.860	1.721	0.979
	RLinear	0.214	0.304	0.254	0.352	0.373	0.451	0.572	0.571	0.850	0.701	1.207	0.837	1.644	0.978	2.160	1.120
	PatchTST	0.204	0.294	0.235	0.332	0.339	0.428	0.495	0.535	0.686	0.639	0.911	0.741	1.174	0.844	1.478	0.947
	iTransformer	0.180	0.270	0.214	0.317	0.335	0.429	0.540	0.563	0.825	0.704	1.194	0.851	1.648	1.001	2.186	1.153
	TimeXer	0.171	0.270	0.181	0.284	0.227	0.330	0.333	0.408	0.516	0.506	0.762	0.612	1.060	0.721	1.410	0.831
	RAP$_{\mathrm{PO}}$	0.185	0.283	0.200	0.302	0.265	0.367	0.367	0.450	0.499	0.538	0.688	0.645	0.954	0.771	1.277	0.902
	RAP$_{\mathrm{RA}}$	0.182	0.281	0.195	0.299	0.254	0.358	0.338	0.427	0.459	0.513	0.631	0.616	0.872	0.736	1.159	0.856
Solar	DLinear	0.330	0.401	0.369	0.434	0.487	0.508	0.684	0.604	0.959	0.714	1.313	0.831	1.746	0.953	2.257	1.078
	RLinear	0.336	0.407	0.363	0.431	0.444	0.485	0.580	0.557	0.770	0.641	1.014	0.732	1.313	0.828	1.666	0.928
	PatchTST	0.259	0.301	0.665	0.583	2.338	1.183	5.503	1.873	10.054	2.567	15.963	3.259	23.234	3.952	31.873	4.644
	iTransformer	0.239	0.262	0.500	0.429	1.792	0.867	4.868	1.526	10.400	2.328	18.631	3.191	29.488	4.074	42.827	4.959
	TimeXer	0.238	0.286	0.517	0.424	2.004	0.889	5.088	1.566	10.240	2.341	17.459	3.146	26.557	3.952	37.404	4.751
	RAP$_{\mathrm{PO}}$	0.214	0.215	1.585	0.974	1.698	1.050	1.821	1.118	1.955	1.182	2.096	1.245	2.244	1.306	2.396	1.364
	RAP$_{\mathrm{RA}}$	0.214	0.214	1.120	0.787	1.139	0.815	1.174	0.843	1.221	0.873	1.275	0.902	1.336	0.932	1.400	0.962
Traffic	DLinear	0.673	0.419	0.699	0.449	0.778	0.514	0.909	0.595	1.093	0.685	1.329	0.781	1.618	0.881	1.960	0.984
	RLinear	0.630	0.391	0.670	0.440	0.789	0.537	0.988	0.652	1.266	0.777	1.624	0.907	2.061	1.042	2.578	1.179
	PatchTST	0.482	0.308	0.503	0.330	0.556	0.379	0.623	0.432	0.694	0.484	0.769	0.533	0.847	0.581	0.930	0.627
	iTransformer	0.422	0.282	0.445	0.319	0.522	0.398	0.654	0.495	0.834	0.598	1.057	0.704	1.323	0.811	1.633	0.919
	TimeXer	0.472	0.285	0.490	0.307	0.542	0.361	0.629	0.431	0.751	0.510	0.902	0.590	1.074	0.671	1.267	0.751
	RAP$_{\mathrm{PO}}$	0.355	0.352	0.390	0.380	0.523	0.465	0.719	0.569	0.895	0.657	1.044	0.728	1.193	0.793	1.364	0.858
	RAP$_{\mathrm{RA}}$	0.363	0.364	0.388	0.387	0.464	0.446	0.577	0.516	0.697	0.579	0.801	0.630	0.886	0.670	0.963	0.705

D.5 Sensitivity to reconstruction weight

In the main experiments, we use the default reconstruction-augmented objective with unit reconstruction weight. To check whether this choice is sensitive, we train the same RAP architecture with $\mathcal{L}_{\mathrm{pred}} + \lambda_{\mathrm{rec}}\mathcal{L}_{\mathrm{rec}}$, where $\lambda_{\mathrm{rec}} \in \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$. Here $\lambda_{\mathrm{rec}} = 0.0$ corresponds to $\mathrm{RAP}_{\mathrm{PO}}$, and $\lambda_{\mathrm{rec}} = 1.0$ corresponds to the default $\mathrm{RAP}_{\mathrm{RA}}$ used in the main paper.

Table 11: Sensitivity to reconstruction weight under Gaussian input corruption. Rows: reconstruction weight λ_{rec}; Columns: noise scales. Each entry reports MSE / MAE averaged over prediction lengths $H = \{96, 192, 336, 720\}$.

Dataset	λ_{rec}	0.0		0.2		0.4		0.6	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Solar	0.0	0.214	0.215	1.698	1.050	1.955	1.182	2.244	1.306
	0.2	0.201	0.208	0.827	0.662	0.845	0.712	0.857	0.730
	0.4	0.209	0.212	1.203	0.817	1.638	0.982	2.059	1.126
	0.6	0.208	0.212	0.762	0.673	0.723	0.679	0.730	0.692
	0.8	0.215	0.215	0.820	0.689	1.139	0.845	1.521	0.984
	1.0	0.214	0.214	1.139	0.815	1.221	0.873	1.336	0.932
Traffic	0.0	0.355	0.352	0.523	0.465	0.895	0.657	1.193	0.793
	0.2	0.365	0.361	0.491	0.458	0.751	0.593	0.921	0.664
	0.4	0.357	0.357	0.467	0.444	0.650	0.557	0.695	0.593
	0.6	0.356	0.358	0.486	0.459	0.823	0.632	1.098	0.743
	0.8	0.368	0.364	0.500	0.466	0.739	0.598	0.866	0.664
	1.0	0.363	0.364	0.464	0.446	0.697	0.579	0.886	0.670

Table 11 shows that reconstruction-augmented training is stable across a range of weights. Once input noise is introduced, the default weight $\lambda_{\mathrm{rec}} = 1.0$ improves over prediction-only training on both Solar and Traffic. Moderate weights can further improve the accuracy–robustness trade-off: the best robust performance is around $\lambda_{\mathrm{rec}} = 0.6$ on Solar and around $\lambda_{\mathrm{rec}} = 0.4$ on Traffic. We keep $\lambda_{\mathrm{rec}} = 1.0$ as the default in the main experiments to preserve the simple fixed-objective design of RAP.

Table 12: Mechanism check on held-out clean windows. Full-sequence loss $\mathbb{E}\|F(X) - G(X)\|^2$ recomputed with unit weights, and Jacobian energy $S = \mathbb{E}\|J_{\text{pred}}(X)\|_F^2$ (Hutchinson estimate, per output dimension), for prediction-only ($\lambda_{\text{rec}} = 0$) and reconstruction-augmented ($\lambda_{\text{rec}} = 1$) training.

	Full-sequence loss ↓		Measured S ↓	
Dataset	RAP_{PO}	RAP_{RA}	RAP_{PO}	RAP_{RA}
ECL	2.841	**2.120**	1.500	**1.240**
Traffic	2.308	**0.360**	3,434	**3,207**

Table 13: Seed variability of RAP_{RA} on Solar (clean setting of Table 1). Five-seed entries are mean $\pm$ standard deviation over five models retrained from scratch.

	MSE		MAE	
H	Single run (Table 1)	5 seeds	Single run (Table 1)	5 seeds
96	0.145	0.139 ± 0.005	0.168	0.164 ± 0.003
192	0.153	0.151 ± 0.002	0.180	0.178 ± 0.001
336	0.210	0.205 ± 0.005	0.217	0.217 ± 0.001
720	0.348	0.341 ± 0.006	0.292	0.288 ± 0.007
Avg.	0.214	0.209 ± 0.005	0.214	0.212 ± 0.003

D.6 Mechanism of the reconstruction objective

Prediction-only training constrains the network only through the future target, so many functions fit the training data equally well, including ones that change sharply between nearby histories. Reconstruction adds supervision on every history coordinate. Since $F = [f_{\text{rec}}, f_{\text{pred}}]$ is one map, the objective fits its history block to the identity and its future block to m together; under bounded residual curvature, a small residual $F - G$ implies a small Jacobian residual

$$J_F - J_G = \big[\, J_{\text{rec}} - I \;\; J_{\text{pred}} - J_m \,\big], \tag{9}$$

whose prediction block is the excess sensitivity $J_{\text{pred}} - J_m$. This effect is not trivial for three reasons: (i) training histories are clean, so there is no noise to memorize; (ii) the reconstruction is discarded at inference, so the gain is not test-time denoising; (iii) W_c, W_p are fixed selections and only the encoder is learned, so reconstruction cannot bypass the representation used for forecasting. Because the reference is J_m rather than zero, sensitivity to changes that affect the future is kept. The bound holds in aggregate, not per direction; the toy task (Table 5) tests this empirically.

The theorems predict that the reconstruction term lowers both the full-sequence loss and S. We compare RAP_{PO} ($\lambda_{\text{rec}} = 0$) and RAP_{RA} ($\lambda_{\text{rec}} = 1$), which differ only in the loss, on held-out clean windows, and estimate S with Algorithm 3. Both quantities decrease on both datasets (Table 12), so the reduction comes from the objective, not from capacity.

D.7 Seed variability

RAP is deterministic at inference, so variability comes only from initialization and optimization. We retrained RAP_{RA} on Solar with five seeds in the clean setting (Table 13). The five-seed mean is slightly better than the single run at every horizon in MSE, and the standard deviation is at most 0.006 in MSE and 0.007 in MAE.

D.8 Fine-grained noise sweep

Table 14 re-evaluates the checkpoints of Table 10 on a finer grid (up to $\sigma = 0.025$, then steps of 0.025 up to 0.2). On Traffic there is no turning point: the gain of RA over PO rises smoothly from -2.0% to $+11.3\%$ and crosses zero near $\sigma \approx 0.1$ (MSE) and 0.13 (MAE). This matches $\mathcal{R}_\sigma \approx \mathcal{R}_0 + \sigma^2 S$: PO has lower clean risk (0.355 vs. 0.362) but RA has lower S (Table 12), so PO wins at small σ and RA at larger σ. On Solar both curves rise sharply at first and then separate:

Table 14: Fine-grained noise sweep for $\mathrm{RAP}_{\mathrm{PO}}$ and $\mathrm{RAP}_{\mathrm{RA}}$ in the small-noise range, reusing the checkpoints of Table 10 (MSE / MAE, averaged over $H \in \{96, 192, 336, 720\}$). Gain is the relative improvement of RA over PO; positive means RA is better. Each entry uses a single noise draw.

	Traffic						Solar					
σ	PO		RA		Gain (%)		PO		RA		Gain (%)	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
0.000	0.355	0.351	0.362	0.363	−2.0	−3.4	0.214	0.215	0.213	0.214	+0.5	+0.5
0.001	0.355	0.351	0.362	0.363	−2.0	−3.4	0.213	0.215	0.213	0.214	0.0	+0.5
0.005	0.355	0.351	0.362	0.363	−2.0	−3.4	0.334	0.281	0.313	0.276	+6.3	+1.8
0.010	0.355	0.351	0.362	0.363	−2.0	−3.4	0.599	0.438	0.589	0.442	+1.7	−0.9
0.015	0.356	0.352	0.363	0.364	−2.0	−3.4	0.701	0.520	0.697	0.568	+0.6	−9.2
0.020	0.356	0.352	0.363	0.364	−2.0	−3.4	0.705	0.549	0.784	0.635	−11.2	−15.7
0.025	0.357	0.353	0.364	0.365	−2.0	−3.4	0.726	0.573	0.816	0.669	−12.4	−16.8
0.050	0.364	0.358	0.369	0.369	−1.4	−3.1	1.066	0.732	0.970	0.712	+9.0	+2.7
0.075	0.375	0.367	0.377	0.377	−0.5	−2.7	1.391	0.855	1.045	0.737	+24.9	+13.8
0.100	0.390	0.380	0.388	0.387	+0.5	−1.8	1.491	0.899	1.101	0.763	+26.2	+15.1
0.125	0.423	0.401	0.407	0.402	+3.8	−0.2	1.544	0.931	1.134	0.780	+26.6	+16.2
0.150	0.457	0.423	0.426	0.417	+6.8	+1.4	1.586	0.959	1.151	0.793	+27.4	+17.3
0.175	0.490	0.444	0.445	0.431	+9.2	+2.9	1.629	0.987	1.167	0.806	+28.4	+18.3
0.200	0.523	0.465	0.464	0.446	+11.3	+4.1	1.670	1.013	1.182	0.818	+29.2	+19.2

Table 15: Relative MSE improvement (%) of adding the reconstruction term to iTransformer, as a function of context length C (averaged over $H \in \{96, 192, 336, 720\}$). Computed from Tables 17 and 18.

	ECL				Traffic			
σ	C=96	192	336	720	C=96	192	336	720
0.0	+2.2	+2.3	+0.6	+0.6	−2.1	−1.8	−0.8	−1.8
0.3	+10.7	+12.4	+12.6	+12.2	+1.5	+2.1	+4.2	+2.8
0.5	+21.7	+24.7	+25.9	+25.9	+11.4	+12.5	+15.5	+14.9
0.7	+27.1	+30.8	+33.6	+33.2	+20.0	+21.9	+26.3	+26.7

from $\sigma = 0.05$ to 0.2, the error of RA rises by 22% and that of PO by 57%. RA is worse only for $\sigma \in [0.015, 0.025]$; since each point uses a single noise draw, we do not treat this band as reliable.

D.9 Effect of context length

We train DLinear and iTransformer with and without reconstruction on ECL and Traffic with $C \in \{96, 192, 336, 720\}$, average over $H \in \{96, 192, 336, 720\}$, and corrupt the history at test time with $\sigma \in [0, 0.7]$. PO and RA differ only in the loss. Full results are in Tables 17 and 18; the $C = 96$ rows match Table 9.

Clean rankings can reverse. All models improve with longer context, but not equally: on Traffic, iTransformer beats DLinear at $C = 96$ (0.422 vs. 0.450), but DLinear wins clearly at $C = 720$ (0.283 vs. 0.379).

Reconstruction turns context into robustness. For iTransformer, the gain from reconstruction grows with C and saturates near $C = 336$ (Table 15). This is not because the baseline degrades faster: its ratio $\mathcal{R}_{0.7}/\mathcal{R}_0$ is flat in C, while that of RA decreases (Table 16). Both variants gain similarly from longer context on clean data, but at $\sigma = 0.7$ RA gains more from $C = 96 \to 720$: 22.4% vs. 15.3% on ECL and 19.0% vs. 11.5% on Traffic.

Backbone dependence. For DLinear on ECL, RA is worse than PO at low noise (e.g., 0.305 vs. 0.256 at $\sigma = 0.1$, $C = 96$) and better only at higher noise, for every C.

Table 16: Where the widening margin comes from (iTransformer backbone, MSE). *Left:* degradation ratio $R_{0.7}/R_0$ per context length. *Right:* relative MSE reduction obtained by extending the context from C=96 to C=720, per noise level. Computed from Tables 17 and 18.

		$R_{0.7}/R_0$				Context benefit C=96→720 (%)			
Dataset	Obj.	C=96	192	336	720	σ=0	0.2	0.4	0.7
ECL	PO	12.14	10.95	10.98	11.15	7.8	9.0	12.6	15.3
	RA	9.05	7.76	7.33	7.49	6.2	9.2	15.6	22.4
Traffic	PO	3.87	3.79	3.82	3.81	10.2	10.5	11.6	11.5
	RA	3.03	2.91	2.79	2.74	10.4	11.4	13.7	19.0

Table 17: Full context-length results on ECL under test-time Gaussian history corruption (MSE / MAE, averaged over $H \in \{96, 192, 336, 720\}$). Models are trained on clean histories. Within each backbone and context length C, PO and RA share architecture, output shape, hyperparameters and inference cost; only $\mathcal{L}_{\text{rec}}$ differs. Bold: better of PO/RA for the same backbone and C.

			$\sigma=0.0$		$\sigma=0.1$		$\sigma=0.2$		$\sigma=0.3$		$\sigma=0.4$		$\sigma=0.5$		$\sigma=0.6$		$\sigma=0.7$	
Backbone	Obj.	C	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
DLinear	PO	96	0.225	0.319	**0.256**	**0.354**	**0.347**	**0.430**	**0.500**	**0.525**	**0.714**	**0.631**	0.988	0.744	1.324	0.860	1.721	0.979
		192	0.187	0.289	**0.213**	**0.320**	**0.289**	**0.390**	**0.417**	**0.476**	**0.596**	**0.574**	0.826	0.678	1.109	0.785	1.444	0.895
		336	0.173	0.275	**0.197**	**0.305**	**0.267**	**0.371**	**0.384**	**0.453**	**0.549**	**0.546**	0.760	0.644	1.019	0.746	1.326	0.851
		720	0.164	0.264	**0.186**	**0.293**	**0.252**	**0.356**	**0.363**	**0.436**	**0.518**	**0.524**	0.715	0.619	0.957	0.716	1.241	0.815
	RA	96	**0.168**	**0.266**	0.305	0.412	0.442	0.508	0.579	0.587	0.716	0.655	**0.853**	**0.717**	**0.990**	**0.774**	**1.127**	**0.826**
		192	**0.140**	**0.241**	0.254	0.373	0.368	0.460	0.482	0.533	0.597	0.596	**0.713**	**0.653**	**0.829**	**0.706**	**0.946**	**0.755**
		336	**0.129**	**0.229**	0.234	0.355	0.340	0.438	0.445	0.507	0.551	0.566	**0.656**	**0.621**	**0.762**	**0.672**	**0.868**	**0.718**
		720	**0.122**	**0.220**	0.222	0.341	0.321	0.421	0.420	0.487	0.519	0.544	**0.617**	**0.596**	**0.715**	**0.644**	**0.812**	**0.687**
iTransformer	PO	96	0.180	0.270	**0.214**	0.317	0.335	0.429	0.540	0.563	0.825	0.704	1.194	0.851	1.648	1.001	2.186	1.153
		192	0.171	0.264	0.203	0.309	0.315	0.415	0.501	0.538	0.749	0.664	1.061	0.793	1.434	0.923	1.872	1.057
		336	0.166	0.260	**0.197**	0.305	0.305	0.408	0.484	0.527	0.723	0.649	1.023	0.776	1.388	0.906	1.822	1.042
		720	0.166	0.261	**0.197**	**0.306**	0.305	0.408	0.482	0.527	0.721	0.649	1.025	0.777	1.398	0.910	1.851	1.051
	RA	96	**0.176**	**0.266**	**0.214**	**0.316**	**0.325**	**0.419**	**0.482**	**0.528**	**0.684**	**0.638**	**0.935**	**0.752**	**1.237**	**0.868**	**1.593**	**0.986**
		192	**0.167**	**0.260**	**0.202**	**0.308**	**0.303**	**0.404**	**0.439**	**0.501**	**0.604**	**0.593**	**0.799**	**0.686**	**1.028**	**0.780**	**1.296**	**0.877**
		336	**0.165**	**0.258**	**0.197**	**0.303**	**0.294**	**0.396**	**0.423**	**0.488**	**0.577**	**0.575**	**0.758**	**0.662**	**0.967**	**0.750**	**1.210**	**0.840**
		720	**0.165**	**0.260**	0.198	**0.306**	**0.295**	**0.399**	**0.423**	**0.490**	**0.577**	**0.577**	**0.760**	**0.665**	**0.977**	**0.757**	**1.236**	**0.854**

E MODEL ARCHITECTURE

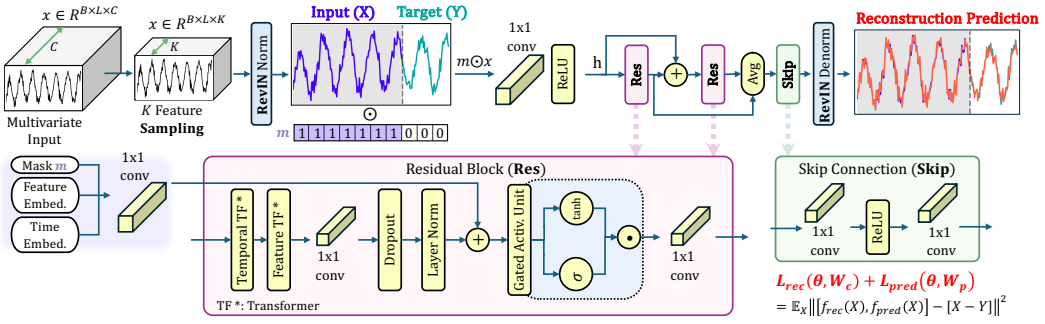


Figure 10: Overall RAP backbone with dual-axis attention and shared representation for reconstruction and forecasting.

RAP uses a single encoder Φ_θ that maps a conditioned history to a shared latent, followed by two linear heads: a reconstruction head $f_{\text{rec}}(x) = W_c\,\Phi_\theta(x)$ and a prediction head $f_{\text{pred}}(x) = W_p\,\Phi_\theta(x)$. The backbone is *DiffWave-inspired* but non-diffusive (no t-embedding/schedule) and mixes information via *dual-axis self-attention* (time-then-feature) inside residual blocks. The network outputs $[\hat{\mathbf{x}}_{1:T}, \hat{\mathbf{y}}_{1:H}]$ in a single pass, which lets us train both reconstruction and forecasting losses under one architecture.

Table 18: Full context-length results on Traffic under test-time Gaussian history corruption (MSE / MAE, averaged over $H \in \{96, 192, 336, 720\}$). Models are trained on clean histories. Within each backbone and context length C, PO and RA share architecture, output shape, hyperparameters and inference cost; only $\mathcal{L}_{\text{rec}}$ differs. Bold: better of PO/RA for the same backbone and C.

			$\sigma = 0.0$		$\sigma = 0.1$		$\sigma = 0.2$		$\sigma = 0.3$		$\sigma = 0.4$		$\sigma = 0.5$		$\sigma = 0.6$		$\sigma = 0.7$	
Backbone	Obj.	C	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
DLinear	PO	96	0.450	0.326	0.582	0.467	0.730	0.559	0.878	0.651	1.025	0.719	1.173	0.786	1.320	0.843	1.468	0.899
		192	0.338	0.269	0.437	0.386	0.549	0.464	0.661	0.544	0.774	0.605	0.888	0.667	1.003	0.721	1.120	0.775
		336	0.304	0.248	0.393	0.356	0.493	0.428	0.594	0.502	0.695	0.560	0.798	0.617	0.901	0.668	1.005	0.719
		720	0.283	0.231	0.366	0.332	0.459	0.399	0.552	0.468	0.645	0.521	0.739	0.573	0.832	0.620	0.927	0.666
	RA	96	**0.437**	**0.301**	**0.569**	**0.457**	**0.702**	**0.552**	**0.834**	**0.628**	**0.966**	**0.695**	**1.098**	**0.755**	**1.230**	**0.810**	**1.363**	**0.860**
		192	**0.328**	**0.248**	**0.427**	**0.378**	**0.528**	**0.458**	**0.628**	**0.525**	**0.729**	**0.585**	**0.831**	**0.640**	**0.935**	**0.693**	**1.040**	**0.741**
		336	**0.295**	**0.229**	**0.384**	**0.348**	**0.474**	**0.423**	**0.564**	**0.485**	**0.655**	**0.541**	**0.747**	**0.593**	**0.839**	**0.642**	**0.934**	**0.688**
		720	**0.275**	**0.214**	**0.358**	**0.325**	**0.441**	**0.394**	**0.525**	**0.451**	**0.608**	**0.503**	**0.692**	**0.551**	**0.776**	**0.595**	**0.861**	**0.637**
iTransformer	PO	96	**0.422**	**0.282**	**0.445**	**0.319**	**0.522**	0.398	0.654	0.495	0.834	0.598	1.057	0.704	1.323	0.811	1.633	0.919
		192	**0.396**	**0.274**	**0.417**	**0.309**	**0.488**	0.383	0.609	0.474	0.774	0.569	0.976	0.668	1.218	0.769	1.501	0.873
		336	**0.387**	**0.273**	0.408	0.308	0.477	0.380	0.595	0.469	0.755	0.562	0.954	0.660	1.193	0.761	1.478	0.867
		720	**0.379**	**0.271**	**0.399**	**0.305**	**0.467**	0.376	0.581	0.462	0.737	0.553	0.931	0.650	1.165	0.752	1.445	0.860
	RA	96	0.431	0.289	0.451	0.320	0.528	**0.392**	**0.644**	**0.474**	**0.782**	**0.557**	**0.936**	**0.638**	**1.110**	**0.718**	**1.307**	**0.798**
		192	0.403	0.280	0.421	**0.309**	0.491	**0.376**	**0.596**	**0.451**	**0.718**	**0.526**	**0.854**	**0.598**	**1.004**	**0.670**	**1.172**	**0.740**
		336	0.390	0.278	**0.406**	**0.306**	**0.472**	**0.370**	**0.570**	**0.441**	**0.684**	**0.511**	**0.806**	**0.578**	**0.940**	**0.643**	**1.089**	**0.707**
		720	0.386	0.278	0.403	0.307	0.468	**0.370**	**0.565**	**0.439**	**0.675**	**0.507**	**0.792**	**0.570**	**0.919**	**0.632**	**1.059**	**0.694**

E.1 Model Architecture details

Our backbone follows DiffWave (Kong et al., 2020) in spirit but differs in three key aspects. Because DiffWave-style generators map inputs to outputs of the *same shape* to learn score function, we feed the model the concatenated sequence $[\,\mathbf{x}_{1:T},\, \mathbf{0}\,]$ (zero-padding the future) and train it to produce $[\,\hat{\mathbf{x}}_{1:T},\, \hat{\mathbf{y}}_{1:H}\,]$. This single-pass, same-shape design naturally supports both the forecasting and reconstruction objectives, enabling comparison of the effect of the reconstruction term within a single architecture.

First, we *remove the diffusion-step (t) embedding and the entire diffusion schedule*, training a one-shot forecaster with a plain MSE objective. Second, each residual block mixes information via *dual-axis self-attention* (along time and along features) instead of dilated convolutions, explicitly capturing long-range temporal patterns and cross-variable interactions. Third, we employ *LayerNorm* and *Dropout* within blocks. For forecasting, we optionally apply *Reversible Instance Normalization* (RevIN). The decoder generates the full H-step target in one shot, conditioned on the encoded history.

Remark. The same-shape interface is the only element inherited from DiffWave; the diffusion timestep embedding, noise schedule, score matching, iterative reverse process and dilated-convolution stack are all removed. We keep this interface because it lets reconstruction and prediction be two slices of one output map. This makes the ablation between $\mathrm{RAP}_{\mathrm{PO}}$ and $\mathrm{RAP}_{\mathrm{RA}}$ free of capacity confounds. Also, The residual blocks of RAP use smooth gates, softmax attention, LayerNorm, and 1×1 convolutions; the pointwise ReLUs in its lift/output paths make the implemented network piecewise smooth, but a Softplus (C^∞) variant, for which the assumption holds exactly, performs indistinguishably. Appendix H.1 discusses which part of the assumption each result uses, a weaker condition that suffices in place of the bounded score, and the treatment of windows as i.i.d. pairs.

E.2 Inputs and Side Information

Let $\mathbf{x} \in \mathbb{R}^{B\times L\times K}$ be a batch of K-variate sequences of length L and $\mathbf{m} \in \{0,1\}^{B\times L\times K}$ a binary mask of observed entries. We form the conditioned input

$$\mathbf{x}^{\text{cond}} \;=\; \mathbf{m} \odot \mathbf{x} \;\in\; \mathbb{R}^{B\times L\times K}.$$

We also provide side information

$$\text{side} \in \mathbb{R}^{B\times S\times K\times L}, \quad \text{side} = \big[\, \mathrm{PE}_{d_{\text{time}}}(t) \,\|\, \mathrm{Emb}_{d_{\text{feat}}}(\text{feature id}) \,\|\, m \,\big],$$

i.e., sinusoidal time positions, learnable feature embeddings, and the mask channel. *There is no diffusion step embedding* (training and inference are one-shot).

E.3 TOP-LEVEL NETWORK

Denote base width C and number of residual blocks B_{res}.

1. **Channel lift.** Reshape x^{cond} to $(B, 1, K{\cdot}L)$, apply a $1{\times}1$ Conv1d to C channels (Kaiming init) followed by a ReLU, and reshape to (B, C, K, L).
2. **Residual stack.** Apply B_{res} identical `ResidualBlock`s. Each block emits an updated hidden state and a skip tensor; skip tensors are summed and scaled by $1/\sqrt{B_{\text{res}}}$.
3. **Output head & zero-init.** Flatten $(B, C, K, L) \to (B, C, K \cdot L)$, apply $1{\times}1$ Conv1d $(C \to C)$ followed by a ReLU, and a final $1{\times}1$ Conv1d $(C \to 1)$ whose weights are *zero-initialized* for stable residual learning. Reshape to (B, K, L) to obtain $\hat{x}$, which contains $[\hat{\mathbf{x}}_{1:T}, \hat{\mathbf{y}}_{1:H}]$.

E.4 RESIDUAL BLOCK: DUAL-AXIS ATTENTION WITH GATED RESIDUAL

Given hidden $h \in \mathbb{R}^{B\times C\times K\times L}$ and side info side, each block performs:

1. **Time-axis attention.** For each feature k, apply a linear-attention along the L time steps, yielding (B, C, K, L).
2. **Feature-axis attention.** For each time index ℓ, apply the same encoder along K features (cross-variate coupling).
3. **Conditioning and normalization.** Flatten to $(B, C, K{\cdot}L)$; project to $2C$ channels via $1{\times}1$ Conv1d. In parallel, project `side` with $1{\times}1$ Conv1d $(S \to 2C)$ and add. Apply **Dropout** (p=0.2) and **LayerNorm** on the $2C$ channels.
4. **Gated update and skip.** Split into $(\text{gate}, \text{filter}) \in \mathbb{R}^{B\times C\times K\cdot L}$, use $\sigma(\text{gate}) \odot \tanh(\text{filter})$, then a $1{\times}1$ Conv1d $(C \to 2C)$ and split to $(\text{residual}, \text{skip})$. Return

$$h' = \frac{h + \text{residual}}{\sqrt{2}}, \qquad \text{skip} \in \mathbb{R}^{B\times C\times K\times L}.$$

This axial (time→feature) attention captures long-range temporal context and cross-variable interactions without quadratic 2D attention cost.

E.5 FORECASTING WRAPPER (REVIN AND FEATURE SUBSAMPLING)

For forecasting, we wrap the base network with: (i) **RevIN** (affine off by default) applied per variable before the model, and inverted on outputs; (ii) optional **feature subsampling** during training to cut memory/compute on high-K datasets. The mask channel is always included in forecasting.

E.6 DIFFERENCES FROM DIFFWAVE/CSDI

Although our backbone is inspired by DiffWave-style architectures, it is *non-diffusive* and departs from both DiffWave and CSDI in several important ways. First, we remove the diffusion time-step embedding and the entire noise schedule, and train a *deterministic, one-shot* mapper with a plain MSE objective. This eliminates score matching and iterative sampling, simplifying the objective and yielding a single forward pass at inference (lower latency and fewer hyperparameters tied to diffusion schedules). Second, instead of the dilated-convolution residual stacks used in DiffWave, each of our residual blocks performs *dual-axis self-attention*—attention along the temporal axis and, separately, along the feature axis. This axial design explicitly models long-range temporal dependencies and cross-variable interactions that are central to multivariate LTSF. Third, we adopt *LayerNorm* and *Dropout* inside blocks (pre-activation), which improves training stability across datasets and provides lightweight regularization. Fourth, *Reversible Instance Normalization (RevIN)* is used only in the forecasting wrapper: inputs are normalized per variable before the encoder and inverted on the forecasted outputs, mitigating distribution shift specifically for the forecasting path.

F RAP HYPERPARAMETERS AND TRAINING PROCEDURE

This appendix describes the hyperparameters and training/inference procedure of RAP only. Baseline models are trained with their own public default/recommended hyperparameter configurations. Across models, we unify only the benchmarking protocol (dataset splits, lookback length, prediction horizons, and evaluation metrics). For RAP, we tune only a small set of hyperparameters—*(i)* **layers**, *(ii)* **channels**, *(iii)* **learning rate**, and *(iv)* whether to use **RevIN** (reversible instance normalization). For each baseline, we adopt the model-specific hyperparameters recommended in their original papers and official implementations; only the shared experimental settings (lookback length, prediction horizons, dataset splits) are unified across all methods for a fair comparison."

For each dataset, we selected the number of layers and channels, tuned the learning rate, and toggled RevIN usage, while keeping all remaining training options (optimizer, weight decay, batch size, scheduler, early stopping criteria, *etc.*). Table 19 summarizes the final settings per dataset.

Table 19: Hyperparameter settings by dataset.

Dataset	Layers	Channels	Learning rate	use_revin
Solar	2	16	0.005	True
ECL	2	64	0.0005	False
Traffic	2	16	0.001	True
PEMS03	2	32	0.001	False
PEMS04	2	32	0.001	False
PEMS07	2	32	0.001	False

Next, we describe the detailed algorithm. RAP follows a standard supervised forecasting loop and inherits the optimizer/scheduler template from the benchmark implementation in the shared codebase, except for the tuned learning rate. Baselines are trained using their own model-specific default/recommended recipes.

Algorithm 1 Inference and Robustness Evaluation (noisy risk sweep)

Input: Trained (θ, W_p, W_c), test set $\{(x, y)\}$, noise levels $\{\sigma_j\}_{j=0}^{M}$ (with $\sigma_0 = 0$).
Output: MSE vs. noise curve: $\mathrm{MSE}(\sigma_j)$, optionally noisy risk estimate $\mathcal{R}_{\sigma_j}$.
for each noise level σ_j **do**
  Initialize accumulators mse $\leftarrow 0$
  for each test pair (x, y) **do**
    Sample $\varepsilon \sim \mathcal{N}(0, I_d)$
    Form noisy input $\tilde{x} \leftarrow x + \sigma_j \varepsilon$
    Compute forecast $\hat{y} \leftarrow f_{\mathrm{pred}}(\tilde{x}) = W_p \Phi_\theta(\tilde{x})$
    Accumulate mse $\leftarrow$ mse $+ \|\hat{y} - y\|^2$
  end for
  Normalize: $\mathrm{MSE}(\sigma_j) \leftarrow \mathrm{mse}/N_{\mathrm{samples}}$
end for
return $\{\mathrm{MSE}(\sigma_j)\}_{j=0}^{M}$

Algorithm 2 RAP: Reconstruction-Augmented Predictors (training & evaluation)

Input: Dataset $\{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$, lookback T, horizon H, Batch size B, epochs E.
Input: Encoder $\Phi_\theta : \mathbb{R}^{T\times F} \to \mathbb{R}^{D\times F}$, linear heads $W_c \in \mathbb{R}^{T\times D}$, $W_p \in \mathbb{R}^{H\times D}$. Batch size B, epochs E.
Output: Trained parameters θ.
Initialize parameters θ and set W_c, W_p by $D = T + H$, $W_c = [I_T, 0]$, $W_p = [0, I_H]$ Initialize optimizer (e.g., Adam), LR schedule (e.g., cosine)
for epoch $= 1$ **to** E **do**
  Shuffle dataset
  for each minibatch $\{(x_b, y_b)\}_{b=1}^{B}$ **do**
    // forward
    Compute latent representations: $z_b \leftarrow \Phi_\theta(x_b)$ for all b
    Compute reconstruction outputs: $\hat{x}_b \leftarrow f_{\text{rec}}(x_b) = W_c z_b$
    Compute prediction outputs: $\hat{y}_b \leftarrow f_{\text{pred}}(x_b) = W_p z_b$
    // losses
    Prediction loss: $\mathcal{L}_{\text{pred}} \leftarrow \frac{1}{B}\sum_b \|\hat{y}_b - y_b\|^2$
    Reconstruction loss: $\mathcal{L}_{\text{rec}} \leftarrow \frac{1}{B}\sum_b \|\hat{x}_b - x_b\|^2$
    if task = PO **then**
      Composite loss: $\mathcal{L} \leftarrow \mathcal{L}_{\text{pred}}$
    else if task = RA **then**
      Composite loss: $\mathcal{L} \leftarrow \mathcal{L}_{\text{pred}} + \mathcal{L}_{\text{rec}}$
    end if
    // backward + update
    Backpropagate $\nabla_\theta \mathcal{L}$
    // optional spectral constraints projection
  end for
  Update LR scheduler
end for
return θ

Algorithm 3 Estimate Jacobian Energy $S = \mathbb{E}_X \|J_{\text{pred}}(X)\|_F^2$ (Hutchinson)

Input: Trained (θ, W_p), sample set $\{x^{(i)}\}_{i=1}^{N_s}$, Hutchinson draws per x: R (e.g., 1–5).
Output: Estimate $\widehat{S}$.
$\widehat{S} \leftarrow 0$
for $i = 1$ **to** N_s **do**
  $x \leftarrow x^{(i)}$
  for $r = 1$ **to** R **do**
    Sample $v \sim \mathcal{N}(0, I_d)$ (or Rademacher $\{\pm 1\}^d$)
    // use auto-diff: Jacobian-vector product (JVP)
    Compute directional Jacobian product: $g \leftarrow \nabla_\delta\big|_{\delta=0} f_{\text{pred}}(x + \delta v)$
    Accumulate $\widehat{S} \leftarrow \widehat{S} + \|g\|^2$
  end for
end for
$\widehat{S} \leftarrow \widehat{S}/(N_s R)$
// approx. $\mathbb{E}_X \|J_{\text{pred}}(X)\|_F^2$
return $\widehat{S}$

G Comparison with SCAM

SCAM (Yang et al., 2025) is a closely related self-reconstruction-based forecasting method. The main difference is the target robustness setting. SCAM is designed for contaminated training data and uses reconstruction to improve training targets. RAP keeps the training data fixed and targets corrupted history windows at evaluation time. The reconstructed object is also different: SCAM

reconstructs forecasting targets, while RAP reconstructs the input history through the same representation used for prediction. This distinction is important under our protocol, where all models are trained on clean histories and tested on corrupted histories.

To make the comparison direct, we evaluate SCAM under the same increasing-noise inference protocol used in Section 7.3. Table 20 compares iTransformer, iTransformer with SCAM, and iTransformer with the reconstruction-augmented objective. SCAM improves the noise-free iTransformer result on ECL and Traffic, matching its training-target correction goal. Under corrupted histories, the reconstruction-augmented objective gives stronger noisy-input performance at larger noise scales. For example, on Traffic at $\sigma = 0.4$, $\mathrm{iTransformer}_{\mathrm{SCAM}}$ gives $1.068/0.709$, while $\mathrm{iTransformer}_{\mathrm{RA}}$ gives $0.782/0.557$.

Table 20: Comparison with SCAM under test-time Gaussian history corruption. Rows: Model; Columns: noise scales. We report averaged results across prediction lengths $H = \{96, 192, 336, 720\}$.

Dataset	Model	0.0		0.1		0.2		0.3		0.4	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ECL	iTransformer	0.180	0.270	0.214	0.317	0.335	0.429	0.540	0.563	0.825	0.704
	$\mathrm{iTransformer}_{\mathrm{SCAM}}$	0.173	0.267	0.236	0.340	0.373	0.456	0.536	0.558	0.724	0.655
	$\mathrm{iTransformer}_{\mathrm{RA}}$	0.176	0.266	0.214	0.316	0.325	0.419	0.482	0.528	0.684	0.638
Traffic	iTransformer	0.422	0.282	0.445	0.319	0.522	0.398	0.654	0.495	0.834	0.598
	$\mathrm{iTransformer}_{\mathrm{SCAM}}$	0.411	0.266	0.466	0.335	0.606	0.459	0.816	0.588	1.068	0.709
	$\mathrm{iTransformer}_{\mathrm{RA}}$	0.431	0.289	0.451	0.320	0.528	0.392	0.644	0.474	0.782	0.557

We further test whether extending SCAM from target reconstruction to history reconstruction addresses test-time input corruption. There are two natural extensions. Case 1 uses the reconstruction map g only during training: the predictor is trained on $\widetilde{X} = g(X + Z_{\mathrm{tr}})$ and receives the corrupted history $U_{\mathrm{te}} = X + Z_{\mathrm{te}}$ directly at test time. Case 2 also applies g at inference, so the predictor receives $g(U_{\mathrm{te}})$. Case 1 creates a train-test input mismatch under corrupted histories. Case 2 changes the method into a test-time denoising pipeline and can suppress predictive structure in the observed history.

This mechanism can be written using the Bayes predictor. Let $m(X) = \mathbb{E}[Y \mid X]$, $Z = g(X)$, and $m_g(Z) = \mathbb{E}[Y \mid Z]$. The best risk using only the reconstructed input Z is

$$R_g = \mathbb{E}|Y - m_g(Z)|^2 = \mathbb{E}|Y - m(X)|^2 + \mathbb{E}[\mathrm{Var}(m(X) \mid Z)]. \tag{10}$$

Thus, whenever g suppresses predictive components of the history, the conditional-variance term becomes positive. This matches the failure mode characterized in Proposition 2.2.

Table 21 reports the two history-extended cases. On ECL, Case 1 slightly improves the clean result, then loses robustness at higher noise levels. Case 2 recovers some robustness and pays a clean-accuracy cost. On Traffic, Case 1 remains worse than original SCAM across the sweep. These results support the distinction between SCAM's training-contamination setting and RAP's test-time history-corruption setting.

Table 21: History-extended SCAM cases under test-time Gaussian history corruption. Rows: Model; Columns: noise scales. We report averaged results across prediction lengths $H = \{96, 192, 336, 720\}$. Case 1 uses the history reconstruction map only during training. Case 2 also applies the reconstruction map at inference.

Dataset	Model	0.0		0.1		0.2		0.3		0.4	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ECL	SCAM	0.173	0.267	0.236	0.340	0.373	0.456	0.536	0.558	0.724	0.655
	Case 1	0.170	0.265	0.224	0.327	0.374	0.460	0.604	0.596	0.873	0.717
	Case 2	0.189	0.280	0.236	0.340	0.371	0.454	0.533	0.557	0.715	0.651
Traffic	SCAM	0.411	0.266	0.466	0.335	0.606	0.459	0.816	0.588	1.068	0.709
	Case 1	0.446	0.294	0.486	0.355	0.643	0.478	0.858	0.605	1.139	0.737
	Case 2	0.423	0.276	0.466	0.335	0.605	0.457	0.811	0.584	1.056	0.704

Table 22: ReLU (as used in the main experiments) vs. Softplus (C^∞) in the lift/output paths of RAP on Solar, under the protocol of Section 7.3 (MSE / MAE, averaged over $H \in \{96, 192, 336, 720\}$). Parameterization and cost are identical.

	$\sigma = 0.0$		0.2		0.4		0.6		0.7	
Activation	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ReLU (default)	0.214	0.214	1.139	0.815	1.221	0.873	1.336	0.932	1.400	0.962
Softplus (C^∞)	0.213	0.215	1.125	0.822	1.201	0.890	1.298	0.934	1.392	0.970

H PROOFS

H.1 SMALL-NOISE ROBUSTNESS VIA JACOBIAN ENERGY

Assumption H.1. The predictor $f_{\text{pred}} \in C^3(\mathbb{R}^d; \mathbb{R}^k)$ has bounded second- and third-order derivatives, denoted by H_f and M_f. The input distribution admits a positive C^1 density with bounded score $\|\nabla \log p_X\|_\infty \le L_p$.[10] The Bayes map m has finite Jacobian energy, and the residual map $r := F - G$ is C^2 with bounded curvature $H_r := \sup_x \|\nabla^2 r(x)\|_{\text{op}} < \infty$.

Regularity of the implemented network. Assumption 4.1 requires C^2/C^3 regularity. Within each residual block the nonlinearities are σ and $\tanh$, and LayerNorm, the 1×1 convolutions and the softmax attention maps are smooth. The channel-lift and output paths contain pointwise ReLUs, so the network used in the main experiments is piecewise smooth rather than globally C^3. Replacing these ReLUs by Softplus gives a C^∞ network with identical parameterization, output shape and cost; for that network the bounded-derivative conditions reduce to bounded learned weights, which can be monitored during training. Dropout is inactive in the deterministic inference map analysed here. Table 22 shows that the two networks behave indistinguishably under the protocol of Section 7.3, so the smoothness assumption is a device for the analysis, not a property the reported numbers depend on.

Which parts of the assumption are used. Theorem 4.2 identifies *what* must be controlled (the Jacobian energy as the leading noise-sensitivity term); its proof uses only a local Taylor expansion and a finite residual moment, and requires neither a positive density nor a bounded score. Theorem 4.3 explains *why* our objective controls that quantity, and the score bound enters it at a single step, a change of measure. What that step actually needs is weaker than a bounded score: only two-sided shift-comparability of p_X at the finite-difference scale t,

$$p_X(y \pm tv) \le e^{L_p t}\, p_X(y) \qquad \text{for all } y,\ \|v\| = 1,\ 0 < t \le t_{\max}, \tag{11}$$

which gives $\int \|r(y)\|^2 p_X(y \pm tv)\, dy \le e^{L_p t}\, \mathbb{E}\|r(X)\|^2$. We state the bounded score because it is the familiar sufficient condition in score-based and diffusion analyses (Hyvärinen, 2005; Song et al., 2021; Tashiro et al., 2021); it can fail for distributions concentrated near low-dimensional manifolds. The dependence is quantitative rather than binary: L_p enters only through a factor $e^{L_p t_{\max}}$ in c_d, so a near-singular distribution loosens the bound continuously instead of invalidating it. Moreover, the conclusion of the theorem is measurable on the data whose regularity is in question: we report the Jacobian energy S directly (Figure 1, Table 12), without assuming anything about p_X.

Scope. The analysis treats history–future windows (X, Y) as draws from a fixed joint distribution and does not model the dependence between overlapping windows; extending it to stationary, mixing sequences is left for future work. The theorem also controls the excess sensitivity in aggregate rather than direction-wise, and the guarantee is local in σ; results at moderate and large noise levels are empirical.

Theorem H.2. *Under Assumption 4.1,*

$$\mathcal{R}_\sigma(f_{\text{pred}}) = \mathcal{R}_0(f_{\text{pred}}) \ + \ \sigma^2\, \mathbb{E}\, \|J_{\text{pred}}(X)\|_F^2 \ + \ \sigma^2\, \mathbb{E}\, \langle f_{\text{pred}}(X) - Y, \Delta f_{\text{pred}}(X) \rangle + \ O(\sigma^3),$$

[10] Score boundedness or regularity is common in score-based and diffusion generative-model analyses, where $\nabla \log p$ is the modeled object; see Hyvärinen (2005); Song et al. (2021); Tashiro et al. (2021).

where Δ is the Laplacian acting componentwise. In particular,

$$\mathcal{R}_\sigma(f_{\text{pred}}) - \mathcal{R}_0(f_{\text{pred}}) \;\le\; \sigma^2\,\mathbb{E}\,\|J_{\text{pred}}(X)\|_F^2 \;+\; c_{d,k}\,\sigma^2\,H\,\sqrt{\mathbb{E}\|f_{\text{pred}}(X)-Y\|^2} \;+\; C\,\sigma^3,$$

for constants $c_{d,k}, C$ depending on (d,k), the derivative bounds, and the finite residual moment used in the Taylor remainder.

Proof. Fix (X,Y) and expand the vector function $g(x) := f_{\text{pred}}(x) - Y$ at $x = X$:

$$g(X+\sigma\varepsilon) = g(X) + \sigma J_{\text{pred}}(X)\,\varepsilon + \frac{1}{2}\sigma^2\,h(X,\varepsilon) + \sigma^3 r_3(X,\varepsilon),$$

where the ℓ-th coordinate of h equals $\varepsilon^\top \nabla^2 g_\ell(X)\,\varepsilon$, and the remainder r_3 satisfies $\|r_3(X,\varepsilon)\| \le \frac{M}{6}\|\varepsilon\|^3$ by Taylor's theorem. Squaring and taking conditional expectation w.r.t. $\varepsilon \sim \mathcal{N}(0,I)$ yields

$$\begin{aligned}\mathbb{E}_\varepsilon\|g(X+\sigma\varepsilon)\|^2 &= \|g(X)\|^2 + \sigma^2\mathbb{E}_\varepsilon\|J_{\text{pred}}(X)\varepsilon\|^2 + \sigma^2\mathbb{E}_\varepsilon\langle g(X), h(X,\varepsilon)\rangle \\ &\quad + \sigma^3\mathbb{E}_\varepsilon\langle J_{\text{pred}}(X)\varepsilon, h(X,\varepsilon)\rangle + \frac{\sigma^4}{4}\mathbb{E}_\varepsilon\|h(X,\varepsilon)\|^2 + \sigma^3 O(\mathbb{E}_\varepsilon\|\varepsilon\|^3).\end{aligned}$$

The cross term $\mathbb{E}_\varepsilon\langle g, J_{\text{pred}}\varepsilon\rangle = 0$ by symmetry. Using $\mathbb{E}[\varepsilon\varepsilon^\top] = I$,

$$\mathbb{E}_\varepsilon\|J_{\text{pred}}(X)\varepsilon\|^2 = \operatorname{Tr}\big(J_{\text{pred}}(X)^\top J_{\text{pred}}(X)\big) = \|J_{\text{pred}}(X)\|_F^2.$$

Next, since the ℓ-th component of $\mathbb{E}_\varepsilon h(X,\varepsilon)$is $\operatorname{Tr}(\nabla^2 g_\ell(X))$,

$$\mathbb{E}_\varepsilon\langle g(X), h(X,\varepsilon)\rangle = \sum_{\ell=1}^{k} g_\ell(X)\;\operatorname{Tr}(\nabla^2 g_\ell(X)) = \langle g(X), \Delta g(X)\rangle.$$

The collected Taylor remainder is $O(\sigma^3)$ under the stated C^3 bounded-derivative assumption and the finite residual moment $\mathbb{E}\|f_{\text{pred}}(X)-Y\|^2 < \infty$; thus we obtain equation 6. For the right upper bound, note that $|\operatorname{Tr}(\nabla^2 g_\ell)| \le d\,\|\nabla^2 g_\ell\|_{\text{op}} \le dH$ and apply Cauchy–Schwarz:

$$|\mathbb{E}\langle g(X), \Delta g(X)\rangle| \le \sqrt{\mathbb{E}\|g(X)\|^2}\,\sqrt{\mathbb{E}\|\Delta g(X)\|^2} \;\le\; \sqrt{\mathbb{E}\|g(X)\|^2}\;\cdot d\sqrt{k}\,H.$$

Absorbing constants gives the inequality. □

Remark H.3. When $L_{\text{pred}} = \mathbb{E}\|f_{\text{pred}}(X)-Y\|^2$ is small, the Laplacian inner-product term in equation 6 is negligible compared to the Jacobian energy. Therefore, reducing $S = \mathbb{E}\|J_{\text{pred}}(X)\|_F^2$ directly improves small-noise robustness.

The next theorem proves the full-sequence function-to-gradient bound used in Theorem 4.3 and derives the predictor sensitivity bound in the same statement.

Lemma H.4. *Let $A \in \mathbb{R}^{m\times d}$ and let $d\sigma$ be the surface measure on the unit sphere $\mathbb{S}^{d-1} \subset \mathbb{R}^d$ with area $|\mathbb{S}^{d-1}|$. Then*

$$\frac{1}{|\mathbb{S}^{d-1}|}\int_{\mathbb{S}^{d-1}} \|Av\|^2\,d\sigma(v) \;=\; \frac{1}{d}\,\|A\|_{\text{F}}^2.$$

In particular, for any differentiable $u : \mathbb{R}^d \to \mathbb{R}^m$ and any x,

$$\frac{1}{|\mathbb{S}^{d-1}|}\int_{\mathbb{S}^{d-1}} \|\nabla u(x)\,v\|^2\,d\sigma(v) \;=\; \frac{1}{d}\,\|\nabla u(x)\|_{\text{F}}^2.$$

Proof. By rotational invariance of the uniform distribution on $\mathbb{S}^{d-1}$,

$$\frac{1}{|\mathbb{S}^{d-1}|}\int_{\mathbb{S}^{d-1}} vv^\top\,d\sigma(v) \;=\; \alpha\,I_d \quad \text{for some } \alpha.$$

Taking traces and using $\|v\|^2 = 1$ gives $1 = \frac{1}{|\mathbb{S}^{d-1}|}\int \operatorname{Tr}(vv^\top)\,d\sigma = \alpha\operatorname{Tr}(I_d) = \alpha d$, hence $\alpha = \frac{1}{d}$. Therefore,

$$\frac{1}{|\mathbb{S}^{d-1}|}\int_{\mathbb{S}^{d-1}} \|Av\|^2\,d\sigma(v) = \frac{1}{|\mathbb{S}^{d-1}|}\int v^\top A^\top Av\,d\sigma(v) = \operatorname{Tr}\Big(A^\top A\cdot\frac{1}{d}I_d\Big) = \frac{1}{d}\,\|A\|_{\text{F}}^2.$$

Substitute $A = \nabla u(x)$ for the particular case. □

The next theorem proves the full-sequence function-to-gradient bound used in Theorem 4.3, and derives the predictor sensitivity bound in the same statement.

Theorem H.5. *Under Assumption H.1, for any $0 < t \le t_{\max}$, there exist constants $c_d, C_d > 0$, depending on $d, L_p, t_{\max}$, such that*

$$\mathbb{E}\|J_F(X) - J_G(X)\|_F^2 \le \frac{c_d}{t^2}\mathbb{E}\|F(X) - G(X)\|^2 + C_d H_r^2 t^2.$$

Consequently, the predictor sensitivity satisfies

$$S := \mathbb{E}\|J_{\mathrm{pred}}(X)\|_F^2 \le 2\mathbb{E}\|J_m(X)\|_F^2 + \frac{2c_d}{t^2} L_{\mathrm{RA}} + 2C_d H_r^2 t^2.$$

If $t_\star := \left(\frac{c_d L_{\mathrm{RA}}}{C_d H_r^2}\right)^{1/4} \le t_{\max}$, then

$$S \le 2\mathbb{E}\|J_m(X)\|_F^2 + 4(c_d C_d)^{1/2} H_r L_{\mathrm{RA}}^{1/2}.$$

Proof. Let $r(x) := F(x) - G(x)$. For $x \in \mathbb{R}^d$, $v \in \mathbb{S}^{d-1}$, and $t > 0$, the central-difference estimate gives

$$\|\nabla r(x)v\| \le \frac{1}{2t}\|r(x + tv) - r(x - tv)\| + \frac{H_r}{2}t.$$

Squaring, averaging over $v \in \mathbb{S}^{d-1}$, and using Lemma H.4 gives

$$\|J_F(x) - J_G(x)\|_F^2 \le \frac{c_d}{t^2}\int_{\mathbb{S}^{d-1}} \|r(x + tv) - r(x - tv)\|^2 \, d\sigma(v) + C_d H_r^2 t^2,$$

where dimensional constants absorb the normalization of the spherical measure. Integrating with respect to $p_X(x)\,dx$ yields

$$\mathbb{E}\|J_F(X) - J_G(X)\|_F^2 \le \frac{c_d}{t^2}\int_{\mathbb{R}^d}\int_{\mathbb{S}^{d-1}} \|r(x + tv) - r(x - tv)\|^2 \, d\sigma(v)\, p_X(x)\, dx + C_d H_r^2 t^2.$$

Using $\|a - b\|^2 \le 2\|a\|^2 + 2\|b\|^2$, Fubini's theorem, and the changes of variables $y = x \pm tv$, we obtain

$$\begin{aligned}&\int_{\mathbb{R}^d}\int_{\mathbb{S}^{d-1}} \|r(x + tv) - r(x - tv)\|^2 \, d\sigma(v)\, p_X(x)\, dx \\ &\le 2\int_{\mathbb{S}^{d-1}}\int_{\mathbb{R}^d} \|r(y)\|^2 \{p_X(y - tv) + p_X(y + tv)\}\, dy\, d\sigma(v).\end{aligned}$$

By the bounded-score assumption, for any $y \in \mathbb{R}^d$ and $v \in \mathbb{S}^{d-1}$,

$$\log p_X(y \pm tv) - \log p_X(y) = \int_0^1 \nabla \log p_X(y \pm \tau tv) \cdot (\pm tv)\, d\tau,$$

and therefore $p_X(y \pm tv) \le e^{L_p t} p_X(y)$.[11]