REVISITING EMBEDDINGS IN TIME SERIES FORECASTING FROM LATENT SPACE AND DATA PERSPECTIVES

Anonymous authors
Paper under double-blind review

ABSTRACT

Embeddings serve as the fundamental representation units in time series forecasting and impose critical inductive biases on model learning, substantially influencing forecasting performance. Despite their importance, existing studies on embeddings largely overlook the distinction between direct mapping and high-order interactions on embedding effectiveness. In this work, we systematically investigate the effectiveness of embeddings across diverse datasets, ranging from first-order mapping to high-order interactions. Through extensive large-scale hyperparameter exploration and evaluation-theoretic analysis, we find that embedding-based models often degenerate into simple linear mappings without high-order interactions. In contrast, under higher-order interactions, embeddings exhibit distinct sensitivities to specific data characteristics and demonstrate distinct behaviors under different encoder architectures, suggesting that embeddings cannot be studied independently of higher-order interaction mechanisms. Our forecasting comparison covers attention, MLP mixing, recurrent, state-space, frequency-based, and convolutional encoders. Dataset characteristics, spectral rank, and frozen linear probes connect view preferences to representation evolution. The spectral and probe analyses cover MLPMixer, Transformer, GRU, Mamba, and FFT. With fixed-budget probes, encoder outputs reduce readout error in 72 of 98 configurations, with gains varying by dataset and token view. A multiscale data-rank rule further selects embedding–encoder pairs before target-model training and recovers six source-derived reference choices under leave-one-dataset-out calibration. Together, these results connect view construction, representation evolution, and practical module selection. Code is available at https://anonymous.4open.science/r/RVEmbed-0C8B

1 INTRODUCTION

Time series forecasting (TSF) plays a critical role in a wide range of domains, including finance, transportation, meteorology, and energy systems (Schetakis et al., 2025; Ao, 2025; Nizam et al., 2022; Barzola-Monteses et al., 2025). However, real-world time series often exhibit complex non-stationarity, cross-variate coupling, and multi-scale temporal dynamics, posing substantial challenges for model design. In recent years, mainstream TSF models have increasingly evolved into a modular architecture consisting of an embedding module, an encoder, and a decoder (Zeng et al., 2023; Wang et al., 2026; Zhao et al., 2026). Among these components, the embedding module maps raw sequences into a latent space and constructs the basic representation on which subsequent interactions are performed.

As the first computational unit applied to raw data in deep forecasting models, the embedding module essentially defines the "initial view" through which the model observes the data, thereby implicitly injecting crucial inductive biases (Lian et al., 2018).

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.

Existing methods typically construct such views using different tokenization strategies, such as point-wise, patch-wise, and variate-wise embeddings. Extensive empirical studies have shown that different embedding strategies exhibit markedly different performance across datasets (Zhao et al., 2026; Chen et al., 2025). Although prior work has compared various embedding methods, most studies still lack a systematic understanding of their underlying mechanisms. More importantly, existing analyses often overlook a key fact: ***The role of embedding is not static; rather, it changes fundamentally with the order of feature interactions.***

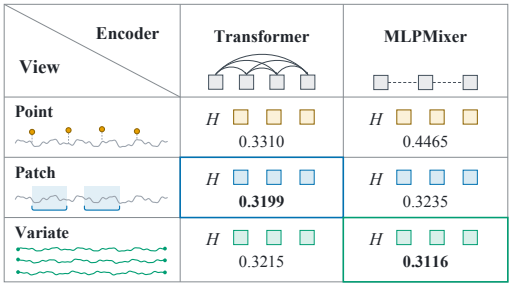


Figure 1: Joint influence of embedding view and encoder on ETTm1. The representative configurations use channel-independent Point/Patch and channel-dependent Variate embeddings (Appendix D). Values are mean test MSE $\hat{\mu}$ (lower is better); bold values and colored borders mark the lowest mean in each column. The preferred view changes with the encoder. Waveforms and encoder connections are schematic; H denotes the encoder representation.

Figure 1 illustrates how embedding views and encoder mechanisms jointly affect forecasting: the preferred view can change with the encoder, and the preferred encoder can change with the view.

We revisit the role of embeddings through four research questions:

Q1. First-order limit. When high-order feature interactions are removed, i.e., in the absence of an encoder, can embedding still provide an effective inductive bias?

Q2. High-order differentiation. Once a deep encoder is introduced, how does the data view constructed by embedding affect high-order nonlinear representation learning?

Q3. Representation evolution. How do spectral geometry and predictive readability vary with the data, embedding view, and encoder?

Q4. Module selection. Can multiscale statistics of the training series guide embedding–encoder selection before training a target model?

To answer these questions, we construct a probabilistic evaluation framework inspired by Evaluatology (Zhan et al., 2024). We compare embedding views across MLPMixer, Transformer, GRU, diagonal state-space, FFT-based auto-correlation, and convolutional encoders to examine how token construction interacts with each mixing mechanism. We characterize representation evolution using effective rank and SVD truncation, and independently fit linear probes before and after the encoder to measure access to forecasting targets. Finally, we use multiscale data-rank profiles to calibrate a rule for choosing an embedding–encoder pair.

Our analysis yields four findings:

F1. Degeneracy of first-order representations. In the absence of high-order interactions, different embeddings are functionally equivalent to linear mappings in terms of expressiveness, and their performance advantages are not statistically significant. This suggests that embedding alone does not independently provide effective modeling capacity.

F2. View sensitivity under high-order interactions. Once an encoder is introduced, the data view constructed by embedding becomes a critical factor affecting model performance. Different embeddings exhibit clear performance differentiation across datasets, and no universally optimal embedding strategy exists.

F3. Geometry and predictive readability. Spectral rank describes how representation energy is distributed, while linear probes measure how readily forecasting targets can be decoded. Both rank compression and expansion can accompany improved readability; their effects depend on the dataset and embedding view.

F4. Data-guided module selection. Effective rank at one input scale and its growth across scales jointly distinguish source-derived structure preferences. A calibrated rule reproduces all six reference choices when numerical thresholds are fitted without the target dataset.

In summary, the main contributions of this paper are as follows: (i) **Probabilistic Unified Evaluation.** Based on an evaluatology-inspired probabilistic evaluation protocol, we conduct large-scale assessments of Embedding performance under both low-order and high-order interaction settings,

yielding generalizable conclusions. (ii) **A representation-evolution view of embeddings.** We systematically analyze the role of embeddings from first-order to high-order interactions, revealing their evolution from *linear degeneracy* to *view-dominant behavior*. (iii) **Complementary representation diagnostics.** We combine rank analysis with frozen linear probes to relate latent geometry to target readability. (iv) **Training-free target selection.** We map multiscale data-rank profiles to source-derived module choices and evaluate agreement and sampling stability under leave-one-dataset-out calibration.

2 Related Work

Backbone. Deep learning-based TSF models can be broadly categorized into autoregressive and non-autoregressive types. Early research frequently relied on autoregressive methods, such as RNNs and LSTMs (Elman, 1990; Hochreiter & Schmidhuber, 1997). In recent years, non-autoregressive models have been widely recognized for their greater potential in developing lightweight architectures. Among these, the model backbones are primarily divided into two structures: Transformers and MLPs. Benefiting from exceptional performance on sequential data like natural language (Vaswani et al., 2017), Transformers have been extensively applied in TSF (Wu et al., 2021; Liu et al., 2024; Wang et al., 2024b). Concurrently, due to the more lightweight structures, MLP-based models have also been widely utilized in recent years (Zeng et al., 2023; Wang et al., 2024a; Tang & Zhang, 2025).

Module. As represented by Zeng et al. (2023), TSF models are frequently decomposed into preprocessing, embedding, encoder, and decoder components. To conduct an in-depth analysis of these modules, related studies typically disassemble models according to this paradigm and substitute different modules for performance evaluation. For example, TimeRecipe (Zhao et al., 2026) toggles the modules used in the embedding and Encoder components across various datasets to search for optimal performance. Building upon this, CombinationTS (Wang et al., 2026) introduces Evaluatology (Zhan et al., 2024) to further enhance the scientific rigor of the evaluation. Additionally, Chen et al. (2025) constructs synthetic datasets to further amplify the contrast between different modules.

Embedding. Beyond the differences in backbone architectures, a considerable amount of research has focused on the antecedent components of the backbone network, such as embedding and other preprocessing modules. Mainstream embedding methods include Patch-based embeddings represented by PatchTST (Nie et al., 2022) and PatchMLP (Tang & Zhang, 2025), Point-based embeddings represented by TimeMixer (Wang et al., 2024a) and Informer (Zhou et al., 2021), and Channel-based embeddings represented by iTransformer (Liu et al., 2024). Meanwhile, there are also models that do not utilize embedding, such as DLinear (Zeng et al., 2023) and RLinear (Li et al., 2026). These diverse strategies have all achieved excellent performance within their respective settings.

Although the aforementioned studies have drawn valid conclusions across embeddings, they neglect the deeply coupled relationships between modules. Specifically, the latent space constructed by the embedding exhibits distinct differences under varying interaction modes, such as the impact caused by whether channels interact or not (Han et al., 2024; Chen et al., 2025). Furthermore, the high-order relationships within the latent space constructed by the encoder also exert a direct impact on the efficacy of the embedding. Therefore, evaluating embeddings in isolation fails to accurately reflect their true effectiveness.

3 Preliminaries

Problem. TSF aims to predict future observations given a historical context window. Following the standard setting (Zeng et al., 2023; Nie et al., 2022; Liu et al., 2024), we consider a multivariate time series with C variables. Given an input sequence $X = [x_1, x_2, \ldots, x_L] \in \mathbb{R}^{L\times C}$, where L denotes the look-back length and $x_t \in \mathbb{R}^C$ is the observation at time step t, the goal is to predict the future sequence $Y = [x_{L+1}, x_{L+2}, \ldots, x_{L+H}] \in \mathbb{R}^{H\times C}$, where H is the forecasting horizon. A forecasting model f_θ learns a mapping from the historical window to the future horizon:

$$\hat{Y} = f_\theta(X), \qquad \hat{Y} \in \mathbb{R}^{H\times C}. \tag{1}$$

Module decomposition. Following a pipeline commonly adopted in recent TSF models(Zeng et al., 2023), we decompose a forecasting model into an embedding module, an encoder, and a decoder:

$$\hat{Y} = \mathcal{D} \circ \Phi \circ \mathcal{E}_m(X), \tag{2}$$

where $\mathcal{E}_m$ denotes an embedding strategy indexed by $m \in \mathcal{M}$, $\mathcal{M}$ is the candidate set of embedding strategies, Φ is the encoder, and $\mathcal{D}$ is the decoder. The embedding module first maps the raw sequence into latent tokens:

$$Z = \mathcal{E}_m(X), \qquad Z \in \mathbb{R}^{N \times d}, \tag{3}$$

where N is the number of tokens produced by the embedding strategy and d is the hidden dimension. Typical choices of $\mathcal{E}_m$ include point-wise, patch-wise, and variate-wise embeddings, which organize raw observations into tokens along different dimensions.

4 Rethinking: Inductive Bias and Evolution in the Embedding Latent Space

Prior studies have shown that the organization of tokens in time series models can substantially affect model behavior (Liu et al., 2024; Nie et al., 2022), and that different embedding strategies often exhibit heterogeneous performance across datasets (Zhao et al., 2026; Chen et al., 2025). In this work, we do not aim to re-establish this data dependence itself. Instead, we focus on a more fundamental mechanism: how the data view constructed by an embedding changes as feature interactions are performed in the encoder.

Data–Embedding dependence. The role of embedding can be understood as a bridge between data structure and the model's interaction mechanism. The data structure determines which patterns are worth organizing into tokens, such as local temporal continuity, channel redundancy, or variate-level dependencies (Zhao et al., 2026). The embedding determines how these patterns enter the model as a latent view. Therefore, the optimal embedding is not determined by the embedding form alone, but by the matching relationship among the data structure, the embedding-induced view, and the encoder mechanism.

Embedding tokens as the minimal units of interaction. From an interaction perspective, the encoder does not operate directly on raw observations, but on the latent tokens produced by the embedding module. Therefore, embedding defines the minimal semantic units that can participate in model interactions. Let

$$Z_m = [z_1, z_2, \ldots, z_{N_m}],$$

where Z_m denotes the latent tokens generated by the embedding strategy m. All subsequent interactions across time steps, local segments, or variates are performed among these tokens. Such token boundaries directly shape the inductive bias of the model. Point-wise tokens provide the finest temporal resolution, but each token contains limited contextual information. Patch-wise tokens aggregate adjacent time steps and naturally introduce local continuity and segment-level sharing. Variate-wise tokens treat the complete historical trajectory of each variable as a whole, making them more suitable for expressing variate-level dependencies. Thus, the essential role of embedding can be summarized as follows: it constructs different data geometries in the latent space through different tokenization strategies, and these geometries directly affect the effectiveness of subsequent high-order interactions.

Intrinsic linear folding structure in the first-order latent space. Given an embedding strategy m, the raw sequence X is mapped into a set of latent tokens: $Z_m = \mathcal{E}_m(X)$. Different embeddings do not merely reshape the input tensor, but construct different data views of the same sequence in the latent space. In the first-order limit without an encoder, the model degenerates into

$$\hat{Y} = \mathcal{D} \circ \mathcal{E}_m(X). \tag{4}$$

If both the embedding and the decoder are linear mappings, their composition remains linear. Therefore, regardless of the embedding strategy, the whole model can be folded into an equivalent affine mapping from the raw input to the forecasting output:

$$\hat{Y} = W_{\text{eff}}^{(m)} X + b, \tag{5}$$

where $W_{\text{eff}}^{(m)}$ is jointly induced by the parameters of the embedding and the decoder. In this case, although different embeddings construct different latent views, there is no further explicit interaction among latent tokens, and the decoder only performs a linear readout. As a result, the token boundaries, local neighborhoods, and variate organizations induced by the embedding are not fully exploited. The differences among embeddings are therefore reflected more in structured parameter

factorization, optimization dynamics, and implicit regularization, rather than in essential differences in the order of expressible interactions. A complete derivation is provided in Appendix A.

Complex interactions in the high-order latent space. Once a nonlinear encoder is introduced, the role of embedding changes. The model no longer directly performs a linear readout of Z_m, but instead conducts multi-layer representation evolution over the latent geometry defined by Z_m:

$$H^{(0)} = Z_m, \qquad H^{(k+1)} = \Phi_k(H^{(k)}), \quad k = 0, \ldots, K-1. \tag{6}$$

At this stage, embedding determines the objects, granularity, and initial geometry on which high-order interactions take place. The encoder determines how these tokens interact: MLPMixer applies learned global token mixing, Transformer uses input-dependent attention, GRU propagates recurrent states, state-space encoders evolve latent states, FFT auto-correlation aggregates token delays, and CNN mixes local token neighborhoods. A patch, time point, or variate therefore enters a different interaction structure under each encoder, leading to different representation evolution paths for the same embedding view.

Therefore, the central claim of this paper is that the effect of embedding cannot be judged in isolation. embedding does not independently improve model capacity; rather, it constructs a data view that must be compatible with subsequent high-order interactions. Only when the latent geometry induced by the embedding matches the interaction mechanism of the encoder can high-order interactions be effectively transformed into useful representation capacity.

5 Experiments and Analysis

The experimental setup, including the evaluation protocol, datasets, and encoder configurations, is detailed in Appendix B.2.

5.1 RQ1: Embedding in the First-Order Case

To evaluate the intrinsic performance differences among embedding strategies, we conduct an evaluatology-based assessment under a simplified architecture without any encoder. Specifically, we compare three embedding strategies, namely Patch-wise, Variate-wise, and Point-wise embeddings, against a simple Linear model without embedding, denoted as None.

Table 1: First-order performance of different embedding strategies across eight datasets. All metrics are lower-is-better. The best and second-best results within each dataset are highlighted in **red** and **blue**, respectively. Values marked with † are affected by OOM and are not directly comparable.

Dataset	Method	$\hat{\mu}\downarrow$	$\hat{\sigma}\downarrow$	$L_{\text{best}}\downarrow$	Dataset	Method	$\hat{\mu}\downarrow$	$\hat{\sigma}\downarrow$	$L_{\text{best}}\downarrow$
ECL	Patch	**0.171**	**0.029**	**0.141**	ETTm2	Patch	**0.173**	**0.009**	**0.164**
	Variate	0.180	0.034	0.141		Variate	**0.174**	0.009	**0.164**
	Point	**0.179**	**0.027**	0.142		Point	0.176	**0.009**	0.164
	None	0.188	0.050	**0.141**		None	0.176	0.010	0.164
ETTh1	Patch	**0.379**	**0.006**	**0.370**	Exchange	Patch	**0.085**	**0.004**	**0.082**
	Variate	**0.382**	0.011	**0.368**		Variate	0.088	**0.006**	0.082
	Point	0.384	**0.009**	0.372		Point	**0.088**	0.008	**0.082**
	None	0.391	0.022	0.371		None	0.092	0.009	0.082
ETTh2	Patch	**0.285**	**0.010**	**0.274**	Traffic	Patch	**0.534**	**0.121**	**0.410**
	Variate	**0.286**	**0.008**	**0.273**		Variate	**0.559**	**0.134**	0.411
	Point	0.291	0.013	0.274		Point	0.645†	0.000†	0.645†
	None	0.290	0.013	0.274		None	0.591	0.183	**0.410**
ETTm1	Patch	**0.325**	**0.025**	**0.300**	Weather	Patch	**0.184**	**0.010**	**0.174**
	Variate	**0.328**	0.026	0.301		Variate	**0.184**	0.010	0.174
	Point	0.328	**0.023**	0.300		Point	0.184	**0.009**	0.174
	None	0.331	0.029	**0.300**		None	0.185	0.011	**0.174**

Note. Rankings are determined using the original unrounded values. The Traffic-Point results marked with † are excluded from best/second-best highlighting because they are affected by OOM and are not directly comparable.

Limited first-order differences. As shown in Table 1, Patch-wise embedding achieves the lowest $\hat{\mu}$ on most datasets with relatively small $\hat{\sigma}$, indicating better accuracy and robustness in the first-order setting. Nevertheless, different embeddings can still obtain the best result on specific datasets, which explains why prior studies may report different SOTA embedding choices under particular configurations. Such case-specific superiority, however, does not necessarily generalize (Brigato

et al., 2025; Wang et al., 2026). The significance test further confirms that there is no significant difference among embeddings.(Details in Appendix A).

5.2 RQ2: Data Views Constructed by Embedding under High-Order Interactions

Embedding determines the objects over which an encoder interacts. We compare embedding views across six encoder families, with a no-encoder reference. Table 2 summarizes the preferred views and their configuration distributions on all eight datasets in this comparison. Detailed settings are given in Appendix D.4, and full view-level results in Appendix D.1.

Table 2: Preferred embedding views on eight datasets, selected by the lowest mean test MSE within each dataset–encoder group. For the selected view, $\hat{\mu}$, $\hat{\sigma}$, and L_{best} denote the mean, sample standard deviation, and minimum across configurations. Within each dataset and metric, **red** and **blue** mark the lowest and second-lowest values, ranked before rounding. Search spaces, filtering rules, and training budgets are given in Appendix D.4.

Encoder	View	$\hat{\mu}$	$\hat{\sigma}$	L_{best}	Encoder	View	$\hat{\mu}$	$\hat{\sigma}$	L_{best}
ECL					**ETTm2**				
No encoder	Patch	0.171	0.029	0.141	No encoder	Patch	**0.173**	0.009	**0.164**
MLPMixer	Patch+CI	0.163	0.022	**0.133**	MLPMixer	Variate+CD	0.180	**0.004**	0.171
Transformer	Variate+CD	**0.163**	0.022	0.137	Transformer	Patch+CI	0.177	0.007	**0.166**
GRU†	Point+CD	0.201	0.028	0.178	GRU	Patch+CI	0.178	0.007	0.166
SSM†	Variate+CD	0.182	0.030	0.145	SSM	Patch+CI	**0.177**	**0.006**	0.168
FFT†	Point+CD	0.184	**0.012**	0.175	FFT	Variate+CD	0.184	0.009	0.171
CNN‡	Variate+CD	**0.158**	**0.022**	**0.131**	CNN‡	Patch+CI	0.178	0.008	0.167
ETTh1					**Exchange**				
No encoder	Patch	**0.379**	**0.006**	**0.370**	No encoder	Patch	**0.085**	**0.004**	**0.082**
MLPMixer	Variate+CD	**0.392**	**0.020**	0.375	MLPMixer	Patch+CI	**0.091**	0.013	0.082
Transformer	Patch+CI	0.404	0.048	**0.375**	Transformer	Patch+CI	0.097	0.025	0.083
GRU	Variate+CD	0.433	0.054	0.382	GRU	Patch+CI	0.104	0.035	0.083
SSM	Variate+CD	0.408	0.041	0.377	SSM	Patch+CI	0.092	**0.009**	**0.082**
FFT	Variate+CD	0.431	0.065	0.379	FFT	Variate+CD	0.104	0.022	0.084
CNN‡	Variate+CD	0.416	0.044	0.389	CNN‡	Patch+CI	0.092	0.009	0.082
ETTh2					**Traffic**				
No encoder	Patch	**0.285**	**0.010**	**0.274**	No encoder	Patch	0.534	0.121	0.410
MLPMixer	Patch+CI	0.309	0.030	0.286	MLPMixer	Patch+CI	**0.467**	**0.062**	**0.382**
Transformer	Point+CI	0.308	0.031	**0.281**	Transformer	Variate+CD	0.469	**0.064**	0.395
GRU	Patch+CI	0.311	0.026	0.286	GRU†	Variate+CD	0.592	0.131	0.437
SSM	Patch+CI	**0.299**	**0.014**	0.288	SSM†	Variate+CD	0.529	0.090	0.414
FFT	Patch+CI	0.309	0.021	0.284	FFT†	Variate+CD	0.537	0.073	0.427
CNN‡	Patch+CI	0.303	0.020	0.286	CNN‡	Variate+CD	**0.463**	0.071	**0.382**
ETTm1					**Weather**				
No encoder	Patch	0.325	0.025	0.300	No encoder	Variate	0.184	**0.010**	0.174
MLPMixer	Variate+CD	**0.312**	**0.019**	**0.283**	MLPMixer	Variate+CD	**0.158**	**0.008**	0.150
Transformer	Patch+CI	0.320	0.025	0.289	Transformer	Variate+CD	0.166	0.014	**0.147**
GRU	Patch+CI	0.327	0.027	0.298	GRU	Variate+CD	0.170	0.014	0.149
SSM	Variate+CD	**0.318**	0.020	0.292	SSM	Variate+CD	**0.163**	0.011	0.148
FFT	Variate+CD	0.342	0.057	**0.287**	FFT	Variate+CD	0.167	0.015	**0.147**
CNN‡	Patch+CI	0.323	**0.018**	0.294	CNN‡	Variate+CD	0.165	0.013	0.149

† Five-epoch searches. ‡ At most ten epochs with seed 2021 and batch size 256 except for ECL and Traffic (32). GRU, SSM, and FFT counts appear in Table 22.

On Weather, Variate+CD has the lowest configuration mean for all six encoders. On ETTm1, MLPMixer, SSM, and FFT favor Variate+CD, whereas Transformer, GRU, and CNN favor Patch+CI. Thus, a preference can recur across distinct interaction mechanisms, but the preferred granularity also depends on the dataset and encoder.

On Traffic, five encoders favor Variate+CD, while MLPMixer favors Patch+CI. ETTh2 shows a different pattern: five encoders favor Patch+CI, while Transformer favors Point+CI. ECL spans all three granularities: MLPMixer favors patches, GRU and FFT favor points, and Transformer, SSM, and CNN favor variates. On Exchange, five encoders favor Patch+CI, while FFT favors Variate+CD.

View preferences also depend on the evaluation conditions and error criterion. The three-seed results at fixed configurations show changes from the configuration-averaged preferences (Appendix D.2), and mean performance and the best individual run can select different views (Table 13). Together, these results connect embedding performance to the interaction units it exposes and the mechanism that combines them. This motivates examining the geometry of the resulting representation and the forecasting information accessible to its readout.

5.3 RQ3: Rank Evolution and Predictive Readability

RQ2 shows that embedding effectiveness varies with the dataset and encoder. We study this dependence through spectral geometry, dataset characteristics, and the linear readability of forecasting targets. The rank analysis covers 116 configurations across eight datasets and five encoder families: MLPMixer, Transformer, GRU, Mamba, and FFT. Frozen linear probes evaluate 98 checkpoints spanning the same dataset and encoder families.

Effective rank. For a dataset s, we denote the encoder output of an embedding–encoder pair (m, ϕ) as $H_{m,\phi,s} \in \mathbb{R}^{N_s \times d_h}$, where N_s is the number of collected latent samples/tokens and d_h is the hidden dimension. Given the centered representation matrix H, we perform singular value decomposition (SVD) and denote $\lambda_i = \sigma_i^2$ and $p_i = \lambda_i / \sum_j \lambda_j$. Following the spectral-entropy-based effective rank (Roy & Vetterli, 2007), we define

$$r_{\text{eff}}(H) = \exp\left(-\sum_i p_i \log p_i\right). \tag{7}$$

Effective rank measures the concentration of representation energy across directions. Figure 2 compares encoder-output ranks across the 116 configurations. Appendix E.6 specifies the configuration coverage and representation-sampling protocols used in the rank analysis.

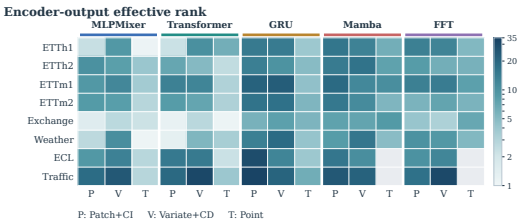


Figure 2: Encoder-output effective rank for 116 configurations across eight datasets and five encoder families, shown on a logarithmic color scale. Gray cells mark four unavailable configurations. Point uses CD for MLPMixer and Transformer on ETTh1, Weather, and Exchange, and CI elsewhere. FFT groups FFT-attention and FFT auto-correlation.

Encoder-dependent spectral geometry. Figure 2 shows that Point embeddings typically concentrate representation energy in fewer directions. Among the 36 dataset–encoder groups with all three embedding views, Patch+CI and Variate+CD have higher effective ranks than Point in 30 and 34 groups, respectively. Rank also varies substantially across encoders under the same view: on ECL, Patch+CI yields effective ranks of 28.09 for GRU, 14.89 for Transformer, and 8.91 for MLPMixer. These patterns characterize how the token view and encoder jointly redistribute representation energy.

Sensitivity to retained spectral directions. An SVD intervention evaluates 98 checkpoints across eight datasets and all five encoder families using the same 64 validation windows per dataset. With the decoder fixed, $68/98$ configurations achieve lower MSE at an evaluated truncation level than with all 128 directions retained (Appendix E.4).

Statistics complement rank. Figure 3 relates seven dataset descriptors to the configuration-averaged test MSE from RQ2 for MLPMixer, Transformer, GRU, diagonal SSM, FFT, and CNN. Each coefficient uses the same nine datasets, including Illness. Trend correlates negatively with mean MSE for all 18 encoder–embedding combinations, with coefficients from -0.80 to -0.67. Seasonality correlates positively with mean MSE across these combinations (0.52–0.72). The corresponding MAE results appear in Figure 9 in Appendix E.7. These recurring associations connect dataset structure to average forecasting difficulty across interaction families.

Linear probes of predictive readability. Following the idea of independently trained linear probes (Alain & Bengio, 2016), we freeze each checkpoint and fit separate ridge readouts to its embedding output E and encoder output H.

Each readout predicts the same 96 future values for a target channel from the full native feature vector. Each readout uses at most 1,024 training (window, channel) rows. We use identical sampled windows and channel indices, choose regularization on validation data, and evaluate on held-out test windows. Define

$$G_{\text{probe}} = \log \frac{\text{MSE}_E}{\text{MSE}_H}. \tag{8}$$

Positive G_{probe} means that a fitted linear readout predicts more accurately after the encoder. The collection spans eight datasets with Patch+CI, Variate+CD, and Point+CI; the full protocol and dataset-level results appear in Appendix E.6.

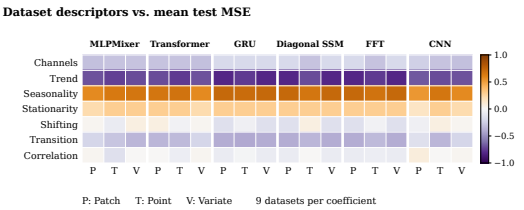


Figure 3: Spearman correlations between seven dataset descriptors and mean test MSE across nine datasets for six RQ2 encoder families. Each embedding-family mean pools its retained configuration runs and available CI/CD settings. Negative values associate larger descriptor values with lower error. The aggregation, dataset coverage, and corresponding MAE results are given in Appendix E.7.

Across the 98 configurations, 72 have positive probe gains. The geometric-mean error reduction is 13.14% on ECL and 20.07% on Traffic, while all nine Exchange configurations have negative gains, with a 3.60% geometric-mean error increase. The latter identifies a setting where the embedding-level representation already supports a stronger linear forecast under the probe protocol. Table 3 compares views on the same 30 complete dataset–encoder groups. Patch and Variate improve more consistently than Point, linking the gain to the interaction view rather than encoder presence alone. For Variate, the gain includes the integration of other channels into the target channel's representation.

Table 3: Paired probe comparison on 30 dataset–encoder groups containing all three views. Error reduction is $100(1 - \exp(-\bar{G}))$, with $\bar{G}$ averaged equally across datasets after averaging encoders within each dataset. Positive reduction favors the encoder output.

View	Configurations with $G_{\text{probe}} > 0$	Error reduction (%)
Patch+CI	27/30	+5.95
Variate+CD	23/30	+9.22
Point+CI	14/30	−0.04

Geometry and readability are complementary. Among 37 configurations whose effective rank decreases from E to H, 26 improve probe accuracy; among 61 whose rank increases, 46 improve. Thus, both spectral concentration and expansion can accompany more readable forecasting targets. Increasing the probe fitting budget from 1,024 to 2,048 rows preserves the gain sign in all 35 configurations with enough sampled training rows for the larger budget. Together with the dataset statistics, these results characterize embedding–encoder matching through both the geometry of the representation and the target information accessible to its readout.

5.4 RQ4: Module Selection from Multiscale Data Complexity

Data complexity guides module selection before target-model training. We map effective-rank statistics of the target training series to an embedding–encoder pair. The rule is calibrated on source datasets; subsequent target selection requires only spectral calculations, without candidate-model training or forward passes.

Baseline rank and cross-scale growth summarize input variation. Let $\bar{r}_L$ denote the mean effective rank of centered, flattened training windows of length L, using the energy-weighted definition in Section 5.3. We construct

$$R = \bar{r}_{96}, \qquad \Gamma = \frac{\bar{r}_{192}}{\bar{r}_{48}}. \tag{9}$$

Here, R measures the effective dimension of variation among windows, while Γ measures its relative growth with observation length. Neither statistic requires a candidate representation.

Source structural preferences determine the decision boundaries. The Rank-Guarded Data–Rank Selector (DRS-RG) supplies source labels from short-run validation performance and rank matching (Appendix F.1). We calibrate a growth threshold τ_Γ separating a Variate+MLPMixer branch from a Patch+CI branch. Within Patch+CI, boundaries t_0, t_1, t_2 in log-rank coordinates follow the empirical encoder order $(c_0, c_1, c_2, c_3) = (\text{None}, \text{GRU}, \text{Transformer}, \text{Mamba})$, where None is the identity encoder.

Table 4: DRS-RG selections and test errors with 96-step inputs and forecasts (seed 2024). The oracle minimizes test MSE over 18 fully trained candidates; its MAE comes from that same configuration. Regret is the relative MSE increase over the oracle.

		DRS-RG		Oracle		
Dataset	Selected structure	MSE	MAE	MSE	MAE	Regret (%)
ETTh1	Patch+CI / Mamba	0.373670	0.396009	0.373670	0.396009	0.0000
ETTh2	Patch+CI / Transformer	0.284035	0.338102	0.284035	0.338102	0.0000
ETTm1	Variate / MLPMixer	0.320872	0.359945	0.320872	0.359945	0.0000
ETTm2	Patch+CI / GRU	0.175478	0.261194	0.175478	0.261194	0.0000
Exchange	Patch+CI / None	0.082489	0.198576	0.082467	0.200818	0.0268
Weather	Variate / MLPMixer	0.160245	0.207264	0.160245	0.207264	0.0000

Cross-scale growth selects the branch, and baseline rank selects the Patch encoder. For a target profile (R, Γ), the calibrated rule returns

$$\widehat{a}(R,\Gamma) = \begin{cases} (\text{Variate}, \text{MLPMixer}), & \Gamma > \tau_\Gamma, \\ (\text{Patch+CI}, c_k), & \Gamma \le \tau_\Gamma, \end{cases} \qquad k = \sum_{j=0}^{2} \mathbf{1}[\log R \ge t_j]. \tag{10}$$

The rule recovers all six reference choices and remains stable under resampling. With 96-step inputs and predictions, leave-one-dataset-out calibration recovers all six DRS-RG structures in Table 4. Each fold fits thresholds on five source datasets with fixed features, branch assignments, and encoder order. Repeated sampling yields $599/600$ matches (99.83%), both when resampling source and target profiles with recalibration and when resampling only the target with fixed thresholds. The sole mismatch occurs on ETTh2 at the embedding boundary. Appendix F gives the calibration, sampling protocol, and analysis.

The selected configurations attain the lowest candidate MSE on five datasets. Table 4 compares their forecasting errors with the minimum-MSE candidate, giving a mean relative MSE regret of 0.0045%. On Exchange, the oracle uses Patch+CI / Mamba, while DRS-RG selects Patch+CI / None: its MSE is 0.0268% higher, but its MAE is lower (0.198576 versus 0.200818). Appendix F.2 provides the full candidate comparison and three-seed results.

6 Discussion

Practical implications. Embedding views define the objects available for interaction: patches, time points, or variates. Encoders change their spectral geometry and target readability, characterized by effective rank and linear probes, respectively. The selection rule uses input statistics to choose a structure before target-model training; probes assess the trained representation.

Scope and next steps. The probe and selection evaluations use 96-step forecasting. Selection has five possible outputs, with branch assignments and encoder order established from six source datasets; agreement measures recovery of those preferences. Future work should expand the candidate set, study horizon-dependent boundaries, and test multi-view readouts that retain complementary temporal and cross-variate information.

7 Conclusion

Embedding effects depend on the interactions applied to time series. Without an encoder, embeddings largely reduce to linear mappings, with performance gaps mainly arising from parameter sharing and optimization-induced biases.

With deep encoders, embedding performance depends on both the dataset and architecture. Effective rank and frozen linear probes characterize spectral geometry and target readability, respectively. A calibrated multiscale rank rule connects this analysis to module selection before target-model training.

AI USE STATEMENT

We used large language models (LLMs) to assist with data organization, code generation, and the translation and language editing of the manuscript. We take responsibility for the final content of this work, including material produced with LLM assistance.

REFERENCES

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

Xiang Ao. Research on the stock price prediction model based on large-scale transactions. In *Proceedings of the 2025 8th International Conference on Computer Information Science and Artificial Intelligence*, pp. 392–396, 2025.

Julio Barzola-Monteses, Juan Gomez-Romero, Mayken Espinoza-Andaluz, and Waldo Fajardo. Time series forecasting techniques applied to hydroelectric generation systems. *International Journal of Electrical Power & Energy Systems*, 164:110424, 2025.

Lorenzo Brigato, Rafael Morand, Knut Strømmen, Maria Panagiotou, Markus Schmidt, and Stavroula Mougiakakou. There are no champions in supervised long-term time series forecasting. *arXiv preprint arXiv:2502.14045*, 2025.

Yu Chen, Nathalia Céspedes, and Payam Barnaghi. A closer look at transformers for time series forecasting: Understanding why they work and where they struggle. In *Forty-second international conference on machine learning*, 2025.

Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

Lu Han, Han-Jia Ye, and De-Chuan Zhan. The capacity and robustness trade-off: Revisiting the channel independent strategy for multivariate time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 36(11):7129–7142, 2024.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Zhe Li, Shiyi Qi, Yiduo Li, and Zenglin Xu. Revisiting long-term time series forecasting: an investigation on affine mapping. *Academia AI and Applications*, 2(2), 2026.

Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1754–1763, 2018.

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 *International conference on learning representations*, volume 2024, pp. 11116–11140, 2024.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.

Hussain Nizam, Samra Zafar, Zefeng Lv, Fan Wang, and Xiaopeng Hu. Real-time deep anomaly detection framework for multivariate time-series data in industrial iot. *IEEE Sensors Journal*, 22 (23):22836–22849, 2022.

Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *2007 15th European signal processing conference*, pp. 606–610. IEEE, 2007.

Nikolaos Schetakis, Paolo Bonfini, Negin Alisoltani, Konstantinos Blazakis, Symeon I Tsintzos, Alexis Askitopoulos, Davit Aghamalyan, Panagiotis Fafoutellis, and Eleni I Vlahogianni. Quantum neural networks with data re-uploading for urban traffic time series forecasting. *Scientific Reports*, 15(1):19400, 2025.

Peiwang Tang and Weitai Zhang. Unlocking the power of patch: Patch-based mlp for long-term time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 12640–12648, 2025.

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.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Zhang, and Jun Zhou. Timemixer: Decomposable multiscale mixing for time series forecasting. In *International conference on learning representations*, volume 2024, pp. 38626–38652, 2024a.

Xiaorui Wang, Fanda Fan, Chenxi Wang, Yuxuan Yang, Rui Tang, Kuoyu Gao, Simiao Pang, Yuanfeng Shang, Zhipeng Liu, Wanling Gao, et al. Combinationts: A modular framework for understanding time-series forecasting models. *arXiv preprint arXiv:2605.01231*, 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, 2024b.

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.

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*, volume 37, pp. 11121–11128, 2023.

Jianfeng Zhan, Lei Wang, Wanling Gao, Hongxiao Li, Chenxi Wang, Yunyou Huang, Yatao Li, Zhengxin Yang, Guoxin Kang, Chunjie Luo, Hainan Ye, Shaopeng Dai, and Zhifei Zhang. Evaluatology: The science and engineering of evaluation. *BenchCouncil Transactions on Benchmarks, Standards and Evaluations*, 4(1):100162, 2024. ISSN 2772-4859. doi: https://doi.org/10.1016/j.tbench.2024.100162. URL https://www.sciencedirect.com/science/article/pii/S2772485924000140.

Zhiyuan Zhao, Juntong Ni, Shangqing Xu, Haoxin Liu, Wei Jin, and B Aditya Prakash. Timerecipe: A time-series forecasting recipe via benchmarking module level effectiveness. In *International Conference on Learning Representations*, volume 2026, pp. 150745–150792, 2026.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11106–11115, 2021.

A Analysis of Embedding Degeneration in First-Order Interactions

A.1 Structural Consistency

To analyze the source of differences among several Embedding strategies, we abstract them into their mathematical forms and examine their essential functional structures. Consider a simplified time series forecasting task where the input sequence is

$$X = [x_1, x_2, x_3, x_4]^\top \in \mathbb{R}^{4\times 1},$$

and the prediction target is a scalar $y \in \mathbb{R}$. Let the hidden dimension be $d_{\text{model}} = 3$, and assume that the Decoder is a linear layer. According to the model design, the Decoder directly maps the flattened latent features to the output. Therefore, its weights are strictly not shared along the sequence or patch dimension.

None: direct linear mapping. Without explicit Embedding, the input is directly mapped to the output by a linear layer:

$$y = \mathbf{w}_{\text{None}}^{\top} X + b, \tag{11}$$

where $\mathbf{w}_{\text{None}} \in \mathbb{R}^{4\times 1}$ is the linear weight vector and b is the bias term.

Point-wise Embedding: scalar-level shared projection. Each time step is independently projected into the hidden space using a shared projection vector $\mathbf{v} \in \mathbb{R}^{3\times 1}$, producing four tokens. After flattening, the feature dimension becomes 12. The Decoder is a $12 \to 1$ linear layer with weight

$$\mathbf{w}_{\text{dec}} = [\mathbf{w}_{d1}^{\top}, \mathbf{w}_{d2}^{\top}, \mathbf{w}_{d3}^{\top}, \mathbf{w}_{d4}^{\top}]^{\top} \in \mathbb{R}^{12\times 1}.$$

The output can be written as

$$y = \sum_{i=1}^{4} \mathbf{w}_{di}^{\top}(\mathbf{v}x_i) + b = \begin{bmatrix} \mathbf{v}^{\top}\mathbf{w}_{d1} \\ \mathbf{v}^{\top}\mathbf{w}_{d2} \\ \mathbf{v}^{\top}\mathbf{w}_{d3} \\ \mathbf{v}^{\top}\mathbf{w}_{d4} \end{bmatrix}^{\top} X + b. \tag{12}$$

Patch-wise Embedding: block-level shared projection. The sequence is divided into two patches, each with patch length 2. The Embedding weight $V \in \mathbb{R}^{2\times 3}$ is shared across patches. After flattening, the feature dimension becomes $2 \times 3 = 6$. The Decoder is a $6 \to 1$ linear layer with weight

$$\mathbf{w}_{\text{dec}} = [\mathbf{w}_{d1}^{\top}, \mathbf{w}_{d2}^{\top}]^{\top} \in \mathbb{R}^{6\times 1}.$$

The output is

$$\begin{aligned} y &= \mathbf{w}_{d1}^{\top}\left(V^{\top}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) + \mathbf{w}_{d2}^{\top}\left(V^{\top}\begin{bmatrix} x_3 \\ x_4 \end{bmatrix}\right) + b \\ &= \begin{bmatrix} V\mathbf{w}_{d1} \\ V\mathbf{w}_{d2} \end{bmatrix}^{\top} X + b. \end{aligned} \tag{13}$$

Variate-wise Embedding: global sequence projection. The entire sequence X is mapped into the hidden space at once through $V \in \mathbb{R}^{4\times 3}$, producing a global feature vector of dimension 3. The Decoder is a $3 \to 1$ linear layer with weight $\mathbf{w}_{\text{dec}} \in \mathbb{R}^{3\times 1}$. The output becomes

$$y = \mathbf{w}_{\text{dec}}^{\top}(V^{\top}X) + b = (V\mathbf{w}_{\text{dec}})^{\top}X + b. \tag{14}$$

The above derivation shows that although different Embedding strategies introduce different degrees of parameter sharing and thus impose different inductive biases, their final outputs after the non-shared Decoder are all linear functions of the original input X. Therefore, from the perspective of model structure and expressive power, these four strategies are strictly equivalent to a linear regression model, since they can all be reduced to the form

$$y = \mathbf{w}_{\text{eff}}^{\top}X + b. \tag{15}$$

A.2 Dynamical Inconsistency

If these model structures are functionally equivalent to linear regression, why do their empirical performance and variance still differ? The reason lies in the different optimization dynamics induced by parameter factorization and the granularity of parameter sharing.

Although the forward mapping can always be folded into an equivalent global weight vector $\mathbf{w}_{\text{eff}}$, backpropagation updates the original factorized parameters directly. Therefore, different parameterizations guide optimization along different trajectories in the loss landscape and induce different forms of implicit regularization.

Let the loss function be $\mathcal{L}$. According to the chain rule,

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}} = \frac{\partial \mathcal{L}}{\partial y}\frac{\partial y}{\partial \mathbf{W}}.$$

We analyze the optimization dynamics of the four strategies as follows.

None: unconstrained global update. Without Embedding, the parameter update corresponds to standard convex optimization under the MSE loss. The gradient directly acts on the global equivalent weight without structural constraints:

$$\Delta \mathbf{w}_{\text{None}} \propto -\nabla_{\mathbf{w}_{\text{None}}} \mathcal{L} = -\frac{\partial \mathcal{L}}{\partial y} X. \tag{16}$$

This formulation has the highest degree of freedom, but it is also more prone to overfitting noisy patterns.

Variate-wise Embedding: deep factorization without temporal sharing. In Variate-wise Embedding, the whole sequence is mapped into the hidden space at once. Both $V \in \mathbb{R}^{L\times d}$ and $\mathbf{w}_{\text{dec}} \in \mathbb{R}^{d\times 1}$ do not involve parameter sharing along the temporal dimension. The gradients are

$$\frac{\partial \mathcal{L}}{\partial V} = \frac{\partial \mathcal{L}}{\partial y} X \mathbf{w}_{\text{dec}}^{\top}, \qquad \frac{\partial \mathcal{L}}{\partial \mathbf{w}_{\text{dec}}} = \frac{\partial \mathcal{L}}{\partial y} V^{\top} X. \tag{17}$$

Although the matrix factorization introduces a form of low-rank regularization, the parameters in V are tied to absolute input positions. As a result, this formulation cannot naturally capture local translation-invariant structures, which are often important in time series data.

Point-wise Embedding: weak scalar-level sharing. Point-wise Embedding shares the projection vector $\mathbf{v}$ across all time steps, while the Decoder weights $\mathbf{w}_{di}$ remain position-specific. The gradient of the shared parameter $\mathbf{v}$ is an aggregation of weighted contributions from all time steps:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{v}} = \sum_{i=1}^{L} \frac{\partial \mathcal{L}}{\partial \mathbf{w}_{\text{eff},i}} \mathbf{w}_{di} = \frac{\partial \mathcal{L}}{\partial y} \sum_{i=1}^{L} x_i \mathbf{w}_{di}. \tag{18}$$

Although this appears to introduce parameter aggregation, the sharing scope is limited to a single scalar observation. Therefore, $\mathbf{v}$ cannot encode structural relations among neighboring time steps, such as local trends or periodic patterns. The representational burden is largely shifted to the independent Decoder weights $\mathbf{w}_{di}$. During backpropagation, the non-shared updates of these Decoder weights dominate the optimization process, making Point-wise Embedding weakly coupled and insufficient to impose an effective structural constraint.

Patch-wise Embedding: strong block-level sharing. Patch-wise Embedding shares the Embedding parameter $V \in \mathbb{R}^{P\times d}$ across different patches of length P, while the Decoder weights remain patch-specific. By the chain rule, the gradient of V aggregates independent feedback from different patches:

$$\frac{\partial \mathcal{L}}{\partial V} = \sum_{c=1}^{N} \frac{\partial \mathcal{L}}{\partial \mathbf{w}_{\text{eff},c}} \mathbf{w}_{dc}^{\top} = \frac{\partial \mathcal{L}}{\partial y} \sum_{c=1}^{N} X_c \mathbf{w}_{dc}^{\top}, \tag{19}$$

where X_c denotes the c-th patch and N denotes the number of patches.

Unlike Point-wise Embedding, each patch input X_c is a vector containing local contextual information. This asymmetric design, namely shared local-structure extraction combined with position-specific global readout, strongly constrains the unordered updates of the low-level feature extractor and acts as a strong inductive bias. It encourages the matrix V to learn reusable local motifs across the entire sequence.

In summary, the block-level gradient aggregation mechanism of Patch-wise Embedding imposes a stronger inductive bias, restricts the effective hypothesis space, and leads to smoother and more robust optimization trajectories.

B Experimental Details

B.1 Model Implementation Details

All models follow the modular Embedding–Encoder–Decoder pipeline. Given an input sequence $X \in \mathbb{R}^{B\times L\times C}$, where B denotes the batch size, L denotes the input length, and C denotes the number of variables, the model outputs $\hat{Y} \in \mathbb{R}^{B\times H\times C}$, where H is the forecasting horizon. To reduce the influence of scale differences across datasets, we apply RevIN normalization (Li et al., 2026) at the input side and perform the corresponding denormalization at the output side.

Embedding. We implement four types of input views. For the setting without explicit Embedding, the model directly uses a linear layer to map the input length L to the forecasting horizon H. For Point-wise Embedding, under the Channel-Dependent mode, the multivariate observation at each time step is mapped into one token through

$$\mathrm{Linear}(C, d_{\mathrm{model}}).$$

Under the Channel-Independent mode, each variable is processed independently, and each scalar observation is mapped into the latent space through

$$\mathrm{Linear}(1, d_{\mathrm{model}}).$$

Both Point-wise Embedding variants are equipped with sinusoidal positional Embedding (Vaswani et al., 2017).

For Patch-wise Embedding, the model first applies replication padding along the temporal dimension and then constructs patches using a sliding window with patch length P and stride s. Each patch is mapped into a latent token through

$$\mathrm{Linear}(P, d_{\mathrm{model}}),$$

followed by positional Embedding. Since the padding size is set to the stride in our implementation, the number of patch tokens for a single variable is

$$N_{\mathrm{patch}} = \left\lfloor \frac{L + s - P}{s} \right\rfloor + 1.$$

In the Channel-Independent mode, different variables are flattened into the batch dimension and modeled independently. In the Channel-Dependent mode, patch tokens from different variables are concatenated into the same token sequence.

For Variate-wise Embedding, the input is rearranged as

$$X \in \mathbb{R}^{B \times C \times L},$$

and the complete historical sequence of each variable is mapped into a variate-level token through

$$\mathrm{Linear}(L, d_{\mathrm{model}}).$$

Therefore, the number of tokens under Variate-wise Embedding is equal to the number of variables C.

Encoder. The forecasting comparison uses MLPMixer, Transformer, GRU, diagonal SSM, FFT auto-correlation, and CNN encoders, with No Encoder as the identity setting that feeds the embedding output directly to the decoder. These forecasting encoders mix embedding tokens through learned token weights, attention, recurrence, state-space processing, delay aggregation, and local convolution, respectively. Representation diagnostics use MLPMixer, Transformer, GRU, Mamba, and FFT.

MLPMixer contains stacked Mixer layers, each with a token-mixing MLP and a channel-mixing MLP in a PreNorm residual structure. Token mixing acts along the token dimension, while channel mixing acts along the hidden dimension. Both MLP expansion factors are 4, with GELU activation and dropout.

Transformer contains stacked self-attention layers with multi-head self-attention, a feed-forward network, LayerNorm, and residual connections. In the RQ2 configuration sweeps, its feed-forward dimension and head count are

$$d_{\mathrm{ff}} = 4d_{\mathrm{model}}, \qquad n_{\mathrm{heads}} = \frac{d_{\mathrm{model}}}{32}.$$

Thus, when $d_{\mathrm{model}} \in \{64, 128, 256\}$, the corresponding numbers of attention heads are $2, 4, 8$, respectively. The fixed-window representation diagnostics use the settings stored with each checkpoint; their Transformer checkpoints use $d_{\mathrm{model}} = 128$, $d_{\mathrm{ff}} = 2048$, and 8 heads.

GRU propagates recurrent states along the embedding-token sequence. The real-valued diagonal SSM implements causal state-space convolution with 16 states per hidden coordinate. FFT auto-correlation estimates token delays and aggregates shifted token representations. It is distinct from the frequency-attention encoder used in the representation diagnostics.

The CNN encoder contains two same-width one-dimensional convolutions with kernel size 3 and padding 1. A LayerNorm precedes the convolutions; GELU and dropout follow the first convolution. After dropout, a residual connection adds the input to the second convolution's output, followed by an output LayerNorm. The convolution axis is the embedding-token axis. The depth is fixed at two convolutional layers throughout the sweep.

The Mamba and FFT implementations used for representation diagnostics are specified with the checkpoint collection in Appendix E.6.

Decoder. The Decoder adopts different linear prediction heads according to the Embedding view. For the None setting, the model directly uses

$$\mathrm{Linear}(L, H)$$

to map the historical window to the forecasting window. For Variate-wise Embedding, the Decoder uses

$$\mathrm{Linear}(d_{\mathrm{model}}, H)$$

to independently generate future predictions for each variate-level token, and finally transposes the output back to

$$\mathbb{R}^{B\times H\times C}.$$

For Point-wise and Patch-wise Embedding, the model first flattens the token dimension and the hidden dimension. Under the Channel-Independent mode, it uses

$$\mathrm{Linear}(N_m d_{\mathrm{model}}, H)$$

to predict each variable independently, where N_m denotes the number of tokens under the corresponding Embedding. Under the Channel-Dependent mode, it uses

$$\mathrm{Linear}(N_m d_{\mathrm{model}}, HC)$$

to predict all variables simultaneously and then reshapes the output into

$$\mathbb{R}^{B\times H\times C}.$$

Routing of Channel-Independent and Channel-Dependent Modes. In the Channel-Independent mode, the model merges the variable dimension into the batch dimension, so that each variable is treated as an independent sample before being fed into the Embedding and Encoder. After prediction, the output is reshaped back to the multivariate form. In the Channel-Dependent mode, all variables are jointly fed into the model, allowing the Embedding and Encoder to directly model cross-variate interactions. This design enables Point-wise and Patch-wise Embeddings to support both CI and CD interaction modes, while Variate-wise Embedding naturally takes variables as tokens and therefore corresponds to a variate-level interaction view by default.

B.2 Experimental Setup

Probabilistic Evaluation Protocol. We adopt a probabilistic evaluation protocol inspired by Evaluatology (Zhan et al., 2024). For each dataset and encoder, embedding strategies are the evaluated objects (EOs), and hyperparameter settings such as learning rate, input length, and hidden dimension are evaluation conditions (ECs). Configuration sweeps share candidate hyperparameter axes, while architecture-specific settings and run coverage determine the evaluated conditions for each embedding strategy. Let $\mathcal{L}(m, c)$ denote the forecasting error of embedding m under condition c. After protocol-specific filtering, let $\{c_{m,k}\}_{k=1}^{K_m}$ be its retained configurations. We summarize their performance by

$$\hat{\mu}(m) = \frac{1}{K_m}\sum_{k=1}^{K_m}\mathcal{L}(m, c_{m,k}), \quad \hat{\sigma}(m) = \sqrt{\frac{1}{K_m - 1}\sum_{k=1}^{K_m}\left(\mathcal{L}(m, c_{m,k}) - \hat{\mu}(m)\right)^2}, \tag{20}$$

where $\hat{\mu}$ measures average error over the retained configurations and $\hat{\sigma}$ measures sensitivity to changes in evaluation conditions. We also report the best observed error, $L_{\mathrm{best}}(m) =$

Table 5: Domain information, split ratios, and data-side statistical indicators of the experimental datasets. Here, C denotes the number of variables and T denotes the sequence length. The ETT series adopts a 12/4/4-month chronological split, while the other datasets use a $7:1:2$ split. The statistical indicators follow the data-characteristic taxonomy used in TimeRecipe (Zhao et al., 2026).

Metric	**ETTh1**	**ETTh2**	**ETTm1**	**ETTm2**	**ECL**	**Exch.**	**Traffic**	**Weather**	**Illness**
Domain	Transf. (Hourly)	Transf. (Hourly)	Transf. (15-min)	Transf. (15-min)	Elec. Consum.	Exch. Rates	Road Occup.	Meteo. Obs.	Influenza (Weekly)
Split	12/4/4 months	12/4/4 months	12/4/4 months	12/4/4 months	7:1:2	7:1:2	7:1:2	7:1:2	7:1:2
$\boldsymbol{C}$	7	7	7	7	321	8	862	21	7
$\boldsymbol{T}$	17420	17420	69680	69680	26304	7588	17544	52696	966
Trend	0.733	0.928	0.777	0.777	0.667	0.998	0.099	0.797	0.476
Seasonality	0.518	0.362	0.422	0.454	0.645	0.118	0.678	0.348	0.595
Stationarity	1.000	1.000	1.000	1.000	1.000	0.125	1.000	0.900	0.714
Shifting	0.641	0.832	0.604	0.840	0.611	0.704	0.627	0.697	0.834
Transition	0.129	0.151	0.164	0.147	0.106	0.211	0.122	0.207	0.169
Correlation	0.115	0.111	-0.280	-0.285	-0.334	-0.255	-0.203	-0.196	-0.348

$\min_{1\le k\le K_m} \mathcal{L}(m, c_{m,k})$, which distinguishes low average error from strong performance at a particular configuration.

Datasets. We use nine TSF benchmarks: ETTh1, ETTh2, ETTm1, ETTm2, ECL, Exchange, Traffic, Weather, and Illness. Table 5 describes all nine datasets used in the view-level configuration distributions and dataset-descriptor analysis. The rank analysis covers the eight datasets other than Illness. Except for the ETT series, all datasets use a chronological $7:1:2$ train/validation/test split. The ETT datasets use the first 12 months for training, the following 4 months for validation, and the final 4 months for testing (Wu et al., 2021). The data-side indicators follow the characterization protocol in TimeRecipe (Zhao et al., 2026), covering trend, seasonality, stationarity, shifting, transition, and correlation.

Encoders and evaluation settings. We compare MLPMixer, Transformer, GRU, diagonal SSM, FFT auto-correlation, and a two-layer CNN in the Embedding–Encoder–Decoder framework, with an identity encoder as the first-order baseline. Configuration sweeps measure sensitivity to hyperparameter choices. The sweep axes are

$$\Omega = \begin{cases} \eta \in \{\,0.05,\ 0.01,\ 0.005,\ 0.001,\ 0.0005,\ 0.0001\,\}, \\ d_{\text{model}} \in \{\,64,\ 128,\ 256\,\}, \\ L_{\text{in}} \in \{\,96,\ 336\,\}. \end{cases}$$

Here, η, d_{model}, and L_{in} denote the learning rate, hidden dimension, and input length. Batch sizes are 32 or 256, depending on the experimental setting and memory requirements. Patch lengths, encoder depths, and attention heads are specified in Appendix B.1, and sweep settings and configuration counts in Appendix D.4. The identity, MLPMixer, and Transformer distributional runs use MAE training loss on an NVIDIA RTX 5090 GPU with 32GB memory. GRU, diagonal SSM, and FFT configuration sweeps use seed 2021, with a five-epoch budget on ECL and Traffic. The CNN sweep uses MSE training loss, seed 2021, at most ten epochs, and prediction length 96.

Representation diagnostics and module selection. Representation diagnostics examine MLPMixer, Transformer, GRU, Mamba, and FFT checkpoints through spectral rank and frozen linear probes (Appendix E.6). Mamba supplies the selective state-space representation in this analysis, while the forecasting comparison uses a diagonal SSM. The module-selection evaluation calibrates a data-based rule on six datasets.

C SUPPORT ANALYSIS OF RQ1

C.1 DISTRIBUTIONAL SIGNIFICANCE TEST

To further examine whether the performance differences among different Embedding strategies are statistically reliable in the first-order setting, we conduct non-parametric significance tests on the error distributions of each dataset separately. Unlike rank-based tests across datasets, this analysis directly uses the error samples obtained under different evaluation conditions within the same dataset, making it possible to determine whether the performance distributions of different Embedding strategies are significantly separated under a fixed data structure. Since the error distributions under different hyperparameter configurations may not satisfy the normality assumption, we adopt the Kruskal–Wallis test as the per-dataset omnibus test, followed by pairwise comparisons when necessary. The pairwise comparisons are conducted using the Mann–Whitney U test with Holm correction for multiple comparisons. In addition, we report Cliff's delta as the effect size to measure the practical magnitude of significant differences. The significance level is set to $\alpha = 0.05$.

Table 6 presents the Kruskal–Wallis omnibus test results for the four first-order Embedding strategies on each dataset.

Table 6: Per-dataset Kruskal–Wallis omnibus test results for different Embedding strategies in the first-order setting. $p < 0.05$ indicates that at least one pair of Embedding strategies has significantly different error distributions on the corresponding dataset.

Dataset	**H statistic**	p**-value**	**Conclusion**
ECL	2.8022	0.4231	n.s.
ETTh1	5.9591	0.1136	n.s.
ETTh2	4.1439	0.2463	n.s.
ETTm1	2.1758	0.5367	n.s.
ETTm2	3.1277	0.3723	n.s.
Exchange	15.4881	0.0014	Significant
Traffic	3.3540	0.3402	n.s.
Weather	0.2594	0.9675	n.s.

As shown in Table 6, among the eight datasets, only Exchange passes the omnibus significance test, while the remaining seven datasets do not reach statistical significance. This indicates that, in the first-order setting without an Encoder, the error distributions of different Embedding strategies usually do not exhibit stable and significant separation. Although the main results show that Patch-wise Embedding achieves lower average errors on multiple datasets, such advantages are insufficient to support strong significance conclusions on most datasets from the perspective of per-dataset distributional testing.

For the Exchange dataset, which passes the omnibus test, we further conduct Holm-corrected pairwise comparisons. Table 7 reports only the comparisons that remain significant after correction. Since lower forecasting error indicates better performance, a negative Cliff's delta means that the first method tends to obtain lower errors than the second method.

Table 7: Significant pairwise comparisons among first-order Embedding strategies on the Exchange dataset. p_{Holm} denotes the Holm-corrected p-value, and δ denotes Cliff's delta effect size.

Comparison	$\hat{\mu}_a$	$\hat{\mu}_b$	p_{Holm}	δ	**Effect**
Patch vs Variate	0.0849	0.0884	0.0466	-0.4545	Medium
Patch vs None	0.0849	0.0923	0.0019	-0.6219	Large

Table 7 shows that Patch-wise Embedding significantly outperforms Variate-wise and None Embedding on the Exchange dataset. The effect size is medium for Patch-wise versus Variate-wise and large for Patch-wise versus None, suggesting that the segment-level structure of Patch-wise Embed-

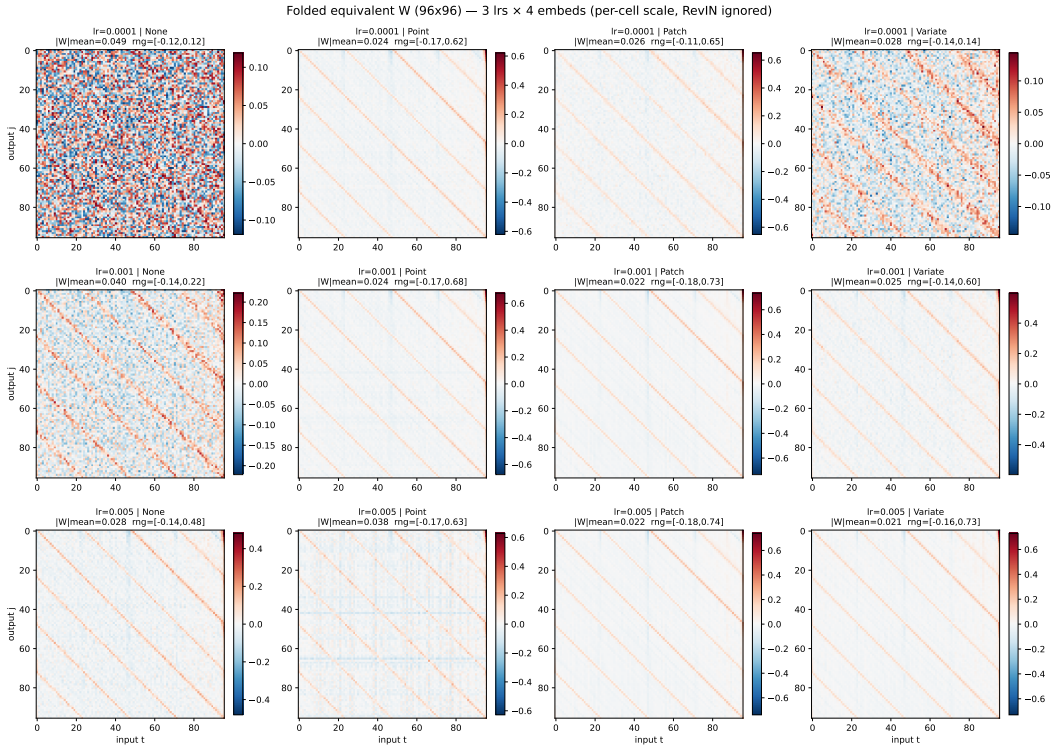


Figure 4: Visualization of folded equivalent weights in the first-order setting. For each model, the weights of all linear components are multiplied and folded into an equivalent input-to-output linear mapping. The highly similar folded weights across different Embedding strategies indicate that, without an Encoder, these models are functionally reducible to linear mappings. The remaining differences mainly reflect optimization dynamics and implicit regularization induced by different parameter-sharing structures.

ding indeed brings a noticeable improvement in the error distribution on this dataset. However, the difference between Patch-wise and Point-wise Embedding does not remain significant after Holm correction, indicating that even on Exchange, the advantage of Patch-wise Embedding is not uniformly significant over all other Embedding strategies.

Overall, this significance analysis further supports the central conclusion in the first-order setting: Embedding does not generally bring an essential increase in the order of expressible interactions when the Encoder is removed. The performance differences among different Embedding strategies are mostly dataset-dependent marginal gains, rather than stable and significant advantages across datasets and configurations. The statistically significant gains of Patch-wise Embedding on specific datasets, especially Exchange, are more reasonably attributed to structured parameter factorization, local sharing mechanisms, and the resulting optimization bias or implicit regularization. Since the composition of a linear Embedding and a linear Decoder can still be folded into an equivalent linear mapping, these significance results should not be interpreted as direct evidence of enhanced higher-order representation capacity.

C.2 Empirical Analysis

We further analyze the trained weights of the three Embedding-based models without an Encoder, together with the model without explicit Embedding, under three different learning rates. As shown in Figure 4, we multiply the weights of all linear layers in each model and fold them into an equivalent input-to-output mapping. Except for a few irregular failure cases, the resulting folded weights are highly similar across different Embedding strategies. This observation empirically supports the above theoretical conclusion: in the first-order setting, different Embedding strategies can be reduced to equivalent linear mappings, and their performance differences mainly arise from

parameterization-induced optimization dynamics rather than from fundamentally different expressive power.

D Support Analysis of RQ2

D.1 Configuration Distribution Results

Table 2 in the main text summarizes the preferred views on eight datasets under all six encoders and the no-encoder reference.

Tables 8, 9, 10, 11, 12, and 13 report the configuration distributions for all six encoders on nine datasets, including Illness. For every observed view, they give the configuration count and the mean, sample standard deviation, and minimum of both MSE and MAE. Point+CI on ECL and Traffic is included in the GRU, diagonal SSM, and FFT tables. Table 14 reports Point+CI configurations evaluated with the encoder and prediction head computed in FP32. The preferred-view summary in Table 2 selects the lowest mean MSE within each dataset–encoder sweep in Tables 8–13. Encoder architectures are specified in Appendix B.1, and search spaces, training budgets, filtering, and run coverage in Appendix D.4. Fixed-configuration results and variation across random seeds are reported in Appendix D.2.

Mean performance and the best individual run can also select different views. In the CNN sweep, Variate+CD has the lowest mean MSE on ETTh1, but Patch+CI attains the lowest individual MSE. On Illness, Point+CD achieves the lowest individual MSE despite a higher mean and larger dispersion than Variate+CD.

The configuration average also depends on the high-error tail. Before filtering, Variate+CD has the lowest mean MSE across completed CNN runs on all nine datasets. Filtering changes the mean-based preference on ETTh2, ETTm1, ETTm2, and Exchange. The filtering rule and retained counts are specified in Appendix D.4.

Table 8: Configuration distributions for the MLPMixer encoder on nine datasets. N is the retained configuration count. For MSE and MAE, we report the mean $\hat{\mu}$, sample standard deviation $\hat{\sigma}$, and minimum L_{best}. Bold marks the lowest mean and minimum separately within each dataset and metric. A dash indicates an unavailable result or an undefined standard deviation for $N = 1$. Search and filtering conditions are specified in Appendix D.4.

Dataset	View	N	MSE			MAE		
			$\hat{\mu}$	$\hat{\sigma}$	L_{best}	$\hat{\mu}$	$\hat{\sigma}$	L_{best}
ETTh1	Patch+CI	24	0.3962	0.0301	0.3767	**0.4079**	0.0219	**0.3935**
	Patch+CD	20	0.5053	0.0697	0.4150	0.4774	0.0444	0.4195
	Point+CI	14	0.4096	0.0372	0.3832	0.4124	0.0264	0.3939
	Point+CD	18	0.5014	0.0733	0.4198	0.4746	0.0450	0.4218
	Variate+CD	28	**0.3915**	0.0199	**0.3749**	0.4117	0.0146	0.3966
ETTh2	Patch+CI	23	**0.3087**	0.0299	**0.2860**	**0.3575**	0.0215	0.3403
	Patch+CD	18	0.3830	0.0301	0.3414	0.4063	0.0222	0.3809
	Point+CI	20	0.3391	0.0423	0.2907	0.3799	0.0299	**0.3402**
	Point+CD	18	0.3586	0.0163	0.3350	0.3913	0.0134	0.3783
	Variate+CD	30	0.3175	0.0223	0.2961	0.3666	0.0187	0.3472
ETTm1	Patch+CI	21	0.3235	0.0290	0.2896	0.3639	0.0177	0.3429
	Patch+CD	17	0.3851	0.0447	0.3355	0.4060	0.0262	0.3748
	Point+CI	19	0.4465	0.2698	0.2922	0.4277	0.1216	0.3453
	Point+CD	20	0.5389	0.4092	0.3208	0.4600	0.1395	0.3689
	Variate+CD	24	**0.3116**	0.0189	**0.2831**	**0.3563**	0.0094	**0.3429**
ETTm2	Patch+CI	32	0.1815	0.0099	**0.1695**	0.2676	0.0100	**0.2570**
	Patch+CD	26	0.2004	0.0118	0.1787	0.2820	0.0112	0.2607
	Point+CI	20	0.1872	0.0102	0.1779	0.2748	0.0103	0.2665
	Point+CD	20	0.1945	0.0095	0.1804	0.2769	0.0086	0.2632
	Variate+CD	28	**0.1795**	0.0041	0.1714	**0.2649**	0.0032	0.2603
Exchange	Patch+CI	25	**0.0909**	0.0130	0.0822	**0.2127**	0.0176	**0.1998**
	Patch+CD	12	0.1174	0.0111	0.1062	0.2430	0.0124	0.2309
	Point+CI	18	0.0927	0.0110	**0.0822**	0.2147	0.0147	0.2008
	Point+CD	11	0.1173	0.0168	0.0974	0.2403	0.0149	0.2226
	Variate+CD	31	0.1028	0.0147	0.0851	0.2299	0.0209	0.2047
Illness	Patch+CI	24	3.2383	1.4320	**1.9044**	1.2715	0.3122	0.9661
	Patch+CD	17	**2.7241**	1.0101	1.9774	**1.1410**	0.2554	**0.9349**
	Point+CI	17	3.0837	1.0945	2.1624	1.2502	0.2415	1.0140
	Point+CD	16	3.1035	1.5305	2.0497	1.2250	0.3284	0.9459
	Variate+CD	29	3.2989	0.7744	2.2362	1.2637	0.1617	1.0124
Weather	Patch+CI	25	0.1707	0.0154	**0.1494**	0.2164	0.0119	**0.2002**
	Patch+CD	17	0.1739	0.0125	0.1545	0.2229	0.0145	0.2016
	Point+CI	19	0.1821	0.0150	0.1542	0.2261	0.0140	0.2044
	Point+CD	19	0.1707	0.0165	0.1518	0.2210	0.0173	0.2059
	Variate+CD	27	**0.1579**	0.0075	0.1496	**0.2090**	0.0062	0.2033
ECL	Patch+CI	21	**0.1630**	0.0224	**0.1328**	**0.2545**	0.0207	**0.2270**
	Patch+CD	0	–	–	–	–	–	–
	Point+CI	1	0.1994	–	0.1994	0.2805	–	0.2805
	Point+CD	17	0.1761	0.0132	0.1647	0.2824	0.0122	0.2701
	Variate+CD	27	0.1795	0.0114	0.1622	0.2834	0.0116	0.2657
Traffic	Patch+CI	23	**0.4671**	0.0619	**0.3817**	**0.3101**	0.0363	**0.2640**
	Patch+CD	0	–	–	–	–	–	–
	Point+CI	0	–	–	–	–	–	–
	Point+CD	10	0.6518	0.0476	0.6163	0.3670	0.0319	0.3350
	Variate+CD	29	0.5846	0.0680	0.4639	0.3657	0.0255	0.3293

Table 9: Configuration distributions for the Transformer encoder on nine datasets. N is the retained configuration count. For MSE and MAE, we report the mean $\hat{\mu}$, sample standard deviation $\hat{\sigma}$, and minimum L_{best}. Bold marks the lowest mean and minimum separately within each dataset and metric. A dash indicates an unavailable result or an undefined standard deviation for $N = 1$. Search and filtering conditions are specified in Appendix D.4.

Dataset	View	N	MSE			MAE		
			$\hat{\mu}$	$\hat{\sigma}$	L_{best}	$\hat{\mu}$	$\hat{\sigma}$	L_{best}
ETTh1	Patch+CI	23	**0.4040**	0.0479	**0.3745**	**0.4130**	0.0292	0.3936
	Patch+CD	20	0.4671	0.0463	0.4179	0.4564	0.0322	0.4200
	Point+CI	20	0.4113	0.0477	0.3808	0.4139	0.0314	**0.3928**
	Point+CD	17	0.4755	0.0476	0.4159	0.4594	0.0323	0.4191
	Variate+CD	28	0.4065	0.0346	0.3830	0.4192	0.0218	0.4011
ETTh2	Patch+CI	23	0.3085	0.0195	0.2819	0.3571	0.0161	0.3400
	Patch+CD	18	0.3658	0.0269	0.3380	0.3972	0.0206	0.3734
	Point+CI	22	**0.3079**	0.0307	**0.2813**	**0.3549**	0.0229	**0.3390**
	Point+CD	22	0.3605	0.0178	0.3379	0.3958	0.0167	0.3758
	Variate+CD	30	0.3171	0.0166	0.2935	0.3655	0.0131	0.3482
ETTm1	Patch+CI	24	**0.3199**	0.0247	**0.2893**	**0.3603**	0.0139	**0.3423**
	Patch+CD	22	0.3479	0.0157	0.3276	0.3845	0.0117	0.3695
	Point+CI	19	0.3310	0.0246	0.2949	0.3684	0.0146	0.3486
	Point+CD	19	0.3493	0.0185	0.3163	0.3853	0.0131	0.3624
	Variate+CD	29	0.3215	0.0160	0.2961	0.3637	0.0068	0.3506
ETTm2	Patch+CI	27	**0.1768**	0.0069	**0.1657**	**0.2632**	0.0067	**0.2523**
	Patch+CD	19	0.1908	0.0094	0.1765	0.2754	0.0113	0.2597
	Point+CI	21	0.1849	0.0085	0.1690	0.2718	0.0083	0.2580
	Point+CD	20	0.1989	0.0085	0.1842	0.2816	0.0085	0.2650
	Variate+CD	27	0.1817	0.0051	0.1721	0.2677	0.0045	0.2603
Exchange	Patch+CI	25	**0.0972**	0.0249	0.0828	**0.2185**	0.0274	0.2012
	Patch+CD	17	0.1433	0.0461	0.1061	0.2685	0.0447	0.2313
	Point+CI	21	0.0987	0.0276	**0.0821**	0.2210	0.0301	**0.2003**
	Point+CD	18	0.1510	0.0456	0.1067	0.2759	0.0454	0.2311
	Variate+CD	31	0.1057	0.0226	0.0857	0.2317	0.0264	0.2059
Illness	Patch+CI	22	**2.7333**	0.5324	2.0331	1.1676	0.1445	0.9747
	Patch+CD	18	2.7675	0.9153	**2.0176**	1.1660	0.2520	**0.9434**
	Point+CI	15	2.8361	0.6904	2.1684	1.1924	0.1868	1.0137
	Point+CD	15	2.7532	0.9667	2.0863	**1.1457**	0.2340	0.9490
	Variate+CD	29	3.2556	0.6858	2.4164	1.2468	0.1527	1.0317
Weather	Patch+CI	25	0.1710	0.0189	0.1473	0.2178	0.0174	**0.1977**
	Patch+CD	18	0.1670	0.0111	0.1557	0.2198	0.0140	0.2033
	Point+CI	16	0.1832	0.0107	0.1662	0.2267	0.0071	0.2157
	Point+CD	19	0.1701	0.0133	0.1597	0.2208	0.0141	0.2078
	Variate+CD	27	**0.1656**	0.0139	**0.1470**	**0.2136**	0.0119	0.1983
ECL	Patch+CI	12	0.1824	0.0314	0.1416	0.2720	0.0276	0.2386
	Patch+CD	0	–	–	–	–	–	–
	Point+CI	0	–	–	–	–	–	–
	Point+CD	21	0.1818	0.0201	0.1627	0.2856	0.0172	0.2676
	Variate+CD	26	**0.1628**	0.0216	**0.1366**	**0.2603**	0.0206	**0.2345**
Traffic	Patch+CI	23	0.4985	0.0977	**0.3847**	**0.3247**	0.0579	**0.2647**
	Patch+CD	0	–	–	–	–	–	–
	Point+CI	0	–	–	–	–	–	–
	Point+CD	8	0.6679	0.0522	0.6270	0.3762	0.0333	0.3466
	Variate+CD	24	**0.4686**	0.0637	0.3946	0.3306	0.0391	0.2837

Table 10: Configuration distributions for the GRU encoder on nine datasets. N is the retained configuration count. For MSE and MAE, we report the mean $\hat{\mu}$, sample standard deviation $\hat{\sigma}$, and minimum L_{best}. Bold marks the lowest mean and minimum separately within each dataset and metric. A dash indicates an unavailable result or an undefined standard deviation for $N = 1$. The seven small and medium datasets use the 31-configuration search; ECL and Traffic use the 12-configuration search with at most five epochs. Point+CI on ECL and Traffic is included wherever observed.

Dataset	View	N	MSE			MAE		
			$\hat{\mu}$	$\hat{\sigma}$	L_{best}	$\hat{\mu}$	$\hat{\sigma}$	L_{best}
ETTh1	Patch+CI	26	0.4714	0.1090	0.3845	0.4493	0.0586	**0.3977**
	Point+CD	18	0.4786	0.0660	0.4236	0.4622	0.0354	0.4245
	Variate+CD	28	**0.4326**	0.0542	**0.3817**	**0.4390**	0.0320	0.4032
ETTh2	Patch+CI	24	**0.3109**	0.0262	**0.2860**	**0.3603**	0.0209	**0.3424**
	Point+CD	21	0.3638	0.0266	0.3351	0.3957	0.0240	0.3716
	Variate+CD	30	0.3203	0.0228	0.2977	0.3709	0.0177	0.3503
ETTm1	Patch+CI	26	**0.3271**	0.0272	0.2978	**0.3648**	0.0160	**0.3476**
	Point+CD	21	0.3452	0.0240	0.3196	0.3829	0.0162	0.3651
	Variate+CD	29	0.3359	0.0307	**0.2940**	0.3765	0.0192	0.3509
ETTm2	Patch+CI	27	**0.1782**	0.0067	**0.1659**	**0.2642**	0.0062	**0.2545**
	Point+CD	20	0.1942	0.0090	0.1781	0.2774	0.0094	0.2594
	Variate+CD	28	0.1865	0.0094	0.1722	0.2747	0.0101	0.2630
Exchange	Patch+CI	26	**0.1036**	0.0345	**0.0833**	**0.2272**	0.0349	**0.2015**
	Point+CD	22	0.1692	0.0687	0.1061	0.2924	0.0633	0.2296
	Variate+CD	31	0.1275	0.0480	0.0884	0.2535	0.0485	0.2099
Illness	Patch+CI	25	7.6675	7.8600	2.1927	1.9696	1.1002	1.0461
	Point+CD	14	6.2456	6.8197	**2.0774**	1.6472	0.8603	**0.9723**
	Variate+CD	29	**3.5853**	0.8507	2.3942	**1.3309**	0.1771	1.0696
Weather	Patch+CI	25	0.1751	0.0173	0.1529	0.2210	0.0149	0.2034
	Point+CD	18	0.1748	0.0168	0.1605	0.2252	0.0169	0.2090
	Variate+CD	27	**0.1699**	0.0137	**0.1491**	**0.2196**	0.0129	**0.2004**
ECL	Patch+CI	11	0.2084	0.0296	0.1647	0.3007	0.0241	0.2727
	Point+CI	8	0.2132	0.0247	**0.1642**	0.3010	0.0203	**0.2697**
	Point+CD	10	**0.2005**	0.0278	0.1779	0.3027	0.0246	0.2843
	Variate+CD	11	0.2044	0.0272	0.1653	**0.2988**	0.0227	0.2704
Traffic	Patch+CI	11	0.6377	0.1263	0.4447	0.4180	0.0612	0.3289
	Point+CI	4	0.6809	0.0113	0.6657	0.4129	0.0062	0.4063
	Point+CD	4	0.6701	0.0223	0.6508	**0.3747**	0.0183	0.3592
	Variate+CD	11	**0.5919**	0.1314	**0.4372**	0.4033	0.0608	**0.3268**

Table 11: Configuration distributions for the diagonal SSM encoder on nine datasets. N is the retained configuration count. For MSE and MAE, we report the mean $\hat{\mu}$, sample standard deviation $\hat{\sigma}$, and minimum L_{best}. Bold marks the lowest mean and minimum separately within each dataset and metric. A dash indicates an unavailable result or an undefined standard deviation for $N = 1$. The seven small and medium datasets use the 31-configuration search; ECL and Traffic use the 12-configuration search with at most five epochs. Point+CI on ECL and Traffic is included wherever observed.

Dataset	View	N	MSE			MAE		
			$\hat{\mu}$	$\hat{\sigma}$	L_{best}	$\hat{\mu}$	$\hat{\sigma}$	L_{best}
ETTh1	Patch+CI	24	0.4115	0.0433	0.3800	**0.4171**	0.0312	**0.3962**
	Point+CD	14	0.4642	0.0246	0.4303	0.4527	0.0204	0.4265
	Variate+CD	24	**0.4081**	0.0412	**0.3767**	0.4209	0.0246	0.3990
ETTh2	Patch+CI	22	**0.2994**	0.0135	**0.2882**	**0.3515**	0.0125	**0.3417**
	Point+CD	19	0.3704	0.0236	0.3436	0.3985	0.0184	0.3764
	Variate+CD	25	0.3082	0.0134	0.2916	0.3605	0.0107	0.3480
ETTm1	Patch+CI	25	0.3197	0.0176	0.2958	**0.3604**	0.0104	**0.3453**
	Point+CD	16	0.3464	0.0072	0.3312	0.3813	0.0051	0.3713
	Variate+CD	25	**0.3179**	0.0203	**0.2920**	0.3621	0.0105	0.3481
ETTm2	Patch+CI	25	**0.1767**	0.0060	**0.1681**	**0.2634**	0.0044	**0.2577**
	Point+CD	21	0.2015	0.0102	0.1806	0.2848	0.0117	0.2626
	Variate+CD	24	0.1791	0.0043	0.1716	0.2648	0.0045	0.2594
Exchange	Patch+CI	23	**0.0915**	0.0085	**0.0816**	**0.2139**	0.0130	**0.2000**
	Point+CD	12	0.1191	0.0079	0.1104	0.2449	0.0105	0.2351
	Variate+CD	25	0.1009	0.0174	0.0879	0.2253	0.0209	0.2079
Illness	Patch+CI	20	3.9440	2.9671	2.1146	1.4121	0.5538	1.0459
	Point+CD	12	**3.3212**	2.0293	**2.0214**	**1.2548**	0.3817	**0.9721**
	Variate+CD	24	3.6465	0.9822	2.2098	1.3472	0.2004	1.0566
Weather	Patch+CI	23	0.1720	0.0118	0.1558	0.2184	0.0092	0.2065
	Point+CD	17	0.1680	0.0086	0.1602	0.2186	0.0098	0.2083
	Variate+CD	21	**0.1632**	0.0114	**0.1479**	**0.2107**	0.0080	**0.1981**
ECL	Patch+CI	12	0.1940	0.0264	**0.1427**	0.2870	0.0291	**0.2415**
	Point+CI	4	0.2006	0.0059	0.1966	0.2809	0.0074	0.2765
	Point+CD	9	0.1918	0.0288	0.1728	0.2967	0.0261	0.2770
	Variate+CD	11	**0.1823**	0.0301	0.1446	**0.2775**	0.0277	0.2449
Traffic	Patch+CI	12	0.5790	0.0944	**0.4121**	0.3839	0.0474	**0.2913**
	Point+CI	1	0.6936	–	0.6936	0.4121	–	0.4121
	Point+CD	5	0.7194	0.0779	0.6730	0.4037	0.0397	0.3774
	Variate+CD	11	**0.5292**	0.0898	0.4135	**0.3613**	0.0542	0.2990

Table 12: Configuration distributions for the FFT auto-correlation encoder on nine datasets. N is the retained configuration count. For MSE and MAE, we report the mean $\hat{\mu}$, sample standard deviation $\hat{\sigma}$, and minimum L_{best}. Bold marks the lowest mean and minimum separately within each dataset and metric. A dash indicates an unavailable result or an undefined standard deviation for $N = 1$. The seven small and medium datasets use the 31-configuration search; ECL and Traffic use the 12-configuration search with at most five epochs. Point+CI on ECL and Traffic is included wherever observed.

Dataset	View	N	MSE			MAE		
			$\hat{\mu}$	$\hat{\sigma}$	L_{best}	$\hat{\mu}$	$\hat{\sigma}$	L_{best}
ETTh1	Patch+CI	27	0.4420	0.0916	**0.3780**	**0.4327**	0.0497	**0.3967**
	Point+CD	19	0.5124	0.1001	0.4278	0.4761	0.0512	0.4267
	Variate+CD	29	**0.4308**	0.0648	0.3792	0.4338	0.0363	0.4004
ETTh2	Patch+CI	26	**0.3091**	0.0213	**0.2844**	**0.3575**	0.0176	**0.3384**
	Point+CD	20	0.3923	0.0436	0.3378	0.4111	0.0305	0.3758
	Variate+CD	29	0.3102	0.0189	0.2898	0.3635	0.0161	0.3481
ETTm1	Patch+CI	26	0.3502	0.0521	0.2995	0.3785	0.0276	0.3504
	Point+CD	21	0.4624	0.1138	0.3357	0.4414	0.0526	0.3726
	Variate+CD	29	**0.3418**	0.0572	**0.2870**	**0.3738**	0.0253	**0.3429**
ETTm2	Patch+CI	28	0.1915	0.0188	0.1730	0.2768	0.0166	0.2599
	Point+CD	20	0.2099	0.0169	0.1857	0.2922	0.0168	0.2681
	Variate+CD	29	**0.1842**	0.0090	**0.1713**	**0.2714**	0.0092	**0.2569**
Exchange	Patch+CI	27	0.1187	0.0466	**0.0834**	0.2417	0.0439	**0.2013**
	Point+CD	18	0.1541	0.0515	0.1090	0.2773	0.0482	0.2339
	Variate+CD	29	**0.1037**	0.0216	0.0837	**0.2289**	0.0260	0.2021
Illness	Patch+CI	26	5.5282	5.6318	2.2318	1.5968	0.7416	1.0339
	Point+CD	16	6.5612	8.6117	**2.0608**	1.6935	1.1669	**0.9841**
	Variate+CD	29	**3.4805**	0.8135	2.3014	**1.3018**	0.1707	1.0453
Weather	Patch+CI	26	0.1848	0.0146	0.1667	0.2296	0.0124	0.2151
	Point+CD	20	0.1824	0.0229	0.1616	0.2335	0.0240	0.2097
	Variate+CD	29	**0.1669**	0.0149	**0.1468**	**0.2144**	0.0116	**0.1997**
ECL	Patch+CI	11	0.2011	0.0331	**0.1471**	0.2908	0.0271	**0.2464**
	Point+CI	2	0.2030	0.0076	0.1976	0.2846	0.0103	0.2773
	Point+CD	8	**0.1841**	0.0115	0.1754	0.2879	0.0090	0.2800
	Variate+CD	11	0.1851	0.0208	0.1507	**0.2792**	0.0173	0.2521
Traffic	Patch+CI	7	0.6319	0.1124	0.5115	0.4041	0.0506	0.3574
	Point+CI	1	0.6518	–	0.6518	0.3902	–	0.3902
	Point+CD	5	0.7216	0.0792	0.6630	0.4105	0.0489	0.3691
	Variate+CD	11	**0.5366**	0.0734	**0.4266**	**0.3658**	0.0390	**0.3074**

Table 13: Configuration distributions for the two-layer CNN encoder on nine datasets. N is the retained configuration count. For MSE and MAE, we report the mean $\hat{\mu}$, sample standard deviation $\hat{\sigma}$, and minimum L_{best}. Bold marks the lowest mean and minimum separately within each dataset and metric. A dash indicates an unavailable result or an undefined standard deviation for $N = 1$. The search uses seed 2021 and at most ten epochs, with batch size 256 except for ECL and Traffic (32).

Dataset	View	N	MSE			MAE		
			$\hat{\mu}$	$\hat{\sigma}$	L_{best}	$\hat{\mu}$	$\hat{\sigma}$	L_{best}
ETTh1	Patch+CI	22	0.4419	0.0840	**0.3801**	0.4327	0.0479	**0.3936**
	Point+CD	15	0.4917	0.0704	0.4299	0.4677	0.0398	0.4255
	Variate+CD	24	**0.4155**	0.0440	0.3892	**0.4274**	0.0245	0.4068
ETTh2	Patch+CI	22	**0.3031**	0.0202	**0.2857**	**0.3547**	0.0163	**0.3407**
	Point+CD	16	0.3630	0.0131	0.3430	0.3944	0.0128	0.3780
	Variate+CD	24	0.3088	0.0138	0.2906	0.3615	0.0123	0.3496
ETTm1	Patch+CI	20	**0.3226**	0.0183	**0.2942**	**0.3613**	0.0111	**0.3428**
	Point+CD	14	0.3427	0.0129	0.3241	0.3789	0.0085	0.3671
	Variate+CD	24	0.3282	0.0191	0.3016	0.3702	0.0105	0.3560
ETTm2	Patch+CI	21	**0.1779**	0.0079	**0.1670**	**0.2645**	0.0071	**0.2560**
	Point+CD	15	0.1941	0.0087	0.1799	0.2765	0.0088	0.2624
	Variate+CD	24	0.1841	0.0036	0.1763	0.2705	0.0031	0.2673
Exchange	Patch+CI	22	**0.0923**	0.0087	**0.0819**	**0.2145**	0.0121	**0.2000**
	Point+CD	16	0.1514	0.0489	0.1063	0.2746	0.0454	0.2304
	Variate+CD	24	0.1113	0.0237	0.0897	0.2392	0.0276	0.2119
Illness	Patch+CI	22	6.6809	5.8240	2.1770	1.8244	0.8865	1.0294
	Point+CD	11	7.9655	12.2803	**2.0147**	1.7890	1.4709	**0.9721**
	Variate+CD	24	**3.5320**	0.8612	2.1924	**1.3223**	0.1885	1.0253
Weather	Patch+CI	22	0.1790	0.0170	0.1560	0.2241	0.0137	0.2068
	Point+CD	15	0.1687	0.0089	0.1600	0.2190	0.0097	0.2073
	Variate+CD	24	**0.1647**	0.0134	**0.1486**	**0.2131**	0.0102	**0.1989**
ECL	Patch+CI	21	0.1625	0.0228	0.1333	0.2556	0.0216	**0.2275**
	Point+CD	16	0.1823	0.0117	0.1676	0.2911	0.0138	0.2728
	Variate+CD	24	**0.1575**	0.0215	**0.1311**	**0.2522**	0.0182	0.2279
Traffic	Patch+CI	19	0.4884	0.0821	0.3935	**0.3197**	0.0404	**0.2691**
	Point+CD	8	0.6457	0.0224	0.6271	0.3658	0.0173	0.3530
	Variate+CD	24	**0.4627**	0.0714	**0.3818**	0.3239	0.0394	0.2739

Table 14: Point+CI configuration results with the encoder and prediction head computed in FP32 within mixed-precision training. All 19 successful configurations are included without IQR filtering. Every run uses seed 2021, input length 336, prediction length 96, batch size 256, and at most 100 epochs with patience 10. N counts configurations. For each metric, $\hat{\mu}$, $\hat{\sigma}$, and L_{best} denote the mean, sample standard deviation, and minimum. A dash marks an undefined standard deviation for $N = 1$.

Encoder	Dataset	N	MSE			MAE		
			$\hat{\mu}$	$\hat{\sigma}$	L_{best}	$\hat{\mu}$	$\hat{\sigma}$	L_{best}
GRU	ETTm1	1	0.3200	–	0.3200	0.3617	–	0.3617
GRU	ETTm2	1	0.1822	–	0.1822	0.2700	–	0.2700
GRU	Weather	9	0.2390	0.0701	0.1653	0.2805	0.0510	0.2143
SSM	Weather	5	32.3310	44.1434	0.1705	2.8211	3.3323	0.2207
FFT	Weather	3	6.4596	10.2037	0.4103	1.1328	1.1422	0.3797

D.2 Fixed-Configuration Results

We evaluate all six encoders at fixed configurations using the architectures specified in Appendix B.1. MLPMixer, Transformer, GRU, diagonal SSM, and FFT auto-correlation use seeds 2021, 2022, and 2023; CNN uses seed 2021. The None baseline directly maps each channel's history to its forecast using the linear readout specified there. Its three-seed results are shared across the five encoders evaluated with three seeds.

The three-seed runs use input length 96, prediction length 96, hidden dimension 128, learning rate 10^{-3}, batch size 256, and RevIN. Patch length and stride are 16 and 8. The maximum epoch count is 100 with early-stopping patience 10 for ETTh1, ETTh2, ETTm1, ETTm2, Exchange, Illness, and Weather. ECL and Traffic use five epochs. Each run is evaluated at its validation-selected checkpoint. All runs use mixed precision.

All five encoders are evaluated with seeds 2021, 2022, and 2023 on all nine datasets and four views (None, Patch+CI, Point+CD, and Variate+CD). For each dataset–encoder–view combination, the architecture, optimization settings, training budget, and checkpoint-selection rule are held fixed across the three seeds. Only the random seed changes.

Encoder architectures are described in Appendix B.1. The comparison retains these view-specific settings, so it concerns embedding–encoder configurations rather than changing tokenization at an identical parameter count.

Table 15 summarizes the mean test MSE for the three embedding views under all five encoders. Tables 16–20 report MSE and MAE as the arithmetic mean and sample standard deviation across the three seeds. Each cell uses $N = 3$ observations, with standard deviation computed using denominator $N - 1$. The bold mean identifies the lowest value among the four views for the same dataset and metric. These statistics describe variation across random seeds at each fixed configuration. The distributional $\hat{\sigma}$ in Table 2 instead describes variation across evaluation conditions for each encoder family.

Table 15: Seed-averaged test MSE for the five encoders at the reported configurations. Every cell uses seeds 2021, 2022, and 2023. Bold marks the lowest mean among the three embedding views. ECL and Traffic use five training epochs. The shared None baseline, MAE, and sample standard deviations appear in Tables 16–20.

Dataset	MLPMixer			Transformer		
	Patch+CI	Point+CD	Variate+CD	Patch+CI	Point+CD	Variate+CD
ETTh1	0.3849	0.4354	**0.3822**	**0.3783**	0.4461	0.3895
ETTh2	0.2976	0.3498	**0.2976**	**0.2946**	0.3464	0.3131
ETTm1	0.3252	0.3381	**0.3207**	**0.3313**	0.3486	0.3347
ETTm2	**0.1766**	0.1958	0.1793	**0.1786**	0.1991	0.1869
Exchange	**0.0853**	0.1210	0.0891	**0.0883**	0.1227	0.0907
Illness	**2.1166**	3.0712	2.6385	**2.1766**	2.3613	2.6371
Weather	0.1679	0.1614	**0.1583**	0.1653	**0.1628**	0.1701
Five-epoch protocol						
ECL	0.1882	0.1737	**0.1691**	0.1843	0.1815	**0.1727**
Traffic	0.5738	0.6684	**0.5651**	0.5371	0.6706	**0.4826**

Dataset	GRU			Diagonal SSM			FFT auto-correlation		
	Patch+CI	Point+CD	Variate+CD	Patch+CI	Point+CD	Variate+CD	Patch+CI	Point+CD	Variate+CD
ETTh1	0.3905	0.4476	**0.3882**	0.3879	0.4596	**0.3855**	**0.3892**	0.4526	0.3983
ETTh2	**0.2901**	0.3409	0.3035	**0.2988**	0.3566	0.3016	**0.2926**	0.3411	0.2973
ETTm1	**0.3204**	0.3407	0.3244	0.3263	0.3580	**0.3208**	**0.3462**	0.5607	0.3694
ETTm2	**0.1766**	0.2001	0.1800	0.1788	0.2075	**0.1770**	**0.1779**	0.2202	0.1838
Exchange	**0.0872**	0.1109	0.0898	**0.0838**	0.1146	0.0908	**0.0846**	0.1225	0.0853
Illness	**2.2423**	2.9836	2.9103	**2.3934**	3.1272	2.9759	**2.2572**	2.6588	2.7073
Weather	0.1690	**0.1640**	0.1685	0.1720	**0.1629**	0.1679	0.1773	**0.1735**	0.1761
Five-epoch protocol									
ECL	0.1986	**0.1917**	0.2022	0.1904	**0.1785**	0.1848	0.1966	**0.1830**	0.1919
Traffic	0.6109	0.6638	**0.5951**	0.5897	0.6919	**0.5515**	0.5672	0.6837	**0.5241**

Table 16: Random-seed results for the MLPMixer encoder. Values are mean ± sample standard deviation over seeds 2021, 2022, and 2023. Bold denotes the lowest mean in each dataset–metric row. ECL and Traffic use five training epochs.

Dataset	Metric	None	Patch+CI	Point+CD	Variate+CD
ETTh1	MSE	0.4178 ± 0.001275	0.3849 ± 0.002006	0.4354 ± 0.013727	$\mathbf{0.3822 \pm 0.002447}$
	MAE	0.4217 ± 0.000930	$\mathbf{0.3943 \pm 0.000716}$	0.4353 ± 0.007853	0.4001 ± 0.001335
ETTh2	MSE	0.3072 ± 0.001842	0.2976 ± 0.006887	0.3498 ± 0.011757	$\mathbf{0.2976 \pm 0.002497}$
	MAE	0.3538 ± 0.001201	$\mathbf{0.3472 \pm 0.006487}$	0.3818 ± 0.003388	0.3480 ± 0.001381
ETTm1	MSE	0.3574 ± 0.000688	0.3252 ± 0.002859	0.3381 ± 0.009452	$\mathbf{0.3207 \pm 0.002589}$
	MAE	0.3748 ± 0.000665	0.3637 ± 0.002295	0.3765 ± 0.006433	$\mathbf{0.3562 \pm 0.000331}$
ETTm2	MSE	0.1848 ± 0.000126	$\mathbf{0.1766 \pm 0.000307}$	0.1958 ± 0.000574	0.1793 ± 0.002466
	MAE	0.2686 ± 0.000221	$\mathbf{0.2624 \pm 0.000179}$	0.2766 ± 0.001218	0.2637 ± 0.002182
Exchange	MSE	0.0911 ± 0.000555	$\mathbf{0.0853 \pm 0.002490}$	0.1210 ± 0.012438	0.0891 ± 0.000580
	MAE	0.2139 ± 0.000799	$\mathbf{0.2041 \pm 0.002824}$	0.2415 ± 0.008019	0.2091 ± 0.000266
Illness	MSE	4.0597 ± 0.212478	$\mathbf{2.1166 \pm 0.015297}$	3.0712 ± 0.753279	2.6385 ± 0.141989
	MAE	1.4438 ± 0.042260	$\mathbf{1.0321 \pm 0.023121}$	1.2317 ± 0.171960	1.1350 ± 0.044071
Weather	MSE	0.1945 ± 0.000406	0.1679 ± 0.000726	0.1614 ± 0.004546	$\mathbf{0.1583 \pm 0.000988}$
	MAE	0.2344 ± 0.000192	0.2125 ± 0.000886	0.2087 ± 0.004397	$\mathbf{0.2061 \pm 0.001480}$
Five-epoch protocol					
ECL	MSE	0.2285 ± 0.000400	0.1882 ± 0.004260	0.1737 ± 0.007415	$\mathbf{0.1691 \pm 0.002997}$
	MAE	0.3136 ± 0.000351	$\mathbf{0.2718 \pm 0.003190}$	0.2775 ± 0.006457	0.2736 ± 0.003598
Traffic	MSE	0.7501 ± 0.001802	0.5738 ± 0.023664	0.6684 ± 0.017958	$\mathbf{0.5651 \pm 0.027584}$
	MAE	0.4572 ± 0.000781	$\mathbf{0.3660 \pm 0.010774}$	0.3714 ± 0.014017	0.3788 ± 0.019703

Table 17: Random-seed results for the Transformer encoder. Values are mean ± sample standard deviation over seeds 2021, 2022, and 2023. Bold denotes the lowest mean in each dataset–metric row. ECL and Traffic use five training epochs.

Dataset	Metric	None	Patch+CI	Point+CD	Variate+CD
ETTh1	MSE	0.4178 ± 0.001275	$\mathbf{0.3783 \pm 0.001881}$	0.4461 ± 0.001931	0.3895 ± 0.005954
	MAE	0.4217 ± 0.000930	$\mathbf{0.3949 \pm 0.001376}$	0.4389 ± 0.001729	0.4074 ± 0.004458
ETTh2	MSE	0.3072 ± 0.001842	$\mathbf{0.2946 \pm 0.012234}$	0.3464 ± 0.002835	0.3131 ± 0.004498
	MAE	0.3538 ± 0.001201	$\mathbf{0.3456 \pm 0.010903}$	0.3798 ± 0.003877	0.3590 ± 0.003715
ETTm1	MSE	0.3574 ± 0.000688	$\mathbf{0.3313 \pm 0.005448}$	0.3486 ± 0.008103	0.3347 ± 0.005659
	MAE	0.3748 ± 0.000665	$\mathbf{0.3661 \pm 0.003494}$	0.3839 ± 0.008167	0.3675 ± 0.002925
ETTm2	MSE	0.1848 ± 0.000126	$\mathbf{0.1786 \pm 0.001747}$	0.1991 ± 0.014254	0.1869 ± 0.000946
	MAE	0.2686 ± 0.000221	$\mathbf{0.2611 \pm 0.001217}$	0.2791 ± 0.011288	0.2695 ± 0.001808
Exchange	MSE	0.0911 ± 0.000555	$\mathbf{0.0883 \pm 0.002078}$	0.1227 ± 0.000887	0.0907 ± 0.002909
	MAE	0.2139 ± 0.000799	$\mathbf{0.2060 \pm 0.002643}$	0.2498 ± 0.000443	0.2133 ± 0.004299
Illness	MSE	4.0597 ± 0.212478	$\mathbf{2.1766 \pm 0.105879}$	2.3613 ± 0.446095	2.6371 ± 0.118256
	MAE	1.4438 ± 0.042260	$\mathbf{1.0161 \pm 0.038326}$	1.0534 ± 0.122648	1.1126 ± 0.025407
Weather	MSE	0.1945 ± 0.000406	0.1653 ± 0.000512	$\mathbf{0.1628 \pm 0.002004}$	0.1701 ± 0.000277
	MAE	0.2344 ± 0.000192	$\mathbf{0.2105 \pm 0.000896}$	0.2125 ± 0.001934	0.2139 ± 0.001339
Five-epoch protocol					
ECL	MSE	0.2285 ± 0.000400	0.1843 ± 0.007063	0.1815 ± 0.012292	$\mathbf{0.1727 \pm 0.013368}$
	MAE	0.3136 ± 0.000351	0.2709 ± 0.005776	0.2821 ± 0.010147	$\mathbf{0.2666 \pm 0.010961}$
Traffic	MSE	0.7501 ± 0.001802	0.5371 ± 0.023218	0.6706 ± 0.018384	$\mathbf{0.4826 \pm 0.022771}$
	MAE	0.4572 ± 0.000781	0.3551 ± 0.018500	0.3782 ± 0.014531	$\mathbf{0.3362 \pm 0.015706}$

The CNN fixed configuration uses input and prediction lengths of 96, hidden width 128, learning rate 10^{-3}, and the two-layer convolutional architecture in Appendix B.1. Patch+CI uses patch length 16 and stride 8. All runs use seed 2021, Adam with MSE training loss, RevIN, mixed precision, dropout 0.1, and at most ten epochs with early-stopping patience 10. Batch size is 256 except for ECL and Traffic, where it is 32. Each test result is evaluated at its validation-selected checkpoint. Table 21

Table 18: Complete three-seed results for the GRU encoder. Values are mean ± sample standard deviation. Bold denotes the lowest mean in each dataset–metric row. ECL and Traffic use five training epochs.

Dataset	Metric	None	Patch+CI	Point+CD	Variate+CD
ETTh1	MSE	0.4178 ± 0.001275	0.3905 ± 0.000987	0.4476 ± 0.007313	$\mathbf{0.3882 \pm 0.000797}$
	MAE	0.4217 ± 0.000930	$\mathbf{0.3973 \pm 0.000389}$	0.4418 ± 0.003839	0.4049 ± 0.000421
ETTh2	MSE	0.3072 ± 0.001842	$\mathbf{0.2901 \pm 0.001421}$	0.3409 ± 0.000594	0.3035 ± 0.002479
	MAE	0.3538 ± 0.001201	$\mathbf{0.3425 \pm 0.001390}$	0.3738 ± 0.001053	0.3565 ± 0.001628
ETTm1	MSE	0.3574 ± 0.000688	$\mathbf{0.3204 \pm 0.001038}$	0.3407 ± 0.005177	0.3244 ± 0.000698
	MAE	0.3748 ± 0.000665	$\mathbf{0.3601 \pm 0.000734}$	0.3788 ± 0.004229	0.3632 ± 0.000464
ETTm2	MSE	0.1848 ± 0.000126	$\mathbf{0.1766 \pm 0.000265}$	0.2001 ± 0.004023	0.1800 ± 0.000030
	MAE	0.2686 ± 0.000221	$\mathbf{0.2618 \pm 0.000444}$	0.2793 ± 0.003298	0.2643 ± 0.000099
Exchange	MSE	0.0911 ± 0.000555	$\mathbf{0.0872 \pm 0.001405}$	0.1109 ± 0.000477	0.0898 ± 0.000282
	MAE	0.2139 ± 0.000799	$\mathbf{0.2062 \pm 0.001518}$	0.2366 ± 0.000063	0.2114 ± 0.000351
Illness	MSE	4.0597 ± 0.212478	$\mathbf{2.2423 \pm 0.018696}$	2.9836 ± 0.137512	2.9103 ± 0.035850
	MAE	1.4438 ± 0.042260	$\mathbf{1.0523 \pm 0.001237}$	1.2578 ± 0.083989	1.1900 ± 0.005624
Weather	MSE	0.1945 ± 0.000406	0.1690 ± 0.000417	$\mathbf{0.1640 \pm 0.002425}$	0.1685 ± 0.000619
	MAE	0.2344 ± 0.000192	0.2134 ± 0.001306	$\mathbf{0.2110 \pm 0.001580}$	0.2154 ± 0.000632
Five-epoch protocol					
ECL	MSE	0.2285 ± 0.000400	0.1986 ± 0.000594	$\mathbf{0.1917 \pm 0.001027}$	0.2022 ± 0.000626
	MAE	0.3136 ± 0.000351	$\mathbf{0.2864 \pm 0.000631}$	0.2930 ± 0.000636	0.2909 ± 0.000564
Traffic	MSE	0.7501 ± 0.001802	0.6109 ± 0.002828	0.6638 ± 0.008517	$\mathbf{0.5951 \pm 0.001783}$
	MAE	0.4572 ± 0.000781	0.3835 ± 0.001581	$\mathbf{0.3707 \pm 0.005102}$	0.4021 ± 0.000943

Table 19: Complete three-seed results for the diagonal SSM encoder. Values are mean ± sample standard deviation. Bold denotes the lowest mean in each dataset–metric row. ECL and Traffic use five training epochs.

Dataset	Metric	None	Patch+CI	Point+CD	Variate+CD
ETTh1	MSE	0.4178 ± 0.001275	0.3879 ± 0.000587	0.4596 ± 0.009479	$\mathbf{0.3855 \pm 0.000246}$
	MAE	0.4217 ± 0.000930	$\mathbf{0.3956 \pm 0.000499}$	0.4437 ± 0.003529	0.4031 ± 0.000447
ETTh2	MSE	0.3072 ± 0.001842	$\mathbf{0.2988 \pm 0.009664}$	0.3566 ± 0.007918	0.3016 ± 0.000217
	MAE	0.3538 ± 0.001201	$\mathbf{0.3474 \pm 0.007626}$	0.3841 ± 0.002265	0.3501 ± 0.001380
ETTm1	MSE	0.3574 ± 0.000688	0.3263 ± 0.000818	0.3580 ± 0.009413	$\mathbf{0.3208 \pm 0.001382}$
	MAE	0.3748 ± 0.000665	0.3658 ± 0.000705	0.3873 ± 0.004635	$\mathbf{0.3593 \pm 0.000767}$
ETTm2	MSE	0.1848 ± 0.000126	0.1788 ± 0.000078	0.2075 ± 0.009280	$\mathbf{0.1770 \pm 0.000417}$
	MAE	0.2686 ± 0.000221	0.2636 ± 0.000527	0.2828 ± 0.005381	$\mathbf{0.2593 \pm 0.000246}$
Exchange	MSE	0.0911 ± 0.000555	$\mathbf{0.0838 \pm 0.001309}$	0.1146 ± 0.002701	0.0908 ± 0.000637
	MAE	0.2139 ± 0.000799	$\mathbf{0.2017 \pm 0.001451}$	0.2386 ± 0.002472	0.2127 ± 0.000697
Illness	MSE	4.0597 ± 0.212478	$\mathbf{2.3934 \pm 0.422302}$	3.1272 ± 0.072468	2.9759 ± 0.019862
	MAE	1.4438 ± 0.042260	$\mathbf{1.1219 \pm 0.084212}$	1.2488 ± 0.013705	1.2307 ± 0.006522
Weather	MSE	0.1945 ± 0.000406	0.1720 ± 0.002000	$\mathbf{0.1629 \pm 0.001775}$	0.1679 ± 0.000311
	MAE	0.2344 ± 0.000192	0.2155 ± 0.002038	$\mathbf{0.2109 \pm 0.001460}$	0.2120 ± 0.000180
Five-epoch protocol					
ECL	MSE	0.2285 ± 0.000400	0.1904 ± 0.000196	$\mathbf{0.1785 \pm 0.000461}$	0.1848 ± 0.001043
	MAE	0.3136 ± 0.000351	$\mathbf{0.2749 \pm 0.000112}$	0.2820 ± 0.000490	0.2769 ± 0.001076
Traffic	MSE	0.7501 ± 0.001802	0.5897 ± 0.002459	0.6919 ± 0.003054	$\mathbf{0.5515 \pm 0.001929}$
	MAE	0.4572 ± 0.000781	0.3763 ± 0.000612	0.3890 ± 0.000767	$\mathbf{0.3669 \pm 0.000955}$

Table 20: Complete three-seed results for the FFT auto-correlation encoder. Values are mean ± sample standard deviation. Bold denotes the lowest mean in each dataset–metric row. ECL and Traffic use five training epochs.

Dataset	Metric	None	Patch+CI	Point+CD	Variate+CD
ETTh1	MSE	0.4178 ± 0.001275	$\mathbf{0.3892 \pm 0.001943}$	0.4526 ± 0.010693	0.3983 ± 0.003528
	MAE	0.4217 ± 0.000930	$\mathbf{0.3970 \pm 0.001247}$	0.4412 ± 0.005043	0.4126 ± 0.002350
ETTh2	MSE	0.3072 ± 0.001842	$\mathbf{0.2926 \pm 0.001195}$	0.3411 ± 0.003360	0.2973 ± 0.001265
	MAE	0.3538 ± 0.001201	$\mathbf{0.3403 \pm 0.001883}$	0.3770 ± 0.001063	0.3489 ± 0.001628
ETTm1	MSE	0.3574 ± 0.000688	$\mathbf{0.3462 \pm 0.005811}$	0.5607 ± 0.078246	0.3694 ± 0.050926
	MAE	$\mathbf{0.3748 \pm 0.000665}$	0.3749 ± 0.003614	0.4883 ± 0.032694	0.3865 ± 0.026508
ETTm2	MSE	0.1848 ± 0.000126	$\mathbf{0.1779 \pm 0.002749}$	0.2202 ± 0.006833	0.1838 ± 0.006420
	MAE	0.2686 ± 0.000221	$\mathbf{0.2626 \pm 0.003518}$	0.2973 ± 0.005487	0.2665 ± 0.005850
Exchange	MSE	0.0911 ± 0.000555	$\mathbf{0.0846 \pm 0.001785}$	0.1225 ± 0.010046	0.0853 ± 0.003085
	MAE	0.2139 ± 0.000799	$\mathbf{0.2034 \pm 0.001853}$	0.2495 ± 0.009577	0.2041 ± 0.003474
Illness	MSE	4.0597 ± 0.212478	$\mathbf{2.2572 \pm 0.021437}$	2.6588 ± 0.282648	2.7073 ± 0.047885
	MAE	1.4438 ± 0.042260	$\mathbf{1.0414 \pm 0.006365}$	1.1459 ± 0.086392	1.1321 ± 0.011167
Weather	MSE	0.1945 ± 0.000406	0.1773 ± 0.001884	$\mathbf{0.1735 \pm 0.007302}$	0.1761 ± 0.000437
	MAE	0.2344 ± 0.000192	0.2210 ± 0.001440	0.2224 ± 0.006562	$\mathbf{0.2201 \pm 0.001222}$
Five-epoch protocol					
ECL	MSE	0.2285 ± 0.000400	0.1966 ± 0.000894	$\mathbf{0.1830 \pm 0.001675}$	0.1919 ± 0.002773
	MAE	0.3136 ± 0.000351	$\mathbf{0.2800 \pm 0.001896}$	0.2850 ± 0.001714	0.2822 ± 0.004746
Traffic	MSE	0.7501 ± 0.001802	0.5672 ± 0.025923	0.6837 ± 0.002475	$\mathbf{0.5241 \pm 0.002483}$
	MAE	0.4572 ± 0.000781	0.3650 ± 0.014805	0.3872 ± 0.006367	$\mathbf{0.3570 \pm 0.002344}$

gives the resulting MSE and MAE point estimates. Each dataset–view cell contains one run, so these results have $N = 1$ and no seed standard deviation. In contrast, the CNN standard deviations in Table 13 describe variation across different configurations.

Table 21: CNN results at the fixed configuration with seed 2021. Each entry is one test-set point estimate ($N = 1$); no seed standard deviation is defined. Input and prediction lengths are 96, hidden width is 128, and the learning rate is 10^{-3}. Training uses at most ten epochs and batch size 256, except for ECL and Traffic (32). Bold marks the lowest error among the three views for each dataset and metric.

Dataset	Metric	Patch+CI	Point+CD	Variate+CD
ETTh1	MSE	**0.3867**	0.4507	0.3920
	MAE	**0.3946**	0.4376	0.4092
ETTh2	MSE	**0.2961**	0.3647	0.3048
	MAE	**0.3467**	0.3862	0.3516
ETTm1	MSE	**0.3251**	0.3418	0.3310
	MAE	**0.3648**	0.3777	0.3674
ETTm2	MSE	**0.1774**	0.1986	0.1833
	MAE	**0.2628**	0.2768	0.2685
Exchange	MSE	**0.0841**	0.1201	0.0928
	MAE	**0.2031**	0.2449	0.2160
Illness	MSE	**2.1788**	4.3272	2.7695
	MAE	**1.0640**	1.5073	1.1672
Weather	MSE	0.1707	**0.1615**	0.1695
	MAE	0.2138	**0.2097**	0.2155
ECL	MSE	0.1695	0.1713	**0.1657**
	MAE	0.2583	0.2801	**0.2569**
Traffic	MSE	0.5112	0.6271	**0.4858**
	MAE	**0.3258**	0.3530	0.3337

The three-seed results retain encoder-dependent view preferences. On ETTh1, MLPMixer favors Variate+CD (mean MSE 0.3822), whereas Transformer favors Patch+CI (0.3783). On ETTh2, MLPMixer gives almost equal mean MSE for Patch+CI and Variate+CD (both 0.2976 at four decimal places). Variation also depends on the dataset and view: on Illness, the MSE standard deviation for Point+CD is 0.753279 under MLPMixer and 0.446095 under Transformer, compared with 0.015297 and 0.105879 for Patch+CI.

View preferences also depend on which evaluation conditions are averaged. At the fixed configurations, GRU, SSM, and FFT all favor Point+CD on Weather, although their configuration distributions favor Variate+CD. On Exchange, the fixed FFT configuration favors Patch+CI, while its distributional mean favors Variate+CD. SSM switches from Patch+CI in the ETTm2 configuration distribution to Variate+CD at the fixed setting. These differences show why configuration variation and seed variation answer different questions: the former describes performance over modeling choices, and the latter describes repeated training at one such choice.

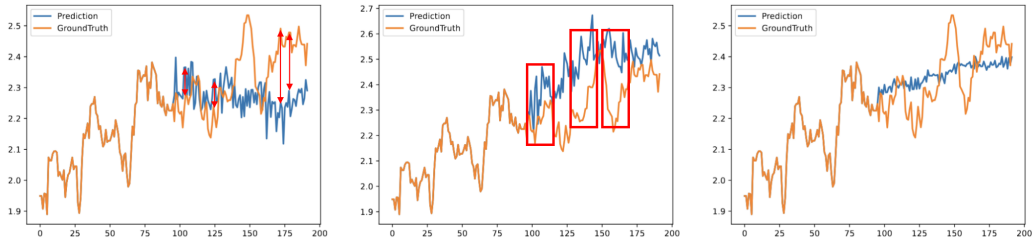
The error criterion matters as well. Under the five-epoch fixed configurations on ECL, Point+CD has the lowest mean MSE for GRU, SSM, and FFT, while Patch+CI has the lowest mean MAE for all three. The view-wise MSE and MAE distributions are given in Tables 18–20.

The CNN fixed configuration favors Patch+CI in MSE on ETTh1, ETTh2, ETTm1, ETTm2, Exchange, and Illness; it favors Point+CD on Weather and Variate+CD on ECL and Traffic. On Traffic, Patch+CI has the lowest MAE even though Variate+CD has the lowest MSE. These are view comparisons at seed 2021 under the stated configuration.

D.3 Prediction Visualization

Figure 5 presents sample-level prediction results of three Embedding strategies combined with Transformer on the Exchange dataset, where Patch-wise Embedding uses the CI setting and Point-wise Embedding uses the CD setting. Since this dataset contains strong trend variations and local fluctuations, it provides a clear example for illustrating the different prediction preferences induced by Point-wise, Patch-wise, and Variate-wise observation views.

Point-wise Embedding preserves the finest-grained temporal information, and thus tends to produce stronger responses at individual local time points. However, its overall trend fitting may be less stable. Patch-wise Embedding aggregates adjacent time steps into local segments, making the model more inclined to capture locally continuous structures. As a result, its predictions exhibit more evident segment-level matching patterns. Variate-wise Embedding treats the entire historical trajectory of each variable as the modeling object, leading to smoother prediction curves that emphasize the global trend rather than sharp local fluctuations.



(a) Exchange: Point-wise (b) Exchange: Patch-wise (c) Exchange: Variate-wise

Figure 5: Prediction visualization of different Embedding strategies on Exchange. (a) Point-wise Embedding performs point-level matching. Some highlighted regions show that even when the overall trend is not well aligned, individual time points can still exhibit locally high correspondence. (b) Patch-wise Embedding performs segment-level matching. The highlighted boxes indicate that the prediction tends to match local segments or patches rather than isolated points. (c) Variate-wise Embedding performs whole-series matching. The prediction is smoother and mainly reflects the global trend of the target sequence. These results illustrate that different Embedding strategies induce different prediction preferences by changing the basic observation unit of the model.

D.4 Configuration Sweep Protocols

The configuration distributions in Appendix D.1 use the shared sweep axes defined in Appendix B.2: input lengths $\{96, 336\}$, hidden dimensions $\{64, 128, 256\}$, and learning rates $\{10^{-4}, 5 \times 10^{-4}, 10^{-3}, 5 \times 10^{-3}, 10^{-2}, 5 \times 10^{-2}\}$. The evaluated subsets, training budgets, and available views vary by encoder. Architectural details are given in Appendix B.1.

MLPMixer and Transformer. The configuration distributions use MAE training loss on an NVIDIA RTX 5090 GPU with 32GB memory. Tables 8 and 9 report the evaluated Point+CI, Point+CD, Patch+CI, Patch+CD, and Variate+CD views. A zero configuration count indicates an unavailable result. MLPMixer has one observed Point+CI configuration on ECL; its mean and minimum are reported, while its sample standard deviation is undefined.

GRU, diagonal SSM, and FFT. These sweeps use seed 2021 and prediction length 96. The seven small and medium datasets use a 31-configuration search, batch size 256, and mixed precision. ECL and Traffic use a 12-configuration search with a five-epoch budget. Their available Point+CI results are included alongside the other views in Tables 10–12; slower configurations were pruned during the search, so coverage varies across encoders and views. All input lengths enter the summaries. Table 22 gives the retained counts for the preferred view of every encoder on all nine datasets. The Point+CI configurations in Table 14 use mixed precision with the encoder and prediction head computed in FP32, input length 336, prediction length 96, seed 2021, batch size 256, and a 100-epoch budget with patience 10. All 19 successful configurations under this numerical precision setting are reported without IQR filtering.

CNN. We schedule 24 configurations per dataset and view as a subset of the Cartesian product of the shared sweep axes. All runs use seed 2021, prediction length 96, Adam with MSE training loss, RevIN, mixed precision, dropout 0.1, and at most ten training epochs. The checkpoint is selected by validation MSE. Patch+CI uses patch length 16 and stride 8. Batch size is 256 for ETTh1, ETTh2, ETTm1, ETTm2, Exchange, Illness, and Weather, and 32 for ECL and Traffic. The sweep contains 626 completed runs from 648 scheduled configurations, of which 533 remain after filtering. Table 23 gives the view-specific coverage.

Filtering and summary statistics. For the GRU, diagonal SSM, FFT, and CNN sweeps in Tables 10–13, one pass of the two-sided $1.5\,\mathrm{IQR}$ rule pools the available views within each encoder–dataset group and removes runs that are outliers in either MSE or MAE. The retained runs are then grouped by view to compute the mean, sample standard deviation with denominator $N - 1$, and minimum for both test MSE and MAE. For Table 2, we select the view with the lowest mean MSE and report all three statistics from that same view. The standard deviation describes variation across configurations. The random-seed results in Appendix D.2 instead summarize repeated training at a fixed configuration.

Table 22: Retained configuration counts for the view with the lowest mean MSE within each dataset–encoder sweep. The corresponding distributions appear in Tables 8–13. ECL and Traffic use five training epochs for GRU, diagonal SSM, and FFT.

Dataset	MLPMixer	Transformer	GRU	SSM	FFT	CNN
ETTh1	28	23	28	24	29	24
ETTh2	23	22	24	22	26	22
ETTm1	24	24	26	25	29	20
ETTm2	28	27	27	25	29	21
Exchange	25	25	26	23	29	22
Illness	17	22	29	12	29	24
Weather	27	27	27	21	29	24
ECL	21	26	10	11	8	24
Traffic	23	24	11	11	11	24

Table 23: CNN coverage: retained/successful configurations, out of 24 scheduled for every dataset–view pair. The retained count is after the joint MSE/MAE IQR filter.

Dataset	Variate+CD	Point+CD	Patch+CI
ETTh1	24/24	15/24	22/24
ETTh2	24/24	16/24	22/24
ETTm1	24/24	14/24	20/24
ETTm2	24/24	15/24	21/24
Exchange	24/24	16/24	22/24
Illness	24/24	11/24	22/24
Weather	24/24	15/24	22/24
ECL	24/24	16/20	21/24
Traffic	24/24	8/9	19/21

E Rank Structure and Predictive Readability for RQ3

Embedding–encoder combinations differ both in the distribution of latent energy and in how readily their representations support forecasting. Effective rank describes spectral concentration, while linear probes measure access to future targets. This appendix presents spectral summaries, fixed-decoder truncation curves, and frozen linear probes, with the configuration coverage specified for each analysis.

E.1 Analysis Setup and Notation

The rank collection contains 116 configurations across ETTh1, ETTh2, ETTm1, ETTm2, Exchange, Weather, ECL, and Traffic with MLPMixer, Transformer, GRU, Mamba, and FFT. Each dataset–encoder pair uses Patch+CI, Variate+CD, and Point, except that Mamba and FFT have no Point results on ECL and Traffic. Point uses CD for MLPMixer and Transformer on ETTh1, Weather, and Exchange, and CI elsewhere. The fixed-decoder truncation and linear-probe analyses use the 98 checkpoints whose Point view uses CI, and truncation summaries cover the 18 configurations whose Point view uses CD. The five encoder families are analyzed jointly, with the Point channel setting identifying the experimental configuration. The three-dataset configurations whose Point view uses CD contain 64 sampled windows on ETTh1 and Weather and 56 on Exchange; the other 98 configurations use the shared 64-window protocol described below.

Given a dataset s, an Embedding strategy m, and an Encoder ϕ, we denote the Encoder output tensor as

$$\mathcal{H}_{m,\phi,s} \in \mathbb{R}^{B_s \times T_{m,s} \times d_h},$$

where B_s denotes the number of validation windows sampled from dataset s, $T_{m,s}$ denotes the number of latent observations produced by a single sample under Embedding strategy m, and d_h denotes the hidden dimension. Here, $T_{m,s}$ varies with the Embedding form. For example, Patch+CI contains both the variate dimension and the patch dimension, Variate+CD corresponds to variate-level tokens, and Point+CD usually corresponds to time-step tokens.

To conduct a unified rank-structure analysis, we flatten the batch dimension and the token-like dimension into a two-dimensional representation matrix:

$$H_{m,\phi,s} = \mathrm{reshape}(\mathcal{H}_{m,\phi,s}) \in \mathbb{R}^{N_{m,s} \times d_h}, \qquad N_{m,s} = B_s T_{m,s}.$$

To analyze the rank structure of the representation, we first center $H_{m,\phi,s}$ along the latent-observation dimension:

$$\bar{H}_{m,\phi,s} = H_{m,\phi,s} - \mathbf{1}\mu_H^\top,$$

where μ_H is the mean feature vector over all latent observations. We then perform singular value decomposition on the centered representation matrix:

$$\bar{H}_{m,\phi,s} = U\Sigma V^\top, \qquad \Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r),$$

where σ_i denotes the i-th singular value and satisfies

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \geq 0.$$

Table 24: Encoder-output rank metrics for ETTh1 and ETTh2. N_{latent} counts flattened representation vectors. r_{50}, r_{95}, and r_{99} count directions retaining the specified energy proportion. Together, Tables 24–27 cover all 116 configurations.

Dataset	Embedding	Encoder	N_{latent}	r_{stable}	r_{eff}	r_{50}	r_{95}	r_{99}
ETTh1	Patch+CI	MLPMixer	5376	1.70	2.34	1	2	4
ETTh1	Variate+CD	MLPMixer	448	2.64	8.91	2	16	40
ETTh1	Point+CD	MLPMixer	6144	1.01	1.11	1	1	2
ETTh1	Patch+CI	Transformer	5376	1.37	2.25	1	2	9
ETTh1	Variate+CD	Transformer	448	2.82	10.25	3	18	46
ETTh1	Point+CD	Transformer	6144	1.60	6.10	1	24	55
ETTh1	Patch+CI	GRU	5376	4.15	14.64	3	27	64
ETTh1	Variate+CD	GRU	448	4.07	14.26	3	32	75
ETTh1	Point+CI	GRU	43008	1.37	3.89	1	14	36
ETTh1	Patch+CI	Mamba	5376	4.58	15.69	3	29	57
ETTh1	Variate+CD	Mamba	448	3.99	12.76	3	30	71
ETTh1	Point+CI	Mamba	43008	1.59	6.09	1	21	30
ETTh1	Patch+CI	FFT	5376	3.99	13.29	3	21	48
ETTh1	Variate+CD	FFT	448	4.54	12.41	3	19	46
ETTh1	Point+CI	FFT	43008	2.45	5.20	2	7	16
ETTh2	Patch+CI	MLPMixer	5376	3.32	10.08	2	15	25
ETTh2	Variate+CD	MLPMixer	448	3.39	7.93	2	22	54
ETTh2	Point+CI	MLPMixer	43008	1.42	4.04	1	16	29
ETTh2	Patch+CI	Transformer	5376	2.43	7.01	2	14	24
ETTh2	Variate+CD	Transformer	448	2.45	5.04	2	8	35
ETTh2	Point+CI	Transformer	43008	1.23	2.51	1	7	23
ETTh2	Patch+CI	GRU	5376	3.49	13.46	2	29	67
ETTh2	Variate+CD	GRU	448	3.26	9.74	2	30	72
ETTh2	Point+CI	GRU	43008	1.48	4.93	1	18	42
ETTh2	Patch+CI	Mamba	5376	4.34	14.20	3	19	33
ETTh2	Variate+CD	Mamba	448	4.31	16.47	3	44	80
ETTh2	Point+CI	Mamba	43008	1.75	7.58	1	23	33
ETTh2	Patch+CI	FFT	5376	2.48	8.32	2	17	39
ETTh2	Variate+CD	FFT	448	2.80	6.08	2	10	35
ETTh2	Point+CI	FFT	43008	2.36	5.79	2	8	17

E.2 Model-side Rank Metrics

Effective rank of Encoder outputs. Let $\lambda_i = \sigma_i^2$ and $p_i = \frac{\lambda_i}{\sum_j \lambda_j}$, where λ_i denotes the energy along the i-th singular direction and p_i denotes its normalized energy proportion. Following the spectral-entropy-based definition of effective rank (Roy & Vetterli, 2007), we define the effective rank of the Encoder output representation matrix H as

$$r_{\text{eff}}(H) = \exp\left(-\sum_i p_i \log p_i\right).$$

This covariance-energy version of effective rank describes spectral concentration. A small value places most energy in a few directions, whereas a larger value spreads energy more evenly. The statistic does not identify which directions the forecasting target requires.

Stable rank. We also compute the stable rank:

$$r_{\text{stable}}(H) = \frac{\sum_i \sigma_i^2}{\sigma_1^2},$$

which measures total spectral energy relative to the energy in the leading direction. In addition, we define the minimum number of singular directions required to reach a cumulative energy ratio q:

$$r_q(H) = \min\left\{k : \frac{\sum_{i=1}^k \lambda_i}{\sum_j \lambda_j} \geq q\right\}, \qquad q \in \{0.50, 0.95, 0.99\}.$$

Here, r_{50}, r_{95}, and r_{99} denote the numbers of singular directions required to explain 50%, 95%, and 99% of the total energy, respectively.

Table 25: Encoder-output rank metrics for ETTm1 and ETTm2. N_{latent} counts flattened representation vectors. r_{50}, r_{95}, and r_{99} count directions retaining the specified energy proportion. Together, Tables 24–27 cover all 116 configurations.

Dataset	Embedding	Encoder	N_{latent}	r_{stable}	r_{eff}	r_{50}	r_{95}	r_{99}
ETTm1	Patch+CI	MLPMixer	5376	2.37	8.87	2	14	33
ETTm1	Variate+CD	MLPMixer	448	3.54	11.82	2	26	64
ETTm1	Point+CI	MLPMixer	43008	1.75	3.61	1	7	23
ETTm1	Patch+CI	Transformer	5376	2.75	13.12	3	23	51
ETTm1	Variate+CD	Transformer	448	3.52	12.23	2	23	52
ETTm1	Point+CI	Transformer	43008	1.27	3.08	1	13	29
ETTm1	Patch+CI	GRU	5376	4.08	21.01	4	46	87
ETTm1	Variate+CD	GRU	448	5.01	22.56	4	51	92
ETTm1	Point+CI	GRU	43008	1.46	4.47	1	16	50
ETTm1	Patch+CI	Mamba	5376	3.51	18.90	3	39	75
ETTm1	Variate+CD	Mamba	448	3.48	11.61	2	26	57
ETTm1	Point+CI	Mamba	43008	2.70	10.48	2	20	45
ETTm1	Patch+CI	FFT	5376	2.46	13.35	2	33	69
ETTm1	Variate+CD	FFT	448	4.10	14.18	3	26	59
ETTm1	Point+CI	FFT	43008	2.01	7.78	2	20	48
ETTm2	Patch+CI	MLPMixer	5376	2.47	8.60	2	14	31
ETTm2	Variate+CD	MLPMixer	448	2.95	8.10	2	17	48
ETTm2	Point+CI	MLPMixer	43008	1.39	2.85	1	7	23
ETTm2	Patch+CI	Transformer	5376	2.31	8.28	2	16	39
ETTm2	Variate+CD	Transformer	448	2.83	7.12	2	11	27
ETTm2	Point+CI	Transformer	43008	1.29	3.14	1	11	30
ETTm2	Patch+CI	GRU	5376	3.59	16.37	3	41	80
ETTm2	Variate+CD	GRU	448	3.60	16.89	2	47	91
ETTm2	Point+CI	GRU	43008	1.53	5.44	1	21	53
ETTm2	Patch+CI	Mamba	5376	2.86	15.11	3	34	67
ETTm2	Variate+CD	Mamba	448	2.97	10.95	2	33	72
ETTm2	Point+CI	Mamba	43008	1.68	5.43	1	13	35
ETTm2	Patch+CI	FFT	5376	1.88	5.62	1	10	31
ETTm2	Variate+CD	FFT	448	2.66	7.26	2	13	37
ETTm2	Point+CI	FFT	43008	2.03	5.66	2	12	37

Tables 24–27 report the complete rank collection. Across the 36 dataset–encoder groups with all three views, Patch and Variate have higher effective ranks than Point in 30 and 34 groups, respectively. The magnitude depends on the encoder and dataset: on Traffic, Patch+CI ranges from 17.13 under FFT to 32.20 under GRU. These contrasts describe the spectra of the collected tokens; their predictive roles also depend on how the decoder reads those tokens.

Table 26: Encoder-output rank metrics for Exchange and Weather. N_{latent} counts flattened representation vectors. r_{50}, r_{95}, and r_{99} count directions retaining the specified energy proportion. Together, Tables 24–27 cover all 116 configurations.

Dataset	Embedding	Encoder	N_{latent}	r_{stable}	r_{eff}	r_{50}	r_{95}	r_{99}
Exchange	Patch+CI	MLPMixer	5376	1.09	1.52	1	2	5
Exchange	Variate+CD	MLPMixer	448	1.75	2.69	1	3	4
Exchange	Point+CD	MLPMixer	5376	1.43	2.21	1	2	7
Exchange	Patch+CI	Transformer	5376	1.03	1.20	1	1	3
Exchange	Variate+CD	Transformer	448	1.51	2.75	1	3	5
Exchange	Point+CD	Transformer	5376	1.02	1.10	1	1	2
Exchange	Patch+CI	GRU	6144	1.92	5.81	1	12	38
Exchange	Variate+CD	GRU	512	2.36	7.75	2	18	55
Exchange	Point+CI	GRU	49152	1.55	5.51	1	18	35
Exchange	Patch+CI	Mamba	6144	2.23	7.58	2	13	29
Exchange	Variate+CD	Mamba	512	2.35	7.19	2	21	59
Exchange	Point+CI	Mamba	49152	1.90	8.71	1	22	30
Exchange	Patch+CI	FFT	6144	1.74	4.15	1	6	16
Exchange	Variate+CD	FFT	512	1.78	3.19	1	3	17
Exchange	Point+CI	FFT	49152	2.31	6.97	2	11	29
Weather	Patch+CI	MLPMixer	16128	1.49	2.66	1	3	5
Weather	Variate+CD	MLPMixer	1344	3.25	10.47	2	20	56
Weather	Point+CD	MLPMixer	6144	1.01	1.07	1	1	1
Weather	Patch+CI	Transformer	16128	1.06	1.31	1	2	3
Weather	Variate+CD	Transformer	1344	2.10	5.30	2	8	25
Weather	Point+CD	Transformer	6144	1.28	3.41	1	14	48
Weather	Patch+CI	GRU	16128	3.04	13.33	3	30	69
Weather	Variate+CD	GRU	1344	5.24	18.55	4	38	81
Weather	Point+CI	GRU	129024	1.43	4.23	1	14	43
Weather	Patch+CI	Mamba	16128	2.36	8.32	2	18	46
Weather	Variate+CD	Mamba	1344	3.60	15.50	3	36	71
Weather	Point+CI	Mamba	129024	1.58	4.76	1	11	31
Weather	Patch+CI	FFT	16128	2.40	9.82	2	19	46
Weather	Variate+CD	FFT	1344	2.97	8.89	2	16	46
Weather	Point+CI	FFT	129024	2.22	5.18	2	8	30

Table 27: Encoder-output rank metrics for ECL and Traffic. N_{latent} counts flattened representation vectors. r_{50}, r_{95}, and r_{99} count directions retaining the specified energy proportion. Together, Tables 24–27 cover all 116 configurations.

Dataset	Embedding	Encoder	N_{latent}	r_{stable}	r_{eff}	r_{50}	r_{95}	r_{99}
ECL	Patch+CI	MLPMixer	246528	2.78	8.91	2	17	33
ECL	Variate+CD	MLPMixer	20544	3.67	12.97	2	28	62
ECL	Point+CI	MLPMixer	1972224	1.53	2.81	1	4	20
ECL	Patch+CI	Transformer	246528	3.44	14.89	3	27	47
ECL	Variate+CD	Transformer	20544	3.68	14.73	2	35	63
ECL	Point+CI	Transformer	1972224	1.16	2.26	1	8	25
ECL	Patch+CI	GRU	246528	6.55	28.09	5	51	91
ECL	Variate+CD	GRU	20544	3.32	12.23	2	34	78
ECL	Point+CI	GRU	1972224	1.39	3.87	1	12	43
ECL	Patch+CI	Mamba	246528	3.50	15.59	3	33	68
ECL	Variate+CD	Mamba	20544	3.24	10.53	2	26	58
ECL	Patch+CI	FFT	246528	2.44	9.79	2	23	41
ECL	Variate+CD	FFT	20544	3.38	12.17	2	28	53
Traffic	Patch+CI	MLPMixer	662016	4.68	18.36	4	26	50
Traffic	Variate+CD	MLPMixer	55168	5.71	24.37	5	41	80
Traffic	Point+CI	MLPMixer	5296128	1.66	2.86	1	4	19
Traffic	Patch+CI	Transformer	662016	3.90	19.07	4	34	60
Traffic	Variate+CD	Transformer	55168	6.65	30.21	6	47	77
Traffic	Point+CI	Transformer	5296128	1.42	3.92	1	12	27
Traffic	Patch+CI	GRU	662016	6.67	32.20	6	58	95
Traffic	Variate+CD	GRU	55168	5.38	20.56	4	41	91
Traffic	Point+CI	GRU	5296128	1.71	6.24	1	19	52
Traffic	Patch+CI	Mamba	662016	5.17	24.11	5	41	76
Traffic	Variate+CD	Mamba	55168	4.72	21.09	4	36	64
Traffic	Patch+CI	FFT	662016	3.71	17.13	3	29	50
Traffic	Variate+CD	FFT	55168	6.46	29.85	6	47	77

E.3 Data-Side Spectral References

We compute two data-side effective ranks as spectral references for the representations. They summarize variation across input windows and channels, respectively.

The window-level spectral reference uses the validation input-window matrix. We flatten all input windows into

$$X_{\text{win}}^{(s)} \in \mathbb{R}^{N_{\text{win}} \times (L \cdot C)},$$

where each row corresponds to one input window. Its effective rank is

$$r_{\text{data}}^{\text{win}}(s) = r_{\text{eff}}\left(X_{\text{win}}^{(s)}\right).$$

This metric summarizes how centered window energy is distributed across linear directions.

The channel-level reference uses the cross-channel matrix. We rearrange the validation inputs as

$$X_{\text{ch}}^{(s)} \in \mathbb{R}^{(N_{\text{win}} \cdot L) \times C},$$

where each row corresponds to a multivariate observation at a certain time step. Its effective rank is

$$r_{\text{data}}^{\text{ch}}(s) = r_{\text{eff}}\left(X_{\text{ch}}^{(s)}\right).$$

This metric summarizes the concentration of centered cross-variate covariance energy.

Exchange has the most concentrated input spectrum in Table 28. Traffic has 862 channels but a channel-level effective rank of 10.28, and the corresponding values for ECL are 321 and 7.61. These quantities summarize covariance concentration across the sampled inputs. Figure 6 shows the complete encoder-output rank collection for comparison across views and encoder families.

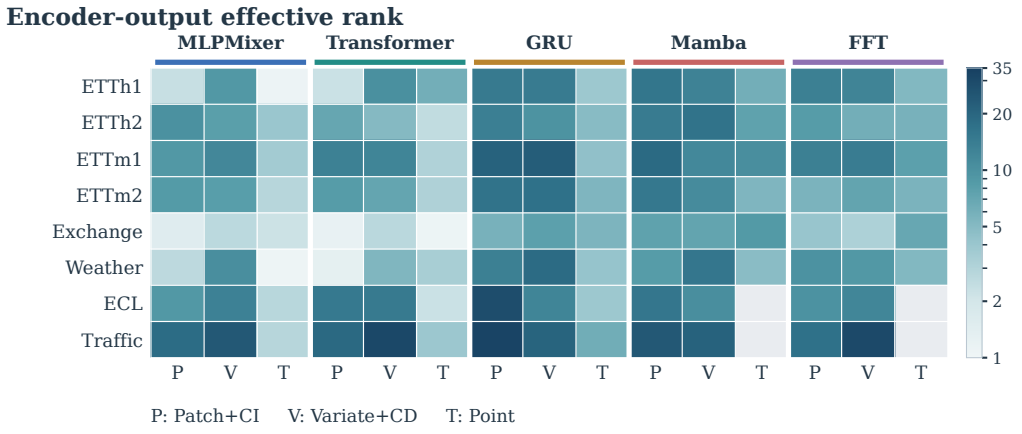


Figure 6: Encoder-output effective ranks for all 116 configurations across eight datasets and five encoder families. P, V, and T denote Patch+CI, Variate+CD, and Point, with the Point channel setting specified in Section E.1. Gray cells mark the four configurations without rank results.

Table 28: Data-side spectral effective ranks on the common 64 validation windows for all eight datasets. The window-level and channel-level ranks summarize the same standardized inputs at two scales.

Dataset	C	$r_{\text{data}}^{\text{win}}$	$r_{\text{data}}^{\text{ch}}$
ETTh1	7	12.61	3.94
ETTh2	7	4.45	2.21
ETTm1	7	8.04	3.90
ETTm2	7	3.45	2.13
Exchange	8	2.78	1.74
Weather	21	13.48	5.62
ECL	321	12.35	7.61
Traffic	862	19.28	10.28

E.4 SVD Rank-Truncation Intervention

The SVD truncation experiment examines which latent directions the trained decoder uses across all five encoder families. For the 98 checkpoints with the common 64-window spectral protocol, we evaluate the same retained-direction grid $k \in \{1, 2, 4, 8, 16, 32, 64, 96, 128\}$. Each representation is projected onto nested spectral subspaces and evaluated with its decoder held fixed. This intervention measures sensitivity to removing directions from a particular trained representation.

Given the Encoder output representation matrix

$$H \in \mathbb{R}^{N_{\text{latent}} \times d_h},$$

we first compute the mean along the latent-observation dimension:

$$\mu_H = \frac{1}{N_{\text{latent}}} \sum_{i=1}^{N_{\text{latent}}} H_i,$$

and obtain the centered representation:

$$\bar{H} = H - \mathbf{1}\mu_H^\top.$$

We then perform singular value decomposition on the centered representation matrix:

$$\bar{H} = U\Sigma V^\top.$$

For a given number of retained directions k, we keep only the top-k right singular directions and project the representation onto the corresponding subspace:

$$\bar{H}_k = \bar{H} V_{:,1:k} V_{:,1:k}^\top.$$

We restore the feature mean so that truncation preserves the representation offset:

$$H_k = \bar{H}_k + \mathbf{1}\mu_H^\top.$$

We then feed H_k into the fixed decoder and record the validation MSE. When $k = d_h$, H_k equals the full encoder output and recovers the full-encoder prediction result.

Figures 7 and 8 show lower validation MSE after truncation in 68 of the 98 checkpoints, using an absolute improvement threshold of 10^{-6}. The median best evaluated count is $k^* = 4$ on ETTh2 and Exchange, but $k^* = 128$ on ECL and Traffic. Traffic has one improving configuration, with a relative decrease of 0.007%, while every Exchange configuration improves and the largest decrease is 7.52%. The count k is distinct from the entropy-based effective rank: a representation can concentrate energy in a few directions while its fixed decoder still uses lower-energy directions.

The 18 configurations on ETTh1, Weather, and Exchange whose Point view uses CD show truncation improvements in 11 cases. Their results include nonmonotonic error curves and substantial sensitivity to removing lower-energy directions for Point+CD with MLPMixer on ETTh1.

Truncation measures how predictions change when selected directions are removed under a fixed decoder. To assess how the encoder changes access to future targets, Section E.6 fits the same family of linear readouts before and after the encoder.

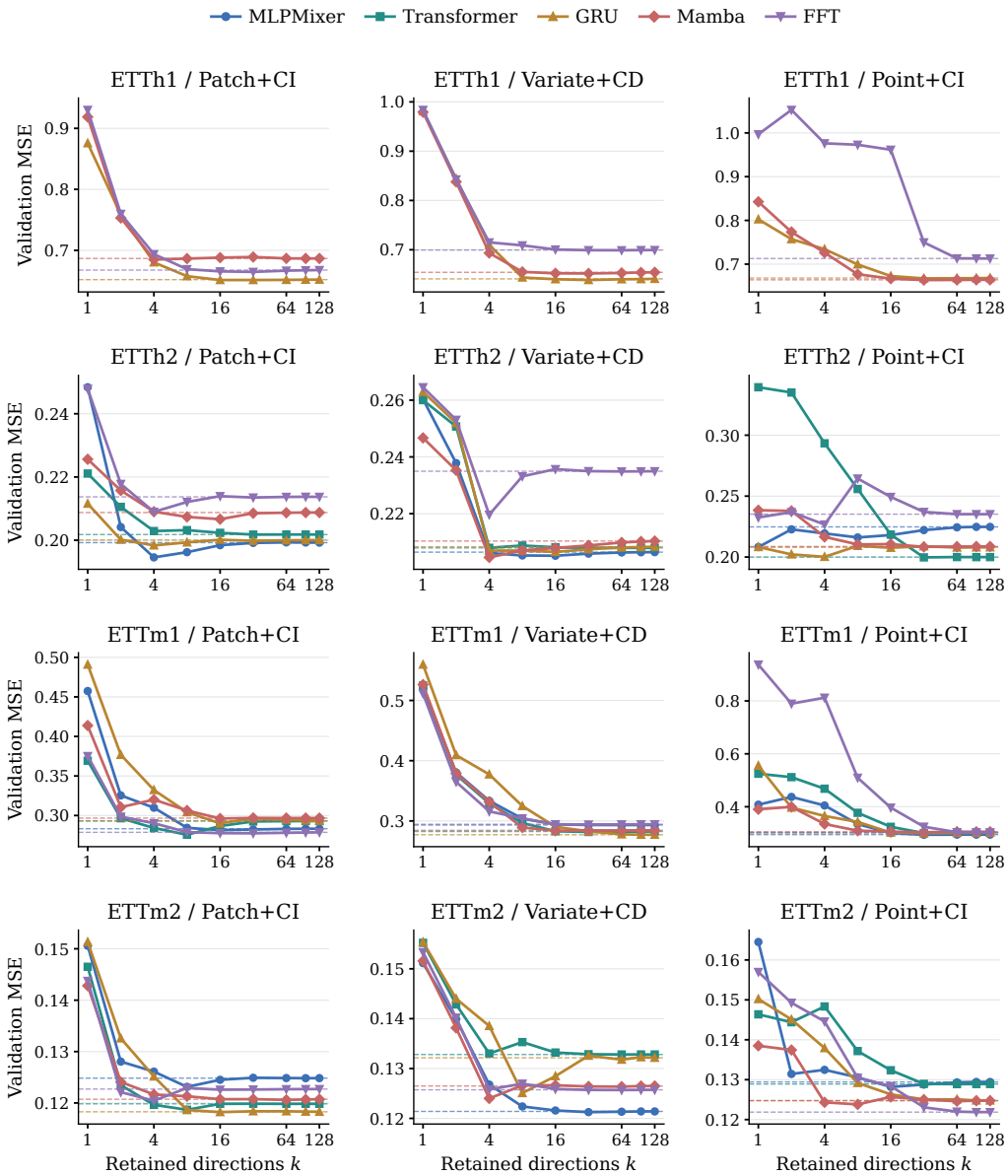


Figure 7: Fixed-decoder SVD rank-truncation curves on the four ETT datasets for the common 64-window protocol. Each panel includes all available encoder families for its embedding view. Dashed lines show the full encoder's validation MSE. The spectral basis and mean are computed over all sampled windows and latent tokens.

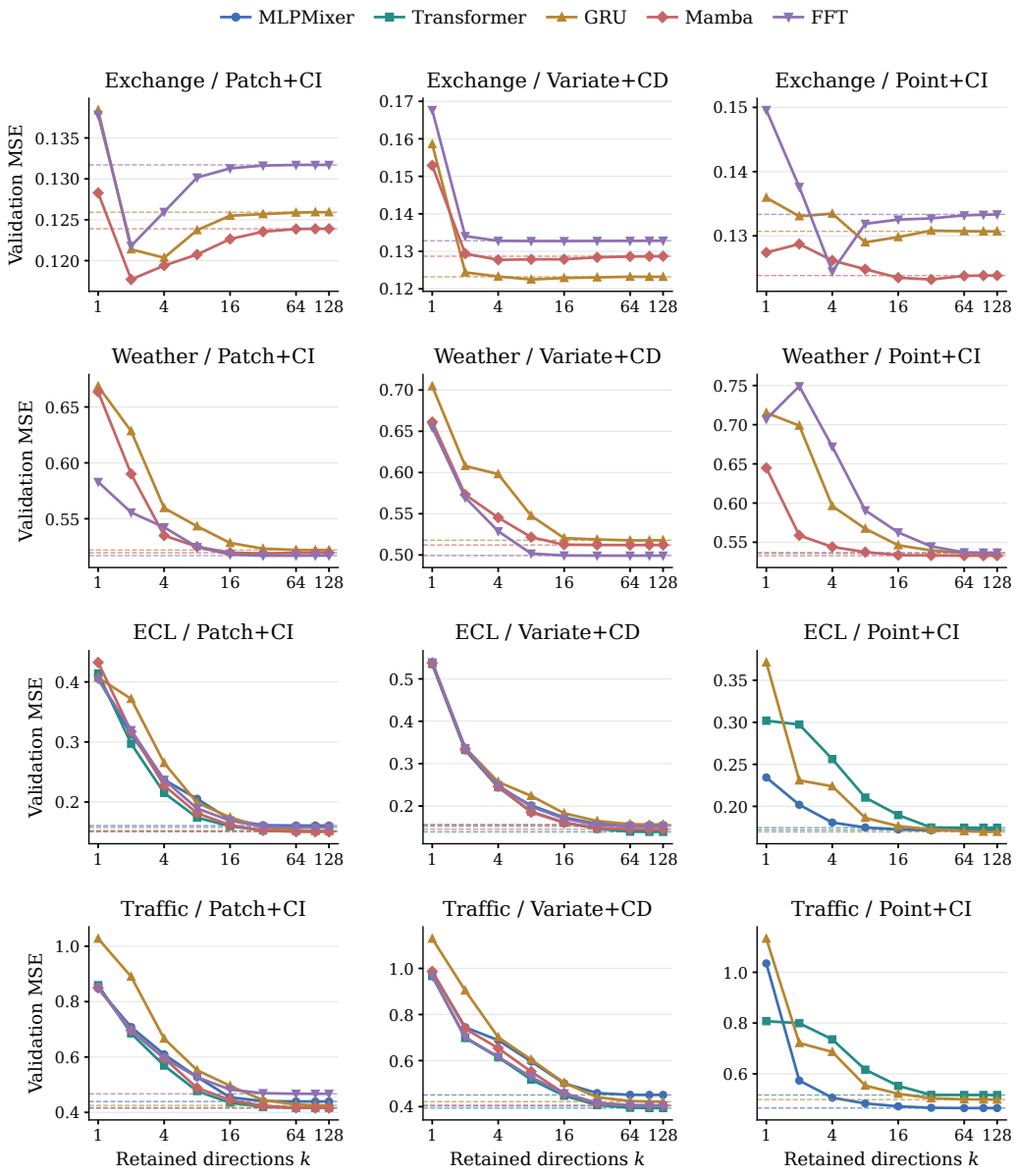


Figure 8: Fixed-decoder SVD rank-truncation curves on Exchange, Weather, ECL, and Traffic, using the same protocol and encodings as Figure 7. Together, the two figures contain all 98 evaluated checkpoints. Mamba and FFT have no Point+CI checkpoints on ECL or Traffic.

E.5 Rank-Truncation Sensitivity and Error Summaries

To summarize the common-protocol curves across encoder families, we define rank-truncation sensitivity as

$$\mathrm{RankSens}(m, \phi, s) = \mathrm{std}_{k \in \mathcal{K}} \left[\mathrm{MSE}(H_k) \right], \qquad \mathcal{K} = \{1, 2, 4, 8, 16, 32, 64, 96, 128\}.$$

A larger value indicates greater variation in validation error across the evaluated truncation levels under the fixed decoder. The best evaluated error is

$$\mathrm{MSE}_{\min}(m, \phi, s) = \min_{k \in \mathcal{K}} \mathrm{MSE}(H_k).$$

Table 29 summarizes all 98 checkpoints. Each family contains configurations that improve after removing some directions, with the median reduction varying from 0.007% for Transformer to 0.199% for Mamba over their respective evaluated configurations.

Across the 98 checkpoints, full-representation validation MSE and the best evaluated rank-truncated MSE have Pearson and Spearman correlations of 0.9998 and 0.9957. Their similar ordering reflects the small truncation gains for many configurations, while the full curves show where strong truncation removes useful directions.

The 18 three-dataset configurations whose Point view uses CD also have associations with stability $\hat{\sigma}$ and best potential L_{best}. The Pearson and Spearman coefficients are 0.658 and 0.181 for $\hat{\sigma}$ versus $\mathrm{RankSens}$, and 0.973 and 0.950 for L_{best} versus $\mathrm{MSE}_{\min}$. These statistics summarize variation across evaluation conditions and under truncation for those configurations. The five-family analysis above instead compares interventions under a common validation-window protocol.

E.6 Linear Probes of Predictive Readability

Checkpoint collection and spectral comparison. The 116-configuration rank collection uses the five encoder families and embedding settings in Section E.1. FFT auto-correlation covers Patch+CI and Variate+CD on ECL and Traffic; FFT-attention covers the other six datasets. Figure 2 groups these implementations as FFT. The linear probes cover 98 checkpoints: 15 MLPMixer, 15 Transformer, 24 GRU, 22 Mamba, 18 FFT-attention, and 4 FFT auto-correlation configurations, all using Patch+CI, Variate+CD, or Point+CI.

The checkpoint spectral analysis uses 64 validation windows per dataset, sampled without replacement with seed 2021 and shared across configurations. Input and prediction lengths are 96, and the hidden dimension is 128. Forecast targets lie in the validation segment; input histories can include the training segment's final 96 observations. After native forward passes, we flatten encoder-output windows and tokens, center each feature, and compute the covariance-energy effective rank. The channel-level data rank uses the centered $(64 \cdot 96) \times C$ input matrix after training-fitted channel standardization.

Frozen representations and readouts. Following independently trained linear probes (Alain & Bengio, 2016), we fit a multivariate ridge readout to each of three stages: the historical sequence after the checkpoint's RevIN normalization (Raw), the embedding output (E), and the encoder output (H). All checkpoint parameters remain fixed. Each readout maps a representation for one window and channel to the next 96 values. It uses all coordinates in the native prediction head's flattening order: 96 for Raw, and 1536, 12288, or 128 for Patch+CI, Point+CI, or Variate+CD. Within each checkpoint, the E and H probes have the same feature dimension and use the same training and validation rows.

For Variate+CD, the E readout uses the current channel's token, while its H token can incorporate other channels through the encoder. Its measured gain therefore includes cross-channel integration as well as changes in linear readability. Patch+CI and Point+CI process the current channel's history. These readouts measure the contribution available to the native channel-wise prediction head.

Temporal sampling and readout budget. Input and prediction lengths are both 96. We sample 128 training, 64 validation, and 64 test windows using seeds 31001, 31002, and 31003, respectively. Every complete input–target window lies inside its chronological split. We use the ETT benchmark split boundaries, and global channel standardization is fitted on the training segment. Model forward passes retain all channels. From the sampled training windows, we choose at most 1024 distinct

Table 29: Fixed-decoder truncation summaries by encoder for the 98 checkpoints. Improvement requires a validation MSE decrease greater than 10^{-6} from the full representation. The last three columns report medians over configurations. Relative reduction is $100(1 - \mathrm{MSE}_{\min}/\mathrm{MSE}_{\mathrm{full}})$.

Encoder	n	Improved	RankSens	k^*	Reduction (%)
MLPMixer	15	10	0.0541	32	0.178
Transformer	15	9	0.0559	64	0.007
GRU	24	16	0.0614	32	0.077
Mamba	22	18	0.0362	32	0.199
FFT	22	15	0.0401	48	0.049

window–channel rows in a fixed order balanced across channels and approximately across windows; validation uses at most 1024 rows. The ETT datasets contribute all 896 available training rows, and Exchange contributes 1024. All configurations of a dataset use identical row indices. Testing averages over all channels of the 64 test windows.

RevIN-aligned ridge regression. For row i, let μ_i and s_i be the mean and scale computed from its historical input, and let a_i and b_i be the checkpoint's channel-specific RevIN affine parameters. With the native inverse denominator $\delta_i = a_i + \epsilon^2$, the transformed future target and its error weight are

$$t_i = \frac{\delta_i}{s_i}(y_i - \mu_i \mathbf{1}) + b_i \mathbf{1}, \qquad w_i = \left(\frac{s_i}{\delta_i}\right)^2. \tag{21}$$

For a predicted $\widehat{t}_i$, inverse RevIN gives $\widehat{y}_i$, with

$$\|\widehat{y}_i - y_i\|_2^2 = w_i \|\widehat{t}_i - t_i\|_2^2. \tag{22}$$

Thus, the probe loss is aligned with forecasting MSE in the dataset coordinates obtained by training-segment standardization. The normalization statistics depend on the historical input.

For a feature matrix Z with n training rows, normalize the training weights to $\bar{w}_i = w_i/(n^{-1}\sum_j w_j)$. Let μ_Z and μ_T be weighted training means, and define centered, weighted rows $\widetilde{Z}_i = \sqrt{\bar{w}_i}(Z_i - \mu_Z)$ and $\widetilde{T}_i = \sqrt{\bar{w}_i}(t_i - \mu_T)$. The readout solves

$$\widehat{B}_\alpha = \arg\min_B \frac{\|\widetilde{T} - \widetilde{Z}B\|_F^2 + \alpha e_Z \|B\|_F^2}{96n}, \qquad e_Z = \frac{\|\widetilde{Z}\|_F^2}{n}, \tag{23}$$

and predicts $\widehat{t} = (z - \mu_Z)\widehat{B}_\alpha + \mu_T$. We use the fixed grid $\alpha \in \{10^{-6}, 10^{-4}, 10^{-2}, 1, 100\}$, selecting each stage's value by validation MSE after inverse RevIN. Validation weights are not rescaled. The selected readout is evaluated on the test windows without refitting on validation. The feature-energy factor e_Z makes this rule invariant to a common rescaling of feature coordinates.

Encoder contribution. Let $\mathcal{E}_E$ and $\mathcal{E}_H$ be the corresponding probe test MSEs, each averaged over the same windows, channels, and forecast steps. We define

$$G = \log \frac{\mathcal{E}_E}{\mathcal{E}_H}. \tag{24}$$

Positive G means that the encoder output supports lower probe error at the specified readout budget. It measures predictive readability under that fitting rule, including its sample size and regularization.

Dataset and view dependence. The H probe improves over E in 72 of 98 configurations. Table 31 reports positive mean gains for each of the five encoder families over its evaluated configurations. Table 30 shows the strongest mean gains on Traffic and ECL, while all nine Exchange configurations have higher H-probe error. ETTm1 improves in all 15 configurations. ETTh1 and ETTm2 have small increases in their geometric mean error despite containing improving configurations. Averaging G equally over the eight datasets gives a geometric error reduction of 5.62%, with the dataset-specific results distinguishing where the encoder contributes most.

For a matched comparison of views, we retain the 30 Dataset×Encoder groups containing all three views. We first average G across encoders within a dataset and then weight the eight datasets equally. Patch+CI improves in 27 of 30 groups, with a geometric error reduction of 5.95%; Variate+CD

Table 30: Linear-probe results by dataset. n is the number of checkpoints; $G > 0$ counts checkpoints with lower H-probe error. The final column is $100[1 - \exp(-\overline{G}_s)]$, with G averaged equally over configurations within dataset s. Positive values indicate lower error.

Dataset	n	$G > 0$	$\overline{G}_s$	**Error reduction (%)**
ETTh1	9	5	-0.00829	-0.83
ETTh2	15	13	$+0.02548$	$+2.52$
ETTm1	15	15	$+0.08755$	$+8.38$
ETTm2	15	7	-0.00522	-0.52
Exchange	9	0	-0.03540	-3.60
Weather	9	7	$+0.03379$	$+3.32$
ECL	13	12	$+0.14086$	$+13.14$
Traffic	13	13	$+0.22396$	$+20.07$

Table 31: Linear-probe results by encoder family for all 98 checkpoints. n counts checkpoints, and $G > 0$ counts lower encoder-output probe error. Error reduction is $100[1 - \exp(-\overline{G})]$, with equal configuration weights within each encoder.

Encoder	n	$G > 0$	$\overline{G}$	Error reduction (%)
MLPMixer	15	13	$+0.08390$	$+8.05$
Transformer	15	13	$+0.09732$	$+9.27$
GRU	24	16	$+0.04657$	$+4.55$
Mamba	22	17	$+0.06577$	$+6.37$
FFT	22	13	$+0.04489$	$+4.39$

improves in 23 of 30, with a reduction of 9.22%. Point+CI improves in 14 of 30, and its aggregate error reduction is -0.04%. The near-zero Point average combines improvements on ETTm1 and Traffic with declines on ETTh1 and ETTm2.

Readability and spectral change. For the 98 probed checkpoints, write the encoder's spectral compression as $C_{\mathrm{rank}} = \log[r_{\mathrm{eff}}(E)/r_{\mathrm{eff}}(H)]$. Among 37 configurations with $C_{\mathrm{rank}} > 0$, 26 improve in probe error and 11 worsen. Among 61 configurations with spectral expansion, 46 improve and 15 worsen. After subtracting Dataset×Encoder means from both C_{rank} and G, their Pearson and Spearman correlations are -0.027 and $+0.020$. Compression and expansion can therefore both accompany improved readability. Spectral change describes energy allocation, while the probe tests how the resulting representation supports the target.

The two analyses use different representation matrices. Spectral ranks stack latent tokens as rows with 128 feature coordinates on 64 validation windows. The probes concatenate all coordinates read by the native head into each window–channel row and use their own fixed temporal samples. Their agreement or disagreement relates a token-spectrum summary to full-readout performance.

Budget sensitivity. A nested enlargement from 1024 to 2048 training rows is available for the 35 Weather, ECL, and Traffic configurations. All 35 retain the sign of G. Their geometric error reductions change from 3.32% to 5.21% on Weather, from 13.14% to 14.22% on ECL, and from 20.07% to 20.76% on Traffic. The other datasets already use all rows in the sampled training windows. This comparison evaluates a larger fitting budget under the same validation-based regularization rule.

Point embedding as a readout control. Point+CI expands each scalar through a shared linear embedding and adds fixed position and bias terms. After centering, this is a scaled isometric expansion of the Raw history, so feature-energy-normalized ridge readouts have the same prediction path. Across all 30 Point+CI checkpoints and 39 distinct checkpoint–budget fits, the selected regularization values agree and the largest relative difference in test MSE is 1.03×10^{-9}. A different token spectrum can thus coexist with essentially unchanged full-readout predictive performance.

E.7 Dataset Descriptors and Marginal Error

Figure 3 (MSE) and Figure 9 (MAE) use the RQ2 configuration distributions for MLPMixer, Transformer, GRU, diagonal SSM, FFT auto-correlation, and CNN. The nine datasets are ETTh1, ETTh2, ETTm1, ETTm2, ECL, Exchange, Illness, Traffic, and Weather. Each dataset contributes one mean error for each encoder and embedding family, giving 162 means for MSE and 162 for MAE. Table 32 lists the seven dataset descriptors; the same descriptor values are used for every encoder and embedding family.

For each encoder and dataset, we pool retained runs within the Patch, Point, or Variate family. When CI and CD settings are both available, their means are weighted by their retained configuration counts, giving the arithmetic mean over all retained runs in that family. Each mean uses the retained configuration runs under its encoder's RQ2 evaluation and filtering protocol (Appendix D.4). The resulting means contain 875 MLPMixer, 880 Transformer, 585 GRU, 506 SSM, 579 FFT, and 533 CNN runs.

For each encoder, embedding family, and error metric, we compute Spearman correlation between the nine dataset means and each descriptor, using average ranks for ties. Every plotted coefficient therefore summarizes nine dataset-level pairs. The two error metrics, six encoders, three embedding families, and seven descriptors give 252 coefficients, with 126 in each figure.

Figure 9 shows the corresponding MAE correlations. Mean MAE preserves the directions observed for MSE, with Trend coefficients from -0.75 to -0.43 and Seasonality coefficients from 0.40 to 0.67 across all 18 encoder–embedding combinations.

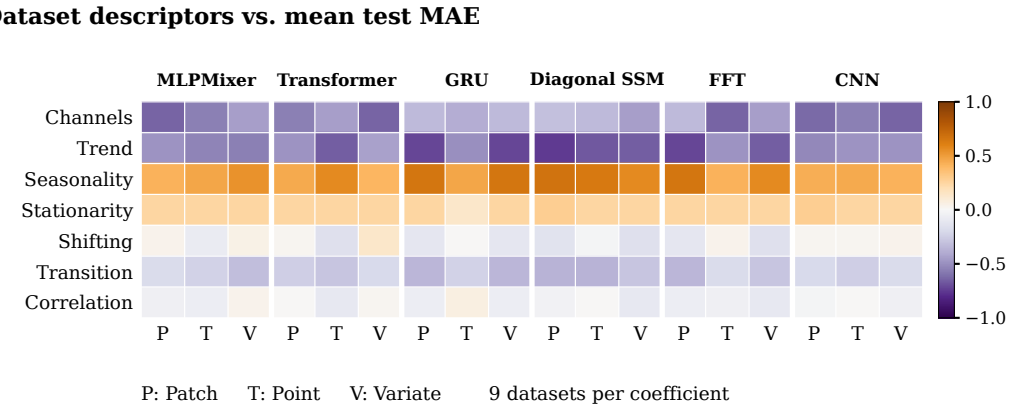


Figure 9: Spearman correlations between seven dataset descriptors and mean test MAE across nine datasets for six RQ2 encoder families. Each embedding-family mean pools its retained configuration runs and available CI/CD settings, using the same aggregation and dataset coverage as Figure 3. Negative values associate larger descriptor values with lower error.

Table 32: Dataset descriptors used in Figures 3 and 9. Values are displayed to three decimals; correlations use the stored precision. Column names match the stored descriptor fields.

Dataset	Channels	Trend	Seasonality	Stationarity	Shifting	Transition	Correlation
ETTh1	7	0.733	0.518	1.000	0.641	0.129	0.115
ETTh2	7	0.928	0.362	1.000	0.832	0.151	0.111
ETTm1	7	0.777	0.422	1.000	0.604	0.164	−0.280
ETTm2	7	0.777	0.454	1.000	0.840	0.147	−0.285
ECL	321	0.667	0.645	1.000	0.611	0.106	−0.334
Exchange	8	0.998	0.118	0.125	0.704	0.211	−0.255
Illness	7	0.476	0.595	0.714	0.834	0.169	−0.348
Traffic	862	0.099	0.678	1.000	0.627	0.122	−0.203
Weather	21	0.797	0.348	0.900	0.697	0.207	−0.196

E.8 Summary of the RQ3 Mechanism

Effective rank describes the concentration of representation energy across directions. The truncation curves identify directions used by a fitted decoder, and the probes quantify how the encoder changes access to future targets under matched readout budgets. Across datasets and views, similar spectral changes can have different predictive effects. These complementary measurements connect data structure, tokenization, and encoder processing.

F Data-Based Embedding–Encoder Selection

The selector in Section 5.4 uses log-rank calibration to choose an embedding and encoder from the target training data. Its source labels come from the Rank-Guarded Data–Rank Selector (DRS-RG), which combines short-run validation performance with a data-view-dependent rank constraint. We first describe how these labels are obtained and report their forecasting results, then specify the multiscale calibration procedure. DRS-RG trains candidate models on the source datasets; the training-free stage is the subsequent selection on a target series.

F.1 DRS-RG: Design and Selection Procedure

The Rank-Guarded Data–Rank Selector (DRS-RG) combines early forecasting performance with the spectral relationship between a data view and its encoded representation. Validation error measures current predictive quality, validation improvement captures early optimization progress, and rank mismatch measures how far the representation's effective dimension lies from its data reference. DRS-RG first uses these quantities to choose a learning rate for each embedding–encoder pair and identify an embedding view. Within that view, it selects an encoder subject to a rank constraint.

Candidate space and training. For each of ETTh1, ETTh2, ETTm1, ETTm2, Exchange, and Weather, we combine three embedding views (Patch+CI, Point+CI, and Variate) with six encoders (Transformer, MLPMixer, GRU, Mamba, FFT-Attention, and None). None is the identity encoder, which passes the embedding output directly to the decoder. Each of the 18 pairs is evaluated at initial learning rates 5×10^{-4} and 10^{-3}, giving 36 three-epoch warm-up runs per dataset and 216 runs in total. All warm-up runs use random seed 2024.

The task is multivariate forecasting with input, prediction, and label lengths 96, 96, and 48. The recorded runs use Adam with MSE loss, batch size 128, mixed-precision training, RevIN, dropout 0.1, GELU, and no data augmentation. The initial learning rate is halved after each epoch starting with the second epoch. The representation width is 128; Transformer uses eight attention heads and a feed-forward width of 2048. The encoder depth is one, and Patch+CI uses patch length 16 and stride 8. Mamba uses state size 16, expansion factor 2, and convolution width 4. All candidates share chronological training, validation, and test splits: $12/4/4$ months of 30 days for ETT, and $70\%/10\%/20\%$ for Exchange and Weather. After learning-rate selection, each pair is trained from initialization for at most 30 epochs with validation early stopping of patience 8, and its best-validation checkpoint is evaluated on the test set. This gives 108 full-training runs for the primary-seed comparison.

Data-view and encoder ranks. We fit per-variable standardization on the training segment and sample $N = 64$ input windows of length 96, using sampling seed 2021. Window starts are drawn without replacement, and the input windows may overlap. For $X \in \mathbb{R}^{N\times 96\times C}$, define

$$\begin{aligned} r_{\text{win}} &= r_{\text{eff}}(\text{reshape}(X, N, 96C)), \\ r_{\text{ch}} &= r_{\text{eff}}(\text{reshape}(X, 96N, C)). \end{aligned} \tag{25}$$

Here r_{eff} is the energy-weighted effective rank in Section 5.3, computed after column centering. Patch+CI and Point+CI use the window reference r_{win}, while Variate uses the channel reference r_{ch}. DRS-RG uses the same 64 windows for all candidates; the multiscale profile in Section 5.4 averages repeated window-rank estimates.

For each candidate, we load the best-validation checkpoint obtained during its three-epoch warm-up and process the same 64 windows in evaluation mode, including the model's RevIN and native

channel routing. We collect the encoder outputs and stack all latent tokens as rows of a matrix $H \in \mathbb{R}^{K\times 128}$. Thus, $K = 64C \cdot 12$ for Patch+CI, $K = 64C \cdot 96$ for Point+CI, and $K = 64C$ for Variate. For None, H contains the embedding outputs. After column centering, the eigenvalues of the 128×128 scatter matrix give $r_{\text{enc}} = r_{\text{eff}}(H)$. Both data and encoder rank calculations retain eigenvalues greater than 10^{-12} times the largest eigenvalue. The mismatch for view v is

$$\begin{aligned} \mathrm{RM}_v &= \left|\log r_{\text{enc}} - \log r^v_{\text{data}}\right|, \\ r^v_{\text{data}} &= \begin{cases} r_{\text{win}}, & v \in \{\text{Patch+CI}, \text{Point+CI}\}, \\ r_{\text{ch}}, & v = \text{Variate}. \end{cases} \end{aligned} \tag{26}$$

Natural logarithms give equal penalties to reciprocal rank ratios. Rank inference disables mixed precision, and scatter matrices and eigendecompositions use double precision.

Warm-up score. Let L_1 and L_3 be the validation MSE recorded at epochs 1 and 3, respectively. The relative validation improvement is $G = (L_1 - L_3)/L_1$. For each statistic, we compute $z(x) = (x - \mu_x)/\sigma_x$ over all 36 candidates of the same dataset, using the population standard deviation. A statistic with standard deviation at most 10^{-12} receives zero standardized scores. The selection score is

$$S = z(\log L_3) + 0.1z(\mathrm{RM}_v) - 0.1z(G). \tag{27}$$

Lower scores favor lower validation error, closer rank matching, and greater early improvement. Standardization places the three terms on comparable scales, with validation error receiving the largest coefficient. The two auxiliary weights are fixed at 0.1 across datasets. Validation losses in the score are the recorded epoch-1 and epoch-3 values, whereas the encoder rank is measured at the best-validation warm-up checkpoint.

Learning-rate and view selection. For each embedding–encoder pair (v, c), retain the learning rate with the smallest score:

$$\eta^*_{v,c} = \arg\min_{\eta\in\{5\times10^{-4},10^{-3}\}} S(v, c, \eta), \qquad S^*_{v,c} = S(v, c, \eta^*_{v,c}). \tag{28}$$

The view v^* is the embedding component of the pair minimizing $S^*_{v,c}$ over all 18 pairs. This step compares each pair at its selected learning rate. Score ties follow encoder-name order and then view-name order. Write $\mathrm{RM}^*_{v,c} = \mathrm{RM}(v, c, \eta^*_{v,c})$ for the corresponding rank mismatch.

Encoder selection with a rank constraint. Within v^*, DRS-RG keeps encoders whose effective rank lies within a factor of e of the data reference. The admissible set and selected encoder are

$$\begin{aligned} \mathcal{A}(v^*) &= \{c : \mathrm{RM}^*_{v^*,c} \le 1\}, \\ c^* &= \begin{cases} \arg\min_{c\in\mathcal{A}(v^*)} S^*_{v^*,c}, & \mathcal{A}(v^*) \neq \emptyset, \\ \arg\min_{c} \mathrm{RM}^*_{v^*,c}, & \mathcal{A}(v^*) = \emptyset. \end{cases} \end{aligned} \tag{29}$$

The selected pair retains $\eta^*_{v^*,c^*}$. The threshold 1 is fixed across datasets and corresponds to $e^{-1} \le r_{\text{enc}}/r^{v^*}_{\text{data}} \le e$. It converts rank matching from a soft score penalty into an admissibility condition within the selected view. If no encoder satisfies the condition, the rule chooses the smallest mismatch. Ties in this fallback are resolved by the warm-up score, and encoder-score ties follow encoder-name order. Learning-rate ties retain the smaller rate. No additional training is needed to apply the constraint.

All decision quantities come from training-window spectra and validation losses. The rank constraint was developed through post-hoc analysis of the candidate runs, and the results below evaluate the resulting rule on those six source datasets. The selected structures provide the empirical calibration labels for the multiscale selector in Section 5.4.

F.2 DRS-RG Selection and Forecasting Results

Table 33 reports the selected pairs at primary seed 2024. For each dataset, the oracle is the pair with the lowest test MSE among the 18 fully trained candidates, each using its own warm-up-selected learning rate. Relative oracle regret is $100(\mathrm{MSE}_{\text{selected}}/\mathrm{MSE}_{\text{oracle}} - 1)\%$. The oracle serves as

Table 33: DRS-RG selections and test performance at seed 2024. Ranks are among 18 pairs per dataset. LR is the initial learning rate, and regret is relative to the lowest test MSE in that candidate set. These selections supply the structural labels in Section 5.4.

Dataset	Embedding / Encoder	LR	MSE	MAE	Rank	Regret (%)
ETTh1	Patch+CI / Mamba	10^{-3}	0.373670	0.396009	1/18	0.0000
ETTh2	Patch+CI / Transformer	10^{-3}	0.284035	0.338102	1/18	0.0000
ETTm1	Variate / MLPMixer	10^{-3}	0.320872	0.359945	1/18	0.0000
ETTm2	Patch+CI / GRU	5×10^{-4}	0.175478	0.261194	1/18	0.0000
Exchange	Patch+CI / None	5×10^{-4}	0.082489	0.198576	2/18	0.0268
Weather	Variate / MLPMixer	10^{-3}	0.160245	0.207264	1/18	0.0000

an evaluation reference. DRS-RG reaches the lowest MSE on five of six datasets, with mean relative regret 0.0045%. On Exchange, it selects Patch+CI with None and ranks second: its MSE is 0.082489, compared with 0.082467 for Patch+CI with Mamba, a relative regret of 0.0268%.

ETTm1 illustrates the role of the rank constraint. The smallest warm-up score identifies Variate with GRU, whose mismatch is 1.725790. Within the same view, MLPMixer has mismatch 0.883720 and the lowest score among admissible encoders, so DRS-RG selects it. Their test MSEs are 0.324794 and 0.320872, respectively, corresponding to ranks $7/18$ and $1/18$. The other five datasets retain the encoder with the smallest warm-up score. The fallback is not activated on these six datasets. Table 35 gives the full candidate MSE comparison.

Table 34 reports means and sample standard deviations across training seeds 2024, 2025, and 2026. The embedding, encoder, and initial learning rate remain fixed at the DRS-RG selection. These runs measure forecasting variability under repeated training of the selected structures. The repeated-window experiment in Section 5.4 instead measures the sampling stability of the subsequent data-based selector.

Table 34: Test performance of the fixed DRS-RG selections across three training seeds. Entries are mean $\pm$ sample standard deviation. Structures and learning rates are those in Table 33.

Dataset	MSE	MAE
ETTh1	0.375208 ± 0.002268	0.397524 ± 0.001485
ETTh2	0.294438 ± 0.009414	0.344694 ± 0.005837
ETTm1	0.320145 ± 0.003082	0.358726 ± 0.002917
ETTm2	0.175900 ± 0.000454	0.261309 ± 0.000250
Exchange	0.081700 ± 0.000706	0.197846 ± 0.000636
Weather	0.159485 ± 0.000716	0.206864 ± 0.000435

Table 35: Test MSE for all 18 embedding–encoder pairs on the six source datasets at seed 2024. Each pair uses its own warm-up-selected learning rate. Bold marks the lowest MSE in each column, and † marks the DRS-RG selection.

Embedding	Encoder	ETTh1	ETTh2	ETTm1	ETTm2	Exchange	Weather
Patch+CI	Transformer	0.375449	**0.284035**†	0.328177	0.177145	0.088303	0.165114
	MLPMixer	0.381171	0.288306	0.326811	0.177219	0.089424	0.167414
	GRU	0.385141	0.289832	0.323532	**0.175478**†	0.083936	0.168030
	Mamba	**0.373670**†	0.296162	0.321474	0.184651	**0.082467**	0.165765
	FFT-Attention	0.377700	0.304864	0.323615	0.180645	0.093929	0.164620
	None	0.383815	0.290228	0.350229	0.182385	0.082489†	0.192972
Point+CI	Transformer	0.394822	0.298301	0.333305	0.186384	0.087899	0.173173
	MLPMixer	0.392133	0.310070	0.334114	0.184241	0.083930	0.174277
	GRU	0.394549	0.298763	0.321484	0.179928	0.084934	0.176660
	Mamba	0.409781	0.297489	0.322323	0.184975	0.087431	0.175024
	FFT-Attention	0.391977	0.307842	0.328626	0.188289	0.088016	0.168927
	None	0.383575	0.288953	0.349843	0.182445	0.083766	0.193001
Variate	Transformer	0.383059	0.297845	0.332513	0.181355	0.089106	0.168742
	MLPMixer	0.379101	0.297041	**0.320872**†	0.178691	0.091865	**0.160245**†
	GRU	0.383110	0.299973	0.324794	0.180476	0.087517	0.167653
	Mamba	0.388428	0.303823	0.325000	0.184335	0.094282	0.168050
	FFT-Attention	0.391137	0.303324	0.338623	0.182826	0.093726	0.165665
	None	0.384955	0.289904	0.349495	0.182795	0.082861	0.193163

F.3 Multiscale Profile Estimation

Repeated window sampling estimates the data-side rank profile. For a training series with C variables, we standardize each variable using its training mean and population standard deviation. At each length $L \in \{48, 96, 192\}$, we draw 64 distinct window starts without replacement from the training segment. Windows may overlap in time. We flatten the windows into $W_{L,b} \in \mathbb{R}^{64\times LC}$ and center its columns. Averaging the energy-weighted effective rank over $B = 50$ samples gives

$$\bar{r}_L = \frac{1}{B}\sum_{b=1}^{B} r_{\text{eff}}(W_{L,b}), \qquad R = \bar{r}_{96}, \qquad \Gamma = \frac{\bar{r}_{192}}{\bar{r}_{48}}. \tag{30}$$

The rank averages are computed before both the ratio defining Γ and the subsequent logarithm used for encoder selection. The default profile averages samples drawn with seeds 2021–2070. Training segments contain 8,640 observations for each hourly ETT dataset, 34,560 for each minute-level ETT dataset, 5,311 for Exchange, and 36,887 for Weather.

The Gram formulation fixes the size of each spectral decomposition. We compute effective rank from the eigenvalues of the 64×64 Gram matrix of the centered windows, using squared-singular-value weights. Eigenvalues no greater than 10^{-12} times the largest eigenvalue are omitted. The eigendecomposition size remains fixed as the number of variables or window length increases. Column centering bounds the effective rank by 63, giving all three scales the same maximum.

F.4 Log-Rank Calibration

Source labels associate growth and rank with structural choices. The DRS-RG selections in Table 33 provide the empirical labels. We associate the higher-growth branch with Variate+MLPMixer and the remaining branch with Patch+CI. Within Patch+CI, the encoder order is $(c_0, c_1, c_2, c_3) = (\text{None}, \text{GRU}, \text{Transformer}, \text{Mamba})$. Let $\mathcal{P}$ and $\mathcal{V}$ contain source datasets assigned to the Patch and Variate branches, respectively, and let $\mathcal{S}_j \subseteq \mathcal{P}$ contain those assigned to c_j. For separated source growth ranges, the branch threshold and observed encoder anchors are

$$\tau_\Gamma = \frac{\max_{d\in\mathcal{P}} \Gamma_d + \min_{d\in\mathcal{V}} \Gamma_d}{2}, \qquad \mu_j = \frac{1}{|\mathcal{S}_j|}\sum_{d\in\mathcal{S}_j} \log R_d. \tag{31}$$

Interpolation supplies anchors for unobserved encoder classes. Let a and b index the nearest observed anchors bracketing a missing class j in the fixed encoder order. For an unobserved endpoint class, use the two observed anchors nearest that endpoint for extrapolation. The completed anchor is

$$\widehat{\mu}_j = \mu_a + \frac{j-a}{b-a}(\mu_b - \mu_a), \tag{32}$$

while observed classes retain $\widehat{\mu}_j = \mu_j$. The boundaries $t_j = (\widehat{\mu}_j + \widehat{\mu}_{j+1})/2$ are midpoints in log-rank coordinates, corresponding to geometric means in the original rank coordinates. The resulting intervals compare relative differences in complexity.

Calibration estimates numerical boundaries within a fixed rule family. It requires both embedding branches, at least two observed Patch encoder classes, separated branch growth ranges, and increasing encoder anchors. Four Patch classes each have one reference dataset, so the fixed class order provides the missing-class position when that dataset is held out. The feature definitions, branch assignments, and encoder order were established from these six datasets. The leave-one analysis re-estimates the numerical boundaries while retaining those definitions and assignments.

The fitted thresholds determine five possible outputs. Calibrating on all six source profiles gives $\tau_\Gamma = 1.4269315475$ and rank-space boundaries $(e^{t_0}, e^{t_1}, e^{t_2}) = (5.0370993779, 7.3106598310, 10.7703356364)$. Growth above τ_Γ selects Variate+MLPMixer. Otherwise, increasing R selects Patch+CI with None, GRU, Transformer, and Mamba in sequence. The leave-one evaluation uses separately estimated boundaries in each fold. The associated forecasting configurations use input and prediction lengths of 96, representation width 128, and Patch length/stride of $16/8$.

F.5 Leave-One-Dataset-Out Evaluation

The evaluation measures agreement with the DRS-RG reference structures. We evaluate the rule on ETTh1, ETTh2, ETTm1, ETTm2, Exchange, and Weather in the 96-step input and 96-step prediction setting. Each leave-one-dataset-out fold estimates its thresholds from the other five datasets, keeping the feature definitions, branch assignments, and encoder order fixed. Table 36 shows that the log-rank rule recovers all six reference choices at the default profiles.

Table 36: Data-based module selection with one dataset held out. All six default predictions match the DRS-RG reference structures. Repeat matches count agreement over 100 new batches, each recomputing the source and target profiles from 50 window samples per scale.

Dataset	R	Γ	Selected structure	Repeat matches
ETTh1	13.8626	1.3330	Patch+CI / Mamba	100/100
ETTh2	8.3678	1.3864	Patch+CI / Transformer	99/100
ETTm1	10.2533	1.4947	Variate / MLPMixer	100/100
ETTm2	6.3870	1.2861	Patch+CI / GRU	100/100
Exchange	3.9725	1.2157	Patch+CI / None	100/100
Weather	10.0045	1.4675	Variate / MLPMixer	100/100

Cross-scale growth distinguishes choices with similar baseline complexity. ETTh1 has the largest baseline rank and selects Patch+CI with Mamba. ETTm1 and Weather select Variate+MLPMixer through their larger cross-scale growth. The two statistics therefore distinguish structural preferences that baseline rank alone would place in a similar complexity range.

F.6 Sampling Stability

Repeated sampling preserves the reference choices in 599 of 600 predictions. The evaluation comprises 100 new batches using seeds 50000–54999, with 50 samples per batch and scale. In the source-and-target setting, each batch recalibrates thresholds from five source profiles before predicting the sixth dataset. In the target-only setting, each fold's default five-source thresholds remain fixed while its target profile is resampled. Both settings produce $599/600$ matches. Table 36 reports the per-dataset repeat matches for the source-and-target setting.

The residual sampling sensitivity occurs at the embedding branch. In the differing ETTh2 batch, $\Gamma = 1.402542$ exceeds the respective branch thresholds 1.400967 and 1.400218, while $R = 8.303772$ remains in the Transformer range. Within the Patch branch, resampled ranks preserve the reference encoder intervals. These results show that the multiscale profile consistently reproduces the source structural preferences under repeated window sampling on the six reference datasets.