RECONSTRUCTION ADVANCE PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

The core objective of diverse time-series prediction methods is to learn a function that maps historical inputs to future outputs. However, the overlap between the input and output segments is often overlooked, which can be deemed the reconstruction of an overlap window. This shared portion naturally carries information about the underlying dynamics but is not explicitly exploited by many methods. Here, we propose **Re**construction **A**dvance **P**rediction (**ReAP**) as a plug-in training paradigm to explicitly leverage this overlap information. Theoretically, we show that reconstructing the overlap window can enhance prediction performance from the perspective of Dynamic Mode Decomposition (DMD). Building on this insight, we introduce a consistent loss to ensure that the learned function aligns with the real dynamics underlying the temporal system. We first demonstrate ReAP's effectiveness with a single linear layer, then generalize the paradigm to several state-of-the-art (SOTA) models. Extensive experiments show that ReAP consistently improves prediction performance across various deep learning backbones on 7 real-world datasets. An ablation study further confirms that a suitable ReAP configuration enables the learned function to follow the scaling law of Time Series Forecasting (TSF, Shi et al. (2024)). Code is available at this repository: https://anonymous.4open.science/r/ReAP-BB71/.

1 INTRODUCTION

Time series prediction is a classical field with applications ranging from climate forecasting to the stock market (Box et al., 2015). Beyond these traditional areas, conceptualizing a system's evolution over time allows a vast body of modern research to fit into this prediction paradigm (Montgomery et al., 2015; Chatfield & Xing, 2019). Specifically, generating token sequences in large language models, denoising Gaussian noise in diffusion models, and modeling user behavior in recommendation systems can be framed as high-dimensional, nonstationary time series prediction (Dathathri et al., 2019; Salimans & Ho, 2022; Kang & McAuley, 2018).

Time series forecasting hinges on leveraging temporal dependencies within history input to predict future trends (Hyndman & Athanasopoulos, 2018). While methods ranging from classical ARIMA (Hamilton, 2020) to modern architectures like Mamba (Gu & Dao, 2024) obey this foundational principle, they differ primarily in how they model these dependencies. The main challenge is representing temporal dependencies that span short-term fluctuations, long-term trends, periodic cycles, and hierarchical structures. In long-term time series forecasting (LTSF), existing approaches typically extract these patterns by mapping past observations directly to future targets (Zhou et al., 2020; Wu et al., 2021; Nie et al., 2022; Zeng et al., 2023). However, despite heavy focus on input and output windows, the role of the overlap between them is frequently overlooked. The overlap window inherently dictates the target task: a zero-overlap window defines pure prediction, whereas a full-overlap window corresponds to sequence imputation (Tashiro et al., 2021). Although some studies incorporate overlapping context to assist forecasting, the explicit value of reconstructing this overlap remains underappreciated.

To address this gap, we propose **Re**construction **A**dvance **P**rediction (**ReAP**), a learning paradigm that explicitly mines the underlying temporal dynamics within the overlap window. Theoretically, we leverage Dynamic Mode Decomposition (DMD) to prove that reconstructing the overlap window serves as a mathematical bridge between history and future windows (Kutz et al., 2016). Building on this insight, we deduce a consistent loss to ensure that the model captures real dynamics lying in the temporal system. As shown in Figure 1, ReAP feeds the history window to LTSF models in a

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.

general form, but modifies the output to include both the overlap and future windows. Additionally, ReAP combines the loss between outputs and observations with the consistent loss, leading to a dual-objective function for training the model. We validate ReAP's effectiveness by first training a single linear layer (ReLinear) to achieve competitive performance against SOTA baselines. Subsequently, we generalize ReAP to these baselines, demonstrating that diverse model backbones trained within ReAP deliver consistent accuracy gains across 7 real-world datasets. Next, we show that ReLinear follows the scaling law of TSF (Shi et al., 2024), while the accuracy of baseline methods declines as the history input increases. We further perform an ablation study to investigate this alignment with TSF's scaling law.

Our contributions are summarized as follows:

- **Theoretical Foundation:** We propose **Re**construction **A**dvance **P**rediction (**ReAP**), leveraging DMD to theoretically prove that reconstructing the overlap window bridges history and future windows. Guided by this insight, we derive a consistent loss that enforces models to capture underlying temporal dynamics.
- **Generalized Enhancement:** We assess ReAP's effectiveness using a single linear layer (ReLinear). Experiments across 7 real-world datasets show that ReLinear matches state-of-the-art (SOTA) baselines. We further apply ReAP as a plug-in training paradigm to these baseline models based on diverse backbones, achieving generalized accuracy improvements.
- **Scaling law Alignment:** Contrary to baselines, ReLinear's prediction accuracy improves as the history input length increases, aligning with the scaling law of TSF. We further conduct an ablation study to investigate this alignment.

2 Related Work

Long-Term Time Series Forecasting (LTSF). In recent years, LTSF has evolved beyond classical statistical approaches like ARIMA (Box et al., 2015). Driven by the need to capture temporal dependencies, initial deep learning methods focused on transformer architectures, such as Informer's ProbSparse attention, Autoformer's series decomposition and autocorrelation, and PatchTST's patch-wise tokenization (Zhou et al., 2020; Wu et al., 2021; Nie et al., 2022). While the necessity of complex attention mechanisms was soon challenged by simple linear baselines that match transformer-based models on standard LTSF benchmarks (Zeng et al., 2023). Consequently, research shifted toward redesigning how temporal and cross-variable dependencies are represented: iTransformer applies attention across inverted variates (Liu et al., 2024), TimeXer integrates exogenous information via global endogenous tokens (Wang et al., 2024b), and TimeFilter adaptively filters noise using patch-specific spatiotemporal graphs (Hu et al., 2025). Parallel efforts address multiscale temporal dynamics; for instance, TimeMixer decomposes time series across sampling resolutions and aggregates past and future multi-scale representations (Wang et al., 2024a), echoing broader frequency and decomposition-based modeling strategies (Xu et al., 2024). Additionally, self-supervised methods leverage masked reconstruction or contrastive learning (Eldele et al., 2021; Yue et al., 2022; Dong et al., 2023), though reconstruction is typically restricted to pretraining or random masking rather than forecasting targets. Despite these architectural advances across variate inversion, graph filtration, and multiscale mixing, existing LTSF models still predom-

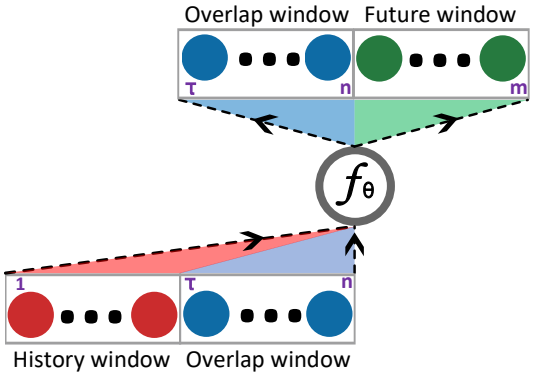


Figure 1: Illustration of ReAP, the history window includes $[1, \cdots, \tau, \cdots, n]$, the future includes spans $[\tau, \cdots, n, \cdots, m]$, and the overlap is the intersection of these two windows. f_θ denotes deep learning models with learnable parameters θ.

inantly formulate forecasting as a direct mapping from historical observations to a disjoint future horizon, often neglecting the dynamic information underlying them.

Dynamic Mode Decomposition (DMD). DMD is a data-driven, equation-free technique originally developed in fluid dynamic systems to extract spatiotemporal coherent structures and approximate nonlinear dynamics via fitting linear transition operators on time-shifted snapshots (Schmid, 2010; Kutz et al., 2016). Its eigendecomposition separates complex dynamics into spatial modes, growth rates, and frequencies, providing interpretable insights into the temporal system (Rowley et al., 2009). DMD is fundamentally grounded in Koopman operator theory, which represents nonlinear dynamical systems through infinite-dimensional linear operators acting on measurement observables (Koopman, 1931; Mezić, 2005). To capture richer non-linear dynamics, variants like Extended DMD, Kernel DMD, and Hankel/delay-coordinate DMD expand the observable space or leverage lagged measurement matrices to determine latent invariant subspaces (Li et al., 2017; Williams et al., 2014; Brunton et al., 2017). Recent work uses autoencoders to learn basis function spaces, mapping inputs into nonlinear latent representations governed by linear evolution operators (Takeishi et al., 2017; Lusch et al., 2018; Azencot et al., 2020). The commute between a nonlinear operator in the observation space and a linear operator in the latent space is used in time series prediction (Liu et al., 2023; Zheng et al., 2025). Crucially, classical DMD and modern Koopman theory both require state coordinates under shared dynamics (Ljung, 2010; Brunton et al., 2021). This dynamical consistency applies to predictions based on the overlap window, which bridges past and future time series to learn the state-transition operator.

3 Effect of Time Lag on DMD Estimation

Given a multivariate time series $\{x_k\}_{k\in\mathbb{Z}} \subset \mathbb{R}^d$, the number of paired snapshots denoted by n , and the time lag denoted by $\tau \geq 1$. We define

$$X_n = [x_1, \ldots, x_n] \in \mathbb{R}^{d\times n}, \qquad X_{\tau,n} = [x_{1+\tau}, \ldots, x_{n+\tau}] \in \mathbb{R}^{d\times n}. \tag{1}$$

DMD seeks a linear transition operator $A_\tau \in \mathbb{R}^{d\times d}$ such that

$$X_{\tau,n} \approx A_\tau X_n. \tag{2}$$

This is formulated as the least-squares problem

$$\widehat{A}_{\tau,n} \in \underset{A\in\mathbb{R}^{d\times d}}{\operatorname{argmin}} \|X_{\tau,n} - AX_n\|_F^2 \implies \widehat{A}_{\tau,n} = X_{\tau,n}X_n^\dagger \in \mathbb{R}^{d\times d}. \tag{3}$$

Estimating DMD's operator critically depends on τ (Schmid, 2010; Tu, 2013).

Proposition 1 (Consistency of the lag-τ DMD estimator). *Let $\{x_k\}_{k\in\mathbb{Z}}$ be a strictly stationary and ergodic process satisfying*

$$\mathbb{E}[x_k] = 0, \qquad \mathbb{E}\|x_k\|_2^2 < \infty. \tag{4}$$

Assume that the contemporaneous covariance matrix

$$C_0 = \mathbb{E}[x_k x_k^\top] \tag{5}$$

is positive definite. For every fixed $\tau \geq 1$, define the lag-τ cross-covariance matrix as

$$C_\tau = \mathbb{E}[x_{k+\tau} x_k^\top]. \tag{6}$$

Then

$$\widehat{A}_{\tau,n} = X_{\tau,n}X_n^\dagger \xrightarrow[n\to\infty]{\text{a.s.}} A_\tau := C_\tau C_0^{-1}. \tag{7}$$

The proof is detailed in Appendix A.1. This proposition shows that the lag τ determines the temporal scale in the learned DMD operator. Under stationarity and ergodicity, the empirical covariance and lagged cross-covariance matrices converge almost surely to their population counterparts (Birkhoff, 1931; Walters, 2000). Consequently, the DMD estimator converges to the population operator

$$A_\tau = C_\tau C_0^{-1}, \tag{8}$$

which is the coefficient matrix of the best linear predictor of $X_{\tau,n}$ from X_n.

Remark. Proposition 1 establishes consistency for every fixed τ, but it does not imply that all choices of τ yield the same finite-sample prediction performance. Selecting τ therefore requires a practical selection tailored to the specific application. Specifically, for large τ, temporal dependence may weaken, causing C_τ and thus A_τ to approach zero in weakly dependent processes. By contrast, if $\tau = 0$, then

$$C_\tau = C_0 \qquad \text{and} \qquad A_0 = C_0 C_0^{-1} = I_d, \tag{9}$$

yielding the trivial identity mapping rather than a prediction operator.

4 TIME LAG τ BRIDGES HISTORY AND FUTURE DATA

We now examine how the time lag τ affects DMD estimation by relating the history snapshot matrix X to the time-shifted future matrix X_τ. Using Singular Value Decomposition (SVD), the resulting reduced-order DMD operator can be factored into components representing spatial alignment, response amplitude, temporal alignment, and predictor scaling (Kutz et al., 2016; Brunton et al., 2021).

Lemma 1 (SVD factorization of the time-shifted DMD operator). *Let*

$$X_k = U_k \Sigma_k V_k^\top \tag{10}$$

be a rank-k truncated SVD approximation of the history snapshot matrix X, and let

$$X_{\tau,k} = U_{\tau,k} \Sigma_{\tau,k} V_{\tau,k}^\top \tag{11}$$

be a rank-k truncated SVD approximation of the time-shifted snapshot matrix X_τ. Here,

$$U_k, U_{\tau,k} \in \mathbb{R}^{d\times k}, \qquad \Sigma_k, \Sigma_{\tau,k} \in \mathbb{R}^{k\times k}, \qquad V_k, V_{\tau,k} \in \mathbb{R}^{n\times k}, \tag{12}$$

the columns of $U_k, U_{\tau,k}, V_k$, and $V_{\tau,k}$ are orthonormal, and Σ_k is nonsingular. Define the truncated pseudoinverse of X_k by

$$X_k^\dagger = V_k \Sigma_k^{-1} U_k^\top. \tag{13}$$

Then the reduced-order DMD operator based on the rank-k approximations,

$$\widehat{A}_{\tau,k} = U_k^\top \left(X_{\tau,k} X_k^\dagger \right) U_k, \tag{14}$$

admits the exact factorization

$$\widehat{A}_{\tau,k} = \left(U_k^\top U_{\tau,k} \right) \Sigma_{\tau,k} \left(V_{\tau,k}^\top V_k \right) \Sigma_k^{-1}. \tag{15}$$

Remark. The factorization separates the reduced-order DMD operator into 4 interpretable components:

$$\widehat{A}_{\tau,k} = \underbrace{\left(U_k^\top U_{\tau,k} \right)}_{\text{spatial alignment}} \underbrace{\Sigma_{\tau,k}}_{\text{response amplitude}} \underbrace{\left(V_{\tau,k}^\top V_k \right)}_{\text{temporal alignment}} \underbrace{\Sigma_k^{-1}}_{\text{predictor scaling}}. \tag{16}$$

1. **Spatial alignment.** The product $U_k^\top U_{\tau,k}$ captures the spatial structure by quantifying the relative orientation between the dominant left singular subspaces derived from the history and time-shifted snapshot matrices. Since $U_{\tau,k}$ is obtained from X_τ, the principal angles between the predictor subspace $\mathrm{span}(U_k)$ and the response subspace $\mathrm{span}(U_{\tau,k})$ vary with τ.The singular values of $U_k^\top U_{\tau,k}$ quantify the spatial alignment between historical features and the spatially shifted dynamics after time lag τ.
2. **Response amplitude.** The diagonal matrix $\Sigma_{\tau,k}$ contains the retained singular values associated with the time-shifted snapshot matrix X_τ. These values represent the strength (energy/magnitude) of the dominant response directions after the time lag τ. Therefore, the τ-dependence of $\Sigma_{\tau,k}$ reflects how much of the time-shifted data lies in the rank-k subspace spanned by $U_{\tau,k}$.
3. **Temporal alignment.** The product $V_{\tau,k}^\top V_k$ captures the temporal structure by quantifying the relative orientation between the dominant right singular subspaces, and the singular values quantify how well the history matches the time-shifted snapshot. Since $V_{\tau,k}$ depends on X_τ, changing τ alters snapshot-to-snapshot variation in the retained subspace.

4. **Predictor scaling.** The factor Σ_k^{-1} depends only on the rank-k SVD of the history snapshot matrix X and rescales the predictor directions according to their singular values. The scaling's influence on the estimated operator depends on τ via the product $\left(U_k^\top U_{\tau,k}\right)\Sigma_{\tau,k}\left(V_{\tau,k}^\top V_k\right)$, as τ alters how response components map to predictor coordinates.

The proof and full remark are detailed in Appendix A.2. The time lag τ determines how the response data are formed through the time-shifted matrix X_τ, and therefore alters the corresponding SVD factors $U_{\tau,k}$, $\Sigma_{\tau,k}$, and $V_{\tau,k}$. As a result, τ changes how the response subspaces align with the predictor subspaces U_k and V_k, and consequently affects the estimated DMD operator.

4.1 Consistent loss and its surrogate

The DMD estimator learns a linear operator that maps a history snapshot matrix to its time-shifted counterpart. When the real system is governed by nonlinear and complex dynamics, this linear map can only serve as an approximation of the ideal regression operator. We quantify the resulting approximate error by a consistency loss

$$\mathcal{L}_{\text{consistent}} := \frac{1}{d}\left\|A_\tau - \widehat{A}_\tau\right\|_F^2 = \frac{1}{d}\left\|(X_\tau - \widehat{X}_\tau)X^\dagger\right\|_F^2, \tag{17}$$

where d denotes the data dimension, † denotes the Moore–Penrose pseudoinverse, and $\|\cdot\|_F$ denotes the Frobenius norm. In the Koopman theory, $\mathcal{L}_{\text{consistent}}$ can vanish in infinite function space (Brunton et al., 2021), but for finite data and truncation, it is nonzero and depends on how $X^\dagger$ weights different singular directions of the snapshot matrix. In particular, since DMD's operator takes the form $X_\tau X^\dagger$, the stability and expressivity of the learned dynamics are determined by the singular spectrum of $X^\dagger$. When $n > d$, the effective rank is limited by d, and truncation suppresses directions associated with small singular values. Conversely, when $d > n$, the regressor has rank at most n, so inverse-like weighting necessarily involves directions corresponding to vanishing singular values. Moreover, truncating the learned operator to rank k restricts the model to a finite-dimensional subspace, which generally discards components associated with high-frequency modes. Consequently, reducing $\mathcal{L}_{\text{consistent}}$ encourages $\widehat{A}_\tau$ to align with the optimal finite-rank approximation of the population linear regression operator.

In practice, explicitly computing $X^\dagger$ within a deep learning training loop can be costly and numerically unstable, especially when X is ill-conditioned or its effective rank changes during optimization. To avoid inverse operations, we replace the $\mathcal{L}_{\text{consistent}}$ with a transpose surrogate

$$\widehat{\mathcal{L}}_{\text{sur}} := \frac{1}{n}\left\|\Delta X_\tau^\top X\right\|_F^2, \qquad \Delta X_\tau := X_\tau - \widehat{X}_\tau. \tag{18}$$

Proposition 2 (Inverse-to-transpose replacement). *Let $\{x_n\}_{n\in\mathbb{Z}}$ be a strictly stationary and ergodic process and let*

$$C_\tau = \mathbb{E}[x_{n+\tau}x_n^\top], \qquad C_0 = \mathbb{E}[x_n x_n^\top] \succ 0. \tag{19}$$

Fix a truncation rank $k \le d$. Assume the data matrix $X \in \mathbb{R}^{d\times n}$ admits a truncated SVD

$$X = U\Sigma V^\top, \qquad \Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_r), \quad \sigma_i > 0, \tag{20}$$

and define the rank-k truncation

$$X_k := U_k\Sigma_k V_k^\top, \qquad X_k^\dagger := V_k\Sigma_k^{-1}U_k^\top, \tag{21}$$

where $\Sigma_k = \operatorname{diag}(\sigma_1, \ldots, \sigma_k)$ and assume $\sigma_k > 0$ so that $\operatorname{rank}(X_k) = k$. Let $\widehat{X}_\tau \in \mathbb{R}^{d\times n}$ be any linear predictor and set $\Delta X_\tau := X_\tau - \widehat{X}_\tau$. Define the transpose-space inverse-weighted and surrogate losses

$$\widehat{\mathcal{L}}_{\text{consistent},k} := \frac{1}{n}\left\|\Delta X_\tau^\top (X_k^\top)^\dagger\right\|_F^2, \qquad \widehat{\mathcal{L}}_{\text{sur},k} := \frac{1}{n}\left\|\Delta X_\tau^\top X_k\right\|_F^2. \tag{22}$$

Assume that

$$\frac{1}{n}X_k X_k^\top \xrightarrow{\text{a.s.}} \Pi_k C_0 \Pi_k, \tag{23}$$

where Π_k is the projector onto $\mathrm{span}\{v_1, \ldots, v_k\}$*, and assume that $\Pi_k C_0 \Pi_k$ has strictly positive eigenvalues on its range. Let $\lambda_k^{\max}$ and $\lambda_k^{\min}$ denote the largest and smallest (nonzero) eigenvalues. Then, as $n \to \infty$,*

$$\frac{1}{n^2(\lambda_k^{\max})^2}\widehat{\mathcal{L}}_{\mathrm{sur},k} \;\le\; \widehat{\mathcal{L}}_{\mathrm{consistent},k} \;\le\; \frac{1}{n^2(\lambda_k^{\min})^2}\widehat{\mathcal{L}}_{\mathrm{sur},k} \quad \textit{a.s.} \tag{24}$$

The proof of Proposition 2 is detailed in Appendix A.3

5 Generalize DMD operator to Neural Network

Classical DMD estimates a linear operator from fixed pairs of time snapshots. This couples the learned operator to the particular snapshot pairing, which limits flexibility when predicting at different horizons. Moreover, applying this explicit calculation across many temporal segments is computationally expensive and may provide poor generalization under non-stationary dynamics. To address these limitations, we propose **Restruction Advance Prediction** (**ReAP**) by replacing the DMD operator $\widehat{A}$ with a learnable predictor parameterized by a neural network f_θ.

Let the data dimension be d, with history length $L_{\mathrm{history}} = n$ and future length $L_{\mathrm{future}} = m$. As shown in Figure 1, we define the history window $[1, \ldots, \tau, \ldots, n]$, the future window $[\tau, \ldots, n, \ldots, m]$, and the overlap window as the intersection of these two windows. Let

$$X_{\mathrm{input}} \in \mathbb{R}^{d \times L_{\mathrm{history}}}, \qquad X_{\mathrm{output}} \in \mathbb{R}^{d \times L_{\mathrm{future}}} \tag{25}$$

be the data matrices extracted from the historical and future windows, respectively. ReAP employs f_θ to learn the predictor

$$X_{\mathrm{output}} = f_\theta(X_{\mathrm{input}})\,, \tag{26}$$

where θ denotes the trainable parameters. To ensure f_θ effectively captures the temporal dynamics underlying the data, ReAP adopts a dual-objective training loss function

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{MSE}}(X_{\mathrm{output}}, X_{\mathrm{true}}) + \alpha\, \mathcal{L}_{\mathrm{consistent}}\Big(\widehat{X}_{\mathrm{output}}, X_{\mathrm{true}}, X_{\mathrm{input}}\Big)\,, \tag{27}$$

where $X_{\mathrm{true}} \in \mathbb{R}^{d \times L_{\mathrm{future}}}$ is the ground-truth future snapshots, and $\alpha > 0$ is hyperparameters. For $\mathcal{L}_{\mathrm{MSE}}$, ReAP calculates the Mean Square Error (MSE) between X_{output} and X_{true} within the overlap window. For $\mathcal{L}_{\mathrm{consistent}}$, ReAP implements it according to Equation 18 like

$$\mathcal{L}_{\mathrm{consistent}}\Big(\widehat{X}_{\mathrm{output}}, X_{\mathrm{true}}, X_{\mathrm{input}}\Big) = \frac{1}{d}\left\|\Big(X_{\mathrm{true}} - \widehat{X}_{\mathrm{output}}\Big)^\top X_{\mathrm{input}}\right\|_F^2. \tag{28}$$

By minimizing $\mathcal{L}_{\mathrm{total}}$ through backpropagation, ReAP trains network parameters to align with the temporal system's dynamics, integrating DMD-derived dynamical consistency into deep neural network learning.

Remark. L_{overlap} and α are critical hyperparameters in ReAP.

1. L_{overlap} controls the extent of shared temporal dynamics between history and future windows. As L_{overlap} increases, X_{history} and X_{future} share more common dynamics information. However, an earlier starting index τ increases the total length of the future window, leading to degraded performance limited due to limited model capacity. Conversely, a smaller L_{overlap} reduces the prediction burden but sacrifices shared dynamic context, weakening the identifiability of the underlying system.
2. α balances pointwise prediction accuracy ($\mathcal{L}_{\mathrm{MSE}}$) and dynamical consistency ($\mathcal{L}_{\mathrm{consistent}}$). A higher α emphasizes the DMD-like operator, potentially reducing pointwise accuracy and even hindering backbone convergence, leading to suboptimal performance. Conversely, a lower α prioritizes pointwise accuracy, potentially neglecting the dynamical constraint and focusing solely on history-to-future pattern learning.
3. Since different deep learning backbones can follow different optimization trajectories depending on the dataset, careful tuning of L_{overlap} and α is crucial to achieve robust and generalizable learning of f_θ.

To evaluate ReAP's effectiveness, we first instantiate f_θ as a single linear layer (ReLinear) to validate that DMD's dynamics theory holds in deep learning. Consequently, we use ReAP as a plug-in training paradigm to train diverse deep learning backbones, enabling non-linear estimation of f_θ.

6 EXPERIMENTS

The overlap window is used only during training; in testing, we discard it and evaluate only the prediction horizon to avoid data leakage.

6.1 DATASET.

To evaluate the effectiveness of ReAP, we conduct experiments on widely used multivariate forecasting benchmarks, including `ETT` (`ETTh1`, `ETTh2`, `ETTm1`, `ETTm2`), `Weather`, `Electricity`, and `Traffic`.. More details on these datasets can be found in Informer (Zhou et al., 2020) and Autoformer (Wu et al., 2021). These datasets are standard in long-term forecasting evaluation and cover diverse fields, including periodic industrial signals, electricity consumption, meteorological observation, and transportation measurements.

6.2 BASELINES.

In Section 6.4, we compare ReLinear's prediction performance with state-of-the-art (SOTA) baselines, including TimesNet (Wu et al., 2022), DLinear Zeng et al. (2023), iTransformer (Liu et al., 2024), TimeMixer (Wang et al., 2024a), TimeXer(Wang et al., 2024b), and TimeFilter (Hu et al., 2025). In Section 6.5, we integrate ReAP as a plug-in training paradigm into each baseline, and compare the prediction accuracy with and without ReAP. The introduction of each baseline is detailed in Appendix B.

6.3 IMPLEMENTATION DETAILS.

Relinear was trained for 50 epochs using the Adam optimizer (Kingma & Ba, 2014) with a learning rate of 1×10^{-2}, selecting the best validation result. Baselines were trained using their official settings. We used grid search to select the overlap window and weight α, with selections for each model and dataset reported in the Appendix C. We selected two common metrics in time series forecasting: Mean Absolute Error (MAE) and Mean Squared Error (MSE). Experiments were conducted on 1 NVIDIA RTX 2080 Ti 11 GB GPU.

6.4 PREDICT PERFORMANCE OF RELINEAR

We evaluate the forecasting performance of ReLinear against prominent LTSF baselines in Table 1. Building on a single linear-layer architecture, ReLinear outperforms DLinear across various history window lengths and datasets, reducing MSE by 8.7% and MAE by 5.9%. As the history window length increases, ReLinear gains more temporal dynamics information, resulting in consistent reductions in MSE and MAE across all datasets. In particular, in small-dimension datasets (e.g., `ETTs`), ReLinear consistently outperforms even sophisticated baselines. Notably, the accuracy gains stem not only from using a longer history window, but also from learning the temporal dynamics via the dual-object loss function. Table 5 shows that a large L_{overlap} is generally chosen, leading to a longer prediction horizon. Although this stresses ReLinear's pointwise mapping, it improves prediction performance by offering more dynamic information.

6.5 GENERAL IMPROVEMENT VIA REAP

To verify ReAP's generalization as a plug-in training paradigm, we compare 6 deep learning models trained with ReAP and official settings. As detailed in Table 2, models trained with the ReAP paradigm (denoted with the **+** suffix) consistently outperform their default counterparts across all benchmark datasets and prediction horizons. Specifically, ReAP lowers MSE by 3.15% and MAE by 2.24% on average across all models, datasets, and prediction lengths. By model, it performs best with TimesNet and ReLinear, improving the averaged results by 3.94% and 3.82%, respectively. By dataset, ReAP reduces MSE in `ETTm1` (dimension = 7) and `Traffic` (dimension = 862) by 3.09% and 2.22%, respectively. These empirical results confirm that learning the underlying temporal dynamics within the overlap window enhances deep learning for LSTF.

Table 1: Long-term forecasting results. Each row corresponds to an input length of [96, 192, 336, 720], and results are averaged over prediction lengths of [96, 192, 336, 720]. Lower MSE and MAE values signify better predictions. Red indicates the best result, Blue indicates the second best, and ✗ indicates a numerical error. Full results are in Table 9 ∼ 12.

Models		ReLinear		DLinear		TimeXer		iTransfomer		TimeFilter		TimesNet		TimeMixer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.452	0.448	0.496	0.481	0.455	0.444	0.450	0.443	0.450	0.437	0.491	0.480	0.449	0.439
	192	0.438	0.442	0.451	0.461	0.451	0.442	0.450	0.451	0.448	0.442	0.460	0.461	0.445	0.440
	336	0.424	0.438	0.455	0.461	0.456	0.461	0.445	0.452	0.456	0.454	0.486	0.481	0.449	0.448
	720	0.421	0.442	0.426	0.445	0.449	0.461	0.465	0.471	0.487	0.474	0.507	0.500	0.551	0.563
ETTh2	96	0.395	0.421	0.564	0.480	0.383	0.405	0.395	0.412	0.376	0.402	0.408	0.423	0.389	0.410
	192	0.384	0.424	0.485	0.474	0.370	0.405	0.381	0.410	0.385	0.414	0.425	0.437	0.380	0.410
	336	0.380	0.414	0.466	0.463	0.362	0.402	0.371	0.408	0.372	0.409	0.416	0.443	0.374	0.407
	720	0.368	0.415	0.508	0.484	0.368	0.409	0.385	0.419	0.388	0.418	0.430	0.452	✗	✗
ETTm1	96	0.402	0.408	0.404	0.408	0.394	0.405	0.420	0.416	0.382	0.397	0.405	0.414	0.387	0.401
	192	0.368	0.386	0.368	0.386	0.368	0.389	0.366	0.388	0.365	0.390	0.385	0.407	0.370	0.394
	336	0.359	0.382	0.363	0.385	0.367	0.393	0.362	0.389	0.361	0.390	0.395	0.413	0.369	0.392
	720	0.359	0.383	0.361	0.384	0.367	0.397	0.365	0.395	0.370	0.399	0.400	0.420	0.365	0.389
ETTm2	96	0.287	0.336	0.357	0.402	0.284	0.327	0.295	0.336	0.274	0.322	0.294	0.332	0.275	0.322
	192	0.269	0.325	0.315	0.373	0.266	0.319	0.277	0.330	0.266	0.319	0.290	0.337	0.268	0.321
	336	0.255	0.322	0.287	0.349	0.262	0.349	0.271	0.328	0.267	0.325	0.287	0.337	0.270	0.326
	720	0.252	0.318	0.268	0.332	0.283	0.334	0.272	0.334	0.306	0.347	0.291	0.352	0.273	0.332
Weather	96	0.264	0.316	0.267	0.318	0.244	0.274	0.259	0.280	0.250	0.276	0.259	0.287	0.244	0.275
	192	0.253	0.306	0.254	0.306	0.234	0.268	0.248	0.276	0.234	0.270	0.261	0.290	0.231	0.265
	336	0.243	0.295	0.246	0.299	0.227	0.266	0.237	0.273	0.230	0.271	0.248	0.287	0.226	0.263
	720	0.334	0.277	0.240	0.293	0.231	0.274	0.240	0.279	0.234	0.277	0.252	0.291	0.230	0.267
ECL	96	0.209	0.291	0.240	0.324	0.170	0.269	0.178	0.269	0.166	0.261	0.208	0.305	0.185	0.274
	192	0.177	0.272	0.201	0.306	0.169	0.265	0.169	0.263	0.173	0.266	0.208	0.306	0.167	0.257
	336	0.167	0.264	0.182	0.287	0.174	0.271	0.167	0.263	0.162	0.255	0.195	0.295	0.164	0.255
	720	0.160	0.258	0.171	0.267	0.171	0.271	0.170	0.268	0.164	0.260	0.202	0.293	✗	✗
Traffic	96	0.624	0.384	0.680	0.422	0.480	0.288	0.428	0.285	0.427	0.281	0.634	0.339	0.515	0.307
	192	0.471	0.314	0.475	0.319	0.432	0.287	0.401	0.276	0.395	0.268	0.620	0.330	0.434	0.295
	336	0.434	0.296	0.436	0.300	0.407	0.279	0.397	0.278	0.387	0.267	0.620	0.337	✗	✗
	720	0.411	0.285	0.413	0.287	0.396	0.280	0.386	0.275	0.378	0.264	0.619	0.332	✗	✗

Table 2: Comparison of several deep learning models trained with ReAP and the official settings. The input length is 96, and results are averaged over prediction lengths of [96, 192, 336, 720]. The **+** suffix denotes models trained with ReAP, **Improve** denotes relative error reduction, and ✗ denotes no improvement. Full results are in Table 13

Models	ETTh1		ETTh2		ETTm1		ETTm2		Weather		ECL		Traffic	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ReLinear	0.457	0.453	0.452	0.443	0.409	0.418	0.305	0.361	0.269	0.323	0.212	0.299	0.653	0.401
ReLinear+	0.452	0.448	0.395	0.421	0.402	0.408	0.287	0.336	0.264	0.316	0.209	0.291	0.624	0.384
Improve	**1.09%**	**1.10%**	**12.59%**	**5.00%**	**1.71%**	**2.39%**	**5.90%**	**6.93%**	**1.86%**	**2.17%**	**1.42%**	**2.67%**	**4.47%**	**4.24%**
TimeMixer	0.454	0.444	0.389	0.410	0.387	0.401	0.275	0.322	0.244	0.275	0.185	0.274	0.515	0.307
TimeMixer+	0.439	0.431	0.375	0.401	0.377	0.393	0.273	0.320	0.242	0.271	0.186	0.275	0.496	0.314
Improve	**3.30%**	**2.92%**	**3.60%**	**2.20%**	**2.59%**	**2.00%**	**0.73%**	**0.62%**	**0.82%**	**1.45%**	✗	✗	**3.68%**	✗
TimeXer	0.455	0.444	0.383	0.405	0.394	0.405	0.284	0.327	0.244	0.274	0.170	0.269	0.480	0.288
TimeXer+	0.449	0.441	0.372	0.398	0.380	0.398	0.274	0.322	0.242	0.273	0.170	0.270	0.468	0.341
Improve	**1.32%**	**0.68%**	**2.89%**	**1.73%**	**3.55%**	**1.73%**	**3.52%**	**1.53%**	**0.82%**	**0.36%**	✗	✗	**2.50%**	✗
iTransformer	0.450	0.443	0.395	0.412	0.420	0.416	0.295	0.336	0.259	0.280	0.178	0.269	0.428	0.285
iTransformer+	0.443	0.441	0.377	0.405	0.389	0.398	0.280	0.326	0.259	0.279	0.178	0.267	0.425	0.280
Improve	**1.56%**	**0.45%**	**4.56%**	**1.70%**	**7.38%**	**4.33%**	**5.08%**	**2.98%**	✗	**0.36%**	✗	**0.75%**	**0.70%**	**1.75%**
TimeFilter	0.450	0.436	0.376	0.402	0.382	0.396	0.274	0.322	0.250	0.276	0.166	0.261	0.427	0.281
TimeFilter+	0.442	0.431	0.370	0.397	0.377	0.394	0.272	0.320	0.243	0.271	0.163	0.256	0.424	0.276
Improve	**1.78%**	**1.15%**	**1.60%**	**1.24%**	**1.31%**	**0.51%**	**0.73%**	**0.62%**	**2.80%**	**1.81%**	**1.81%**	**1.91%**	**0.70%**	**1.78%**
TimesNet	0.491	0.480	0.408	0.423	0.405	0.414	0.294	0.332	0.259	0.287	0.208	0.305	0.634	0.339
TimesNet+	0.448	0.442	0.384	0.407	0.397	0.406	0.287	0.327	0.253	0.280	0.190	0.291	0.626	0.333
Improve	**8.77%**	**7.92%**	**5.88%**	**3.78%**	**1.98%**	**1.93%**	**2.38%**	**1.51%**	**2.32%**	**2.44%**	**8.65%**	**4.59%**	**1.26%**	**1.77%**

6.6 Reconstruction Aligns with Scaling Law of TSF

In TSF, extending the history window length L_{history} often leads to diminishing model performance, contrary to the ideal scaling law (Shi et al., 2024). This phenomenon may arise because models prioritize learning pointwise relationships between history and future, often neglecting the temporal dynamics underlying time series. ReAP's dual-object loss function explicitly promotes the learning of temporal dynamics via $\mathcal{L}_{\text{consistent}}$, aligning models with the TSF scaling law. As shown in Figure 2, SOTA baselines deteriorate as historical input length increases from 96 to 720, whereas ReLinear+ (red solid curve) improves monotonically. To highlight the vital role of $\mathcal{L}_{\text{consistent}}$, we

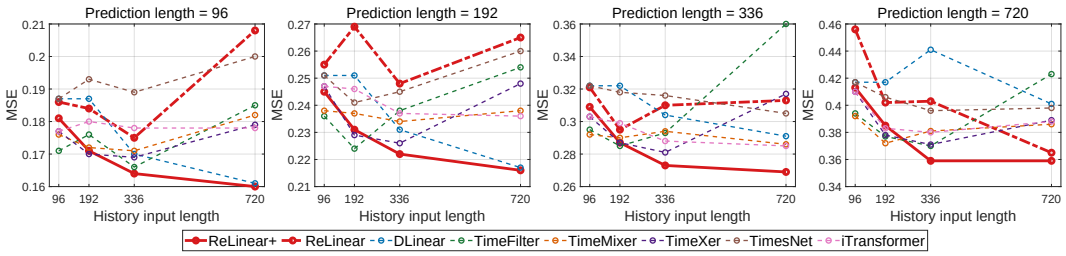


Figure 2: MSE versus history input length in `ETTm2`. Each subplot denotes a prediction length in [96, 192, 336, 720], with input length on the x-axis and MSE on the y-axis. Full results are detailed in Appendix E.

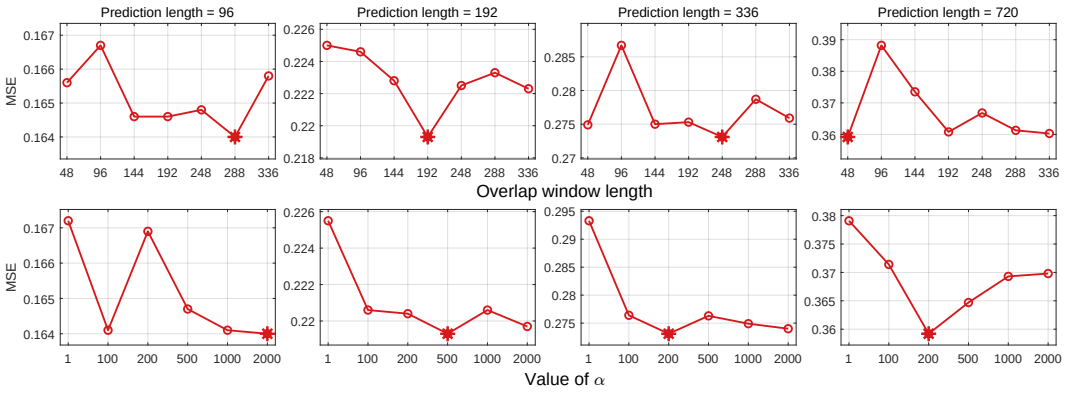


Figure 3: Ablation hyperparameters L_{overlap} and α in `ETTm2`, with history input length of 336. The top panel plots MSE versus L_{overlap}, and the bottom panel plots MSE versus α, with the hyperparameter value on the x-axis and MSE on the y-axis. The * marks the best results.

trained ReLinear without it (red dashed curve, using the same L_{overlap}). This resulted in significantly inferior performance compared to ReLinear+, especially for shorter prediction lengths (96, 192).

We further conduct an ablation study on key hyperparameters L_{overlap} and α in the dual-object loss function in Equation 27. Figure 3 shows that L_{overlap} and α significantly enhance prediction results, particularly for longer prediction lengths (336, 720). Specifically, for a prediction length of 720, the optimal selections of L_{overlap} and α reduce the MSE by 7.47% and 5.25%. As Remark 5 details, selecting both L_{overlap} and α is essential for ReAP to capture the temporal dynamics bridging the history and future time series.

7 Conclusion

In this work, we introduced Reconstruction Advance Prediction (**ReAP**), a plug-in training paradigm designed to enhance long-term time-series forecasting by integrating DMD's dynamics theory into deep learning. Building on DMD analysis of time series, we demonstrate how reconstructing the overlap window reveals underlying temporal system dynamics. To integrate these dynamics into neural network training, we derive a consistent loss function that avoids inverse operations. Experimental results indicate that ReLinear matches the performance of advanced baselines, and that ReAP consistently improves these baselines. We further show that ReLinear exhibits monotonic error reduction as history input increases, aligning with the scaling law of TSF. Finally, we conduct an ablation study on ReAP's key hyperparameters to explore its effectiveness in capturing temporal dynamics. For these hyperparameters' selection, several promising directions remain for future work. The reconstruction of the overlap window can be viewed as a state-space representation, and the weight α is well suited for adaptive weighting mechanisms.

REFERENCES

Omri Azencot, N Benjamin Erichson, Vanessa Lin, and Michael Mahoney. Forecasting sequential data using consistent koopman autoencoders. In *International conference on machine learning*, pp. 475–485. PMLR, 2020.

George D Birkhoff. Proof of the ergodic theorem. *Proceedings of the National Academy of Sciences*, 17(12):656–660, 1931.

George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

Steven L Brunton, Bingni W Brunton, Joshua L Proctor, Eurika Kaiser, and J Nathan Kutz. Chaos as an intermittently forced linear system. *Nature communications*, 8(1):19, 2017.

Steven L Brunton, Marko Budišić, Eurika Kaiser, and J Nathan Kutz. Modern koopman theory for dynamical systems. *arXiv preprint arXiv:2102.12086*, 2021.

Chris Chatfield and Haipeng Xing. *The analysis of time series: an introduction with R*. Chapman and hall/CRC, 2019.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. Plug and play language models: A simple approach to controlled text generation. *arXiv preprint arXiv:1912.02164*, 2019.

Jiaxiang Dong, Haixu Wu, Haoran Zhang, Li Zhang, Jianmin Wang, and Mingsheng Long. Simmtm: A simple pre-training framework for masked time-series modeling. *Advances in Neural Information Processing Systems*, 36:29996–30025, 2023.

Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, Chee Keong Kwoh, Xiaoli Li, and Cuntai Guan. Time-series representation learning via temporal and contextual contrasting. *arXiv preprint arXiv:2106.14112*, 2021.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First conference on language modeling*, 2024.

James D Hamilton. *Time series analysis*. Princeton university press, 2020.

Yifan Hu, Guibin Zhang, Peiyuan Liu, Disen Lan, Naiqi Li, Dawei Cheng, Tao Dai, Shu-Tao Xia, and Shirui Pan. Timefilter: Patch-specific spatial-temporal graph filtration for time series forecasting. *arXiv preprint arXiv:2501.13041*, 2025.

Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pp. 197–206. IEEE, 2018.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Bernard O Koopman. Hamiltonian systems and transformation in hilbert space. *Proceedings of the National Academy of Sciences*, 17(5):315–318, 1931.

J Nathan Kutz, Steven L Brunton, Bingni W Brunton, and Joshua L Proctor. *Dynamic mode decomposition: data-driven modeling of complex systems*. SIAM, 2016.

Qianxiao Li, Felix Dietrich, Erik M Bollt, and Ioannis G Kevrekidis. Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the koopman operator. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 27(10), 2017.

Yong Liu, Chenyu Li, Jianmin Wang, and Mingsheng Long. Koopa: Learning non-stationary time series dynamics with koopman predictors. *Advances in neural information processing systems*, 36:12271–12290, 2023.

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.

Lennart Ljung. Perspectives on system identification. *Annual Reviews in Control*, 34(1):1–12, 2010.

Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature communications*, 9(1):4950, 2018.

Igor Mezić. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 41(1):309–325, 2005.

Douglas C Montgomery, Cheryl L Jennings, and Murat Kulahci. *Introduction to time series analysis and forecasting*. John Wiley & Sons, 2015.

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

Clarence W Rowley, Igor Mezić, Shervin Bagheri, Philipp Schlatter, and Dan S Henningson. Spectral analysis of nonlinear flows. *Journal of fluid mechanics*, 641:115–127, 2009.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

Peter J Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, 656:5–28, 2010.

Jingzhe Shi, Qinwei Ma, Huan Ma, and Lei Li. Scaling law for time series forecasting. *Advances in Neural Information Processing Systems*, 37:83314–83344, 2024.

Naoya Takeishi, Yoshinobu Kawahara, and Takehisa Yairi. Learning koopman invariant subspaces for dynamic mode decomposition. *Advances in neural information processing systems*, 30, 2017.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in neural information processing systems*, 34:24804–24816, 2021.

Jonathan H Tu. *Dynamic mode decomposition: Theory and applications*. Princeton University, 2013.

Peter Walters. *An introduction to ergodic theory*, volume 79. Springer Science & Business Media, 2000.

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.

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.

Matthew O Williams, Clarence W Rowley, and Ioannis G Kevrekidis. A kernel-based approach to data-driven koopman spectral analysis. *arXiv preprint arXiv:1411.2260*, 2014.

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

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186*, 2022.

Zhijian Xu, Ailing Zeng, and Qiang Xu. Fits: Modeling time series with $10k$ parameters. In *International Conference on Learning Representations*, volume 2024, pp. 26295–26318, 2024.

Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. Ts2vec: Towards universal representation of time series. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 8980–8987, 2022.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11121–11128, 2023.

Ronghua Zheng, Hanru Bai, and Weiyang Ding. Koonpro: a variance-aware koopman probabilistic model enhanced by neural process for time series forecasting. 2025:29457–29481, 2025.

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. *arXiv preprint arXiv:2012.07436*, 2020.

A PROOF

A.1 PROOF OF PROPOSITION 1

Proof. For every fixed τ, the matrix-valued processes

$$\{x_k x_k^\top\}_{k\in\mathbb{Z}} \qquad \text{and} \qquad \{x_{k+\tau} x_k^\top\}_{k\in\mathbb{Z}} \tag{29}$$

are stationary and ergodic because they are measurable transformations of the underlying stationary and ergodic process. Their entries are integrable. In particular, by the Cauchy–Schwarz inequality and stationarity,

$$\mathbb{E}[|x_{k+\tau,i}x_{k,j}|] \le \sqrt{\mathbb{E}[x_{k+\tau,i}^2]}\sqrt{\mathbb{E}[x_{k,j}^2]} < \infty. \tag{30}$$

The same argument establishes the integrability of the entries of $x_k x_k^\top$.

Applying Birkhoff's ergodic theorem entrywise gives

$$\frac{1}{n} X_n X_n^\top = \frac{1}{n}\sum_{k=1}^{n} x_k x_k^\top \xrightarrow[n\to\infty]{\text{a.s.}} C_0 \tag{31}$$

and

$$\frac{1}{n} X_{\tau,n} X_n^\top = \frac{1}{n}\sum_{k=1}^{n} x_{k+\tau} x_k^\top \xrightarrow[n\to\infty]{\text{a.s.}} C_\tau. \tag{32}$$

Since $C_0 \succ 0$, continuity of the minimum eigenvalue implies

$$\lambda_{\min}\left(\frac{1}{n} X_n X_n^\top\right) \xrightarrow[n\to\infty]{\text{a.s.}} \lambda_{\min}(C_0) > 0. \tag{33}$$

Therefore, with probability one, $X_n X_n^\top$ is invertible for all sufficiently large n. For such n, X_n has full row rank and

$$X_n^\dagger = X_n^\top (X_n X_n^\top)^{-1}. \tag{34}$$

Consequently,

$$\begin{aligned} \widehat{A}_{\tau,n} &= X_{\tau,n} X_n^\top (X_n X_n^\top)^{-1} \\ &= \left(\frac{1}{n} X_{\tau,n} X_n^\top\right)\left(\frac{1}{n} X_n X_n^\top\right)^{-1} \\ &\xrightarrow[n\to\infty]{\text{a.s.}} C_\tau C_0^{-1}, \end{aligned} \tag{35}$$

where the final step follows from the continuity of matrix inversion on the set of nonsingular matrices. ■

A.2 PROOF AND REMARK OF LEMMA 1

Proof. Substituting the definition of the truncated pseudoinverse and using $U_k^\top U_k = I_k$, we obtain

$$\begin{aligned} \widehat{A}_{\tau,k} &= U_k^\top X_{\tau,k} \left(V_k \Sigma_k^{-1} U_k^\top\right) U_k \\ &= U_k^\top X_{\tau,k} V_k \Sigma_k^{-1}. \end{aligned} \tag{36}$$

Substituting $X_{\tau,k} = U_{\tau,k}\Sigma_{\tau,k}V_{\tau,k}^\top$ gives

$$\begin{aligned} \widehat{A}_{\tau,k} &= U_k^\top \left(U_{\tau,k}\Sigma_{\tau,k}V_{\tau,k}^\top\right) V_k \Sigma_k^{-1} \\ &= \left(U_k^\top U_{\tau,k}\right) \Sigma_{\tau,k} \left(V_{\tau,k}^\top V_k\right) \Sigma_k^{-1}. \end{aligned} \tag{37}$$

This proves the stated factorization. ■

Remark. The factorization separates the reduced-order DMD operator into four interpretable components:

$$\widehat{A}_{\tau,k} = \underbrace{\left(U_k^\top U_{\tau,k}\right)}_{\text{spatial alignment}} \underbrace{\Sigma_{\tau,k}}_{\text{response amplitude}} \underbrace{\left(V_{\tau,k}^\top V_k\right)}_{\text{temporal alignment}} \underbrace{\Sigma_k^{-1}}_{\text{predictor scaling}}. \tag{38}$$

1. **Spatial alignment.** The matrix $U_k^\top U_{\tau,k}$ measures the relative orientation of the dominant left singular subspaces extracted from the historical snapshots X and the time-shifted snapshots X_τ. Since $U_{\tau,k}$ is obtained from X_τ, increasing or decreasing τ changes the response subspace that best explains the time-shifted data. As a result, the principal angles between the predictor subspace $\mathrm{span}(U_k)$ and the response subspace $\mathrm{span}(U_{\tau,k})$ vary with τ. Equivalently, the singular values of $U_k^\top U_{\tau,k}$ quantify how well the features learned from the historical snapshots match the spatial structure of the dynamics after a time shift of length τ. In particular, when τ is such that the system exhibits strong temporal coherence, $U_k^\top U_{\tau,k}$ tends to be close to an isometry on the retained subspace, whereas for poorly chosen τ it can become nearly rank-deficient, indicating a mismatch between input and output spatial structures.
2. **Response amplitude.** The diagonal matrix $\Sigma_{\tau,k}$ contains the retained singular values associated with the time-shifted snapshot matrix X_τ. These values represent the strength (energy/magnitude) of the dominant response directions after the time delay τ. Therefore, the τ-dependence of $\Sigma_{\tau,k}$ reflects how much of the time-shifted data lies in the rank-k subspace spanned by $U_{\tau,k}$. If τ increases to the point where the evolution mixes scales or decorrelates the signal, the leading singular values in $\Sigma_{\tau,k}$ can decay, reducing the signal-to-noise ratio in the response part of the factorization. Conversely, for τ aligned with meaningful dynamical timescales (e.g., faster/slower modes), $\Sigma_{\tau,k}$ can preserve larger leading singular values, improving the stability and fidelity of the estimated DMD operator.
3. **Temporal alignment.** The matrix $V_{\tau,k}^\top V_k$ measures the relative orientation of the dominant right singular subspaces extracted from the historical and time-shifted snapshot matrices. While U_k and $U_{\tau,k}$ encode spatial structure, V_k and $V_{\tau,k}$ encode the temporal directions within the data matrices. Since $V_{\tau,k}$ depends on X_τ, changing τ alters the way snapshot-to-snapshot variation is expressed in the retained temporal subspace. The singular values of $V_{\tau,k}^\top V_k$ equal the cosines of principal angles between these temporal subspaces, thus quantifying how coherently the historical time structure matches the time structure observed after a delay τ. In practice, good choices of τ yield strong temporal alignment (singular values near 1), whereas misaligned τ leads to poorer overlap, indicating that the DMD fit is attempting to relate temporally incompatible components.
4. **Predictor scaling.** The factor Σ_k^{-1} depends only on the rank-k SVD of the historical snapshot matrix X (hence it is fixed once X and k are fixed). It rescales the predictor directions according to their singular values. However, the overall influence of this scaling on the estimated operator does depend on τ through the product

$$\left(U_k^\top U_{\tau,k}\right) \Sigma_{\tau,k} \left(V_{\tau,k}^\top V_k\right), \tag{39}$$

because τ changes the components of the response that are mapped into the predictor coordinates. Small retained singular values in Σ_k (i.e., weakly excited predictor directions) can amplify the contribution of any response components that project onto this product. As τ varies, the projection strength can increase or decrease, thereby modulating how strongly the regularization induced by rank truncation influences the estimation. In this sense, τ affects the effective conditioning of the DMD fit by changing how much response energy couples into each predictor mode.

A.3 Proof of Proposition 2

Proof. Fix n and define

$$A_k := X_k^\top X_k \in \mathbb{R}^{n\times n}, \qquad B := \Delta X_\tau^\top X_k \in \mathbb{R}^{n\times n}. \tag{40}$$

Then $\widehat{\mathcal{L}}_{\mathrm{sur},k} = \frac{1}{n}\|B\|_F^2$ and $\widehat{\mathcal{L}}_{\mathrm{consistent},k} = \frac{1}{n}\|\Delta X_\tau^\top (X_k^\top)^\dagger\|_F^2$.

Step 1: SVD-derived identity for the pseudoinverse. Using $X_k = U_k \Sigma_k V_k^\top$ with Σ_k invertible,

$$X_k^\top = V_k \Sigma_k U_k^\top \quad \Rightarrow \quad (X_k^\top)^\dagger = U_k \Sigma_k^{-1} V_k^\top. \tag{41}$$

Also,

$$A_k = X_k^\top X_k = V_k \Sigma_k^2 V_k^\top \quad \Rightarrow \quad A_k^\dagger = V_k \Sigma_k^{-2} V_k^\top. \tag{42}$$

Hence

$$X_k A_k^\dagger = (U_k \Sigma_k V_k^\top)(V_k \Sigma_k^{-2} V_k^\top) = U_k \Sigma_k^{-1} V_k^\top = (X_k^\top)^\dagger. \tag{43}$$

Therefore,

$$\Delta X_\tau^\top (X_k^\top)^\dagger = \Delta X_\tau^\top X_k A_k^\dagger = B A_k^\dagger, \tag{44}$$

and thus

$$\widehat{\mathcal{L}}_{\text{consistent},k} = \frac{1}{n} \|B A_k^\dagger\|_F^2, \qquad \widehat{\mathcal{L}}_{\text{sur},k} = \frac{1}{n} \|B\|_F^2. \tag{45}$$

Step 2: Frobenius-norm sandwich with $\alpha_{\max}, \alpha_{\min}$. Let

$$B = \Delta X_\tau^\top X_k = \Delta X_\tau^\top U_k \Sigma_k V_k^\top =: M V_k^\top, \qquad M := \Delta X_\tau^\top U_k \Sigma_k \in \mathbb{R}^{n \times k}. \tag{46}$$

Since $A_k^\dagger = V_k \Sigma_k^{-2} V_k^\top$,

$$B A_k^\dagger = (M V_k^\top)(V_k \Sigma_k^{-2} V_k^\top) = M \Sigma_k^{-2} V_k^\top. \tag{47}$$

Using orthonormality of V_k,

$$\|B\|_F = \|M V_k^\top\|_F = \|M\|_F, \qquad \|B A_k^\dagger\|_F = \|M \Sigma_k^{-2} V_k^\top\|_F = \|M \Sigma_k^{-2}\|_F. \tag{48}$$

Write $\Sigma_k^{-2} = \operatorname{diag}(\sigma_1^{-2}, \ldots, \sigma_k^{-2})$ and let m_i be the i-th column of M. Then

$$\|M \Sigma_k^{-2}\|_F^2 = \sum_{i=1}^k \sigma_i^{-4} \|m_i\|_2^2. \tag{49}$$

Define $\alpha_i := \sigma_i^2$ and $\alpha_{\max} := \max_{1 \le i \le k} \alpha_i = \sigma_1^2$, $\alpha_{\min} := \min_{1 \le i \le k} \alpha_i = \sigma_k^2$ (all positive). Since $\alpha_{\max}^{-2} \le \alpha_i^{-2} \le \alpha_{\min}^{-2}$ for each i, we get

$$\alpha_{\max}^{-2} \sum_{i=1}^k \|m_i\|_2^2 \;\le\; \sum_{i=1}^k \sigma_i^{-4} \|m_i\|_2^2 \;\le\; \alpha_{\min}^{-2} \sum_{i=1}^k \|m_i\|_2^2. \tag{50}$$

That is,

$$\alpha_{\max}^{-2} \|B\|_F^2 \;\le\; \|B A_k^\dagger\|_F^2 \;\le\; \alpha_{\min}^{-2} \|B\|_F^2. \tag{51}$$

Dividing by n and rewriting in terms of the losses gives, for every n,

$$\alpha_{\max}^{-2} \;\le\; \frac{\widehat{\mathcal{L}}_{\text{consistent},k}}{\widehat{\mathcal{L}}_{\text{sur},k}} \;\le\; \alpha_{\min}^{-2}, \tag{52}$$

with the understanding that if $\widehat{\mathcal{L}}_{\text{sur},k} = 0$ then necessarily $\widehat{\mathcal{L}}_{\text{consistent},k} = 0$ as well (so the ratio is immaterial for liminf/limsup).

Step 3: Relate $\alpha_{\max}, \alpha_{\min}$ to $\lambda_k^{\max}, \lambda_k^{\min}$. The nonzero eigenvalues of $X_k X_k^\top$ and $X_k^\top X_k = A_k$ coincide and equal $\{\sigma_i^2\}_{i=1}^k$. Hence

$$\alpha_{\max} = \sigma_1^2 = \lambda_{\max}(A_k), \qquad \alpha_{\min} = \sigma_k^2 = \lambda_{\min}(A_k), \tag{53}$$

and the assumption

$$\frac{1}{n} X_k X_k^\top \xrightarrow{\text{a.s.}} \Pi_k C_0 \Pi_k \tag{54}$$

implies (on an a.s. event) the convergence of the ordered nonzero eigenvalues:

$$\frac{\alpha_{\max}}{n} \to \lambda_k^{\max}, \qquad \frac{\alpha_{\min}}{n} \to \lambda_k^{\min} \quad \text{a.s.} \tag{55}$$

Since $\lambda_k^{\min} > 0$, $t \mapsto t^{-2}$ is continuous near these limits, so

$$\left(\frac{n}{\alpha_{\max}}\right)^2 \to \frac{1}{(\lambda_k^{\max})^2}, \qquad \left(\frac{n}{\alpha_{\min}}\right)^2 \to \frac{1}{(\lambda_k^{\min})^2} \quad \text{a.s.} \tag{56}$$

Step 4: Apply liminf/limsup to the ratio bound. From Step 2,

$$\alpha_{\max}^{-2} \;\le\; \frac{\widehat{\mathcal{L}}_{\text{consistent},k}}{\widehat{\mathcal{L}}_{\text{sur},k}} \;\le\; \alpha_{\min}^{-2}. \tag{57}$$

Multiplying by n^2 yields

$$\left(\frac{n}{\alpha_{\max}}\right)^2 \le n^2 \frac{\widehat{\mathcal{L}}_{\text{consistent},k}}{\widehat{\mathcal{L}}_{\text{sur},k}} \le \left(\frac{n}{\alpha_{\min}}\right)^2. \tag{58}$$

Taking $\liminf$ and using the almost sure convergence of the left endpoint gives

$$\liminf_{n\to\infty} n^2 \frac{\widehat{\mathcal{L}}_{\text{consistent},k}}{\widehat{\mathcal{L}}_{\text{sur},k}} \ge \frac{1}{(\lambda_k^{\max})^2} \quad \text{a.s.} \tag{59}$$

Similarly, taking $\limsup$ yields

$$\limsup_{n\to\infty} n^2 \frac{\widehat{\mathcal{L}}_{\text{consistent},k}}{\widehat{\mathcal{L}}_{\text{sur},k}} \le \frac{1}{(\lambda_k^{\min})^2} \quad \text{a.s.} \tag{60}$$

This completes the proof. ■

B BASELINES

We select SOTA baselines with diverse backbone designs, and briefly summarize the core idea of each method as follows:

1. **TimesNet** (Wu et al., 2022): Employs a 2D kernel to uncover intra-period and inter-period variations in the frequency domain, and further refines and fuses these variations through an adaptive aggregation module to enhance forecasting.
2. **DLinear** (Zeng et al., 2023): Decomposes the series into trend and seasonality components and feeds both into a single linear layer for forecasting.
3. **iTransformer** (Liu et al., 2024): Employs an inverted Transformer that treats each channel (variable) as a token to primarily capture cross-variable correlations, and then applies a feed-forward network over these tokens to improve multivariate forecasting efficiency.
4. **TimeMixer** (Wang et al., 2024a): Integrates past-decomposable mixing and future-multipredictor mixing to capture temporal patterns relevant to both the observed history and the prediction horizon, and further models these patterns at multiple time granularities to improve forecasting.
5. **TimeXer** (Wang et al., 2024b): Builds an exogenous module that explicitly performs channel-wise attention for each channel, and integrates it with an endogenous module to learn global patterns for forecasting the exogenous variables.
6. **TimeFilter** (Hu et al., 2025): Treats multivariable time series as channel-independent data, uses graph neural networks to capture multi-scale spatiotemporal dependencies across channels, and applies learned filtering to suppress irrelevant components, thereby improving forecasting.

C HYPERPARAMETERS

Table 3: Selection of α in Relinear (Part I).

Datasets	ETTh1				ETTh2				ETTm1				ETTm2			
input \ output	96	192	336	720	96	192	336	720	96	192	336	720	96	192	336	720
96	1×10^{2}	4×10^{1}	1×10^{2}	1×10^{3}	6×10^{1}	1×10^{0}	1×10^{-1}	1×10^{0}	1×10^{1}	2×10^{2}	5×10^{1}	2×10^{2}	5×10^{1}	1×10^{0}	1×10^{0}	1×10^{1}
192	5×10^{2}	4×10^{2}	1×10^{2}	1×10^{2}	7.5×10^{1}	2×10^{1}	1×10^{-2}	1×10^{0}	5×10^{1}	1×10^{2}	5×10^{1}	5×10^{1}	2×10^{1}	3×10^{1}	1×10^{1}	1×10^{0}
336	2×10^{3}	1×10^{3}	1×10^{3}	1×10^{3}	3.5×10^{2}	2×10^{1}	1×10^{2}	5×10^{1}	5×10^{2}	1×10^{3}	1×10^{3}	1×10^{3}	2×10^{3}	5×10^{2}	2×10^{2}	2×10^{2}
720	5×10^{4}	5×10^{4}	5×10^{3}	1.5×10^{4}	5×10^{2}	3×10^{1}	1.5×10^{1}	4×10^{1}	1×10^{3}	5×10^{3}	1×10^{2}	5×10^{1}	1.5×10^{5}	5×10^{2}	5×10^{2}	5×10^{2}

Table 4: Selection of α in Relinear (Part II).

Datasets	Weather				ECL				Traffic			
input \ output	96	192	336	720	96	192	336	720	96	192	336	720
96	1×10^{1}	3×10^{1}	1×10^{2}	5×10^{2}	5×10^{-1}	1×10^{0}	2×10^{0}	2×10^{0}	1×10^{-1}	1×10^{-1}	1×10^{-1}	1×10^{-1}
192	1×10^{1}	5×10^{2}	1×10^{2}	5×10^{2}	1×10^{1}	1×10^{0}	1×10^{0}	1×10^{0}	5×10^{-1}	1×10^{0}	1×10^{0}	1×10^{0}
336	1×10^{4}	1×10^{3}	1×10^{3}	1×10^{3}	2×10^{1}	2×10^{0}	2×10^{0}	2×10^{0}	2×10^{0}	5×10^{-1}	5×10^{-1}	5×10^{-1}
720	1×10^{4}	1×10^{4}	1×10^{4}	5×10^{2}	1×10^{0}	1×10^{0}	1×10^{0}	1×10^{0}	1×10^{0}	1×10^{0}	1×10^{0}	1×10^{0}

Table 5: Selection of the overlap window in Relinear (Part I).

Datasets	ETTh1				ETTh2				ETTm1				ETTm2			
input \ output	96	192	336	720	96	192	336	720	96	192	336	720	96	192	336	720
96	36	2	72	96	96	96	48	96	2	48	84	12	96	96	96	96
192	48	96	144	72	192	96	168	96	72	48	168	2	24	120	48	192
336	240	288	336	288	336	24	288	336	336	48	192	48	288	192	240	48
720	288	288	192	288	720	720	384	192	624	720	288	288	384	48	384	720

Table 6: Selection of the overlap window in Relinear (Part II).

Datasets	Weather				ECL				Traffic			
input \ output	96	192	336	720	96	192	336	720	96	192	336	720
96	96	24	12	36	96	24	12	36	96	96	2	48
192	192	192	192	192	192	192	192	192	2	2	48	48
336	288	288	48	48	288	288	48	48	48	288	336	48
720	720	192	384	480	720	192	384	480	96	288	96	96

Table 7: Selection of α in baselines.

Dataset \ Model	TimeMixer	TimeXer	iTransfomer	TimeFilter	TimesNet
ETTh1	1×10^{-3}	1×10^{-2}	1×10^{-2}	1×10^{-3}	1×10^{-3}
ETTh2	1×10^{-3}	1×10^{-2}	1×10^{-2}	1×10^{-3}	1×10^{-3}
ETTm1	1×10^{-3}	1×10^{-2}	1×10^{-2}	1×10^{-3}	1×10^{-3}
ETTm2	1×10^{-3}	1×10^{-2}	1×10^{-2}	1×10^{-3}	1×10^{-3}
Weather	1×10^{-5}	1×10^{-6}	1×10^{-3}	1×10^{-5}	1×10^{-5}
ECL	1×10^{-6}	1×10^{-5}	1×10^{-4}	1×10^{-5}	1×10^{-5}
Traffic	1×10^{-5}	1×10^{-5}	1×10^{-8}	1×10^{-7}	1×10^{-5}

Table 8: Selection of overlap window in baselines.

Models		TimeMixer	TimeXer	iTransfomer	TimeFilter	TimesNet
ETTh1	96	48	60	24	24	24
	192	60	96	36	24	96
	336	24	24	24	12	72
	720	48	96	2	12	48
ETTh2	96	72	24	96	48	24
	192	96	12	84	24	96
	336	24	96	60	12	72
	720	36	24	84	60	48
ETTm1	96	24	72	36	24	60
	192	84	48	84	96	24
	336	24	84	12	36	32
	720	12	84	60	84	84
ETTm2	96	96	2	48	48	24
	192	84	96	72	48	96
	336	2	12	60	24	84
	720	96	72	36	96	60
Weather	96	2	60	48	96	2
	192	24	48	36	48	72
	336	48	96	96	72	12
	720	12	24	48	48	24
ECL	96	12	84	84	48	96
	192	12	48	84	48	60
	336	2	84	48	72	96
	720	2	60	24	48	48
Traffic	96	96	96	60	48	48
	192	48	24	24	48	96
	336	48	48	60	24	48
	720	48	48	48	60	48

D EXPERIMENT RESULTS

Table 9: Input length = 96.

Models		ReLinear+		Relinear		DLinear		TimeXer		iTransfomer		TimeFilter		TimesNet		TimeMixer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.384	0.398	0.390	0.406	0.443	0.442	0.390	0.400	0.378	0.397	0.382	0.396	0.436	0.445	0.383	0.399
	192	0.435	0.430	0.437	0.432	0.484	0.466	0.445	0.432	0.435	0.433	0.438	0.431	0.473	0.484	0.441	0.430
	336	0.479	0.457	0.484	0.461	0.518	0.487	0.482	0.452	0.476	0.451	0.487	0.452	0.525	0.488	0.495	0.458
	720	0.509	0.507	0.517	0.511	0.540	0.527	0.501	0.491	0.511	0.490	0.491	0.467	0.528	0.502	0.476	0.470
	Avg	0.452	0.448	0.457	0.453	0.496	0.481	0.455	0.444	0.450	0.443	0.450	0.437	0.491	0.480	0.449	0.439
ETTh2	96	0.290	0.344	0.297	0.355	0.341	0.395	0.296	0.343	0.294	0.345	0.288	0.339	0.330	0.368	0.294	0.347
	192	0.377	0.400	0.390	0.414	0.482	0.323	0.379	0.401	0.379	0.399	0.379	0.397	0.395	0.409	0.391	0.404
	336	0.429	0.446	0.543	0.461	0.593	0.542	0.415	0.425	0.434	0.436	0.416	0.432	0.457	0.456	0.423	0.436
	720	0.484	0.492	0.578	0.542	0.840	0.661	0.441	0.450	0.471	0.469	0.419	0.441	0.451	0.459	0.448	0.454
	Avg	0.395	0.421	0.452	0.443	0.564	0.480	0.383	0.405	0.395	0.412	0.376	0.402	0.408	0.423	0.389	0.410
ETTm1	96	0.339	0.370	0.349	0.397	0.346	0.374	0.330	0.367	0.340	0.370	0.319	0.358	0.341	0.378	0.320	0.360
	192	0.381	0.391	0.382	0.393	0.382	0.391	0.380	0.394	0.399	0.401	0.361	0.380	0.382	0.400	0.367	0.388
	336	0.414	0.417	0.418	0.413	0.415	0.415	0.406	0.414	0.436	0.427	0.390	0.402	0.415	0.422	0.391	0.404
	720	0.474	0.454	0.486	0.468	0.473	0.451	0.458	0.444	0.506	0.465	0.459	0.446	0.482	0.457	0.470	0.452
	Avg	0.402	0.408	0.409	0.418	0.404	0.408	0.394	0.405	0.420	0.416	0.382	0.397	0.405	0.414	0.387	0.401
ETTm2	96	0.181	0.268	0.186	0.276	0.193	0.293	0.177	0.258	0.185	0.266	0.171	0.257	0.187	0.265	0.176	0.259
	192	0.245	0.309	0.255	0.330	0.293	0.361	0.247	0.305	0.253	0.311	0.236	0.299	0.251	0.308	0.238	0.298
	336	0.309	0.350	0.321	0.377	0.385	0.429	0.303	0.343	0.325	0.358	0.295	0.337	0.322	0.349	0.292	0.335
	720	0.413	0.417	0.456	0.462	0.556	0.523	0.410	0.403	0.418	0.409	0.394	0.395	0.417	0.404	0.392	0.395
	Avg	0.287	0.336	0.305	0.361	0.357	0.402	0.284	0.327	0.295	0.336	0.274	0.322	0.294	0.332	0.275	0.322
Weather	96	0.194	0.256	0.198	0.261	0.196	0.255	0.159	0.206	0.172	0.211	0.161	0.206	0.172	0.223	0.161	0.209
	192	0.235	0.293	0.238	0.297	0.237	0.296	0.207	0.250	0.227	0.259	0.217	0.257	0.229	0.269	0.208	0.253
	336	0.281	0.331	0.288	0.341	0.283	0.335	0.262	0.290	0.281	0.299	0.272	0.295	0.277	0.301	0.264	0.294
	720	0.346	0.382	0.353	0.392	0.355	0.385	0.346	0.348	0.357	0.349	0.349	0.345	0.358	0.353	0.344	0.344
	Avg	0.264	0.316	0.269	0.323	0.267	0.318	0.244	0.274	0.259	0.280	0.250	0.276	0.259	0.287	0.244	0.275
ECL	96	0.194	0.256	0.197	0.280	0.224	0.308	0.140	0.242	0.150	0.242	0.141	0.235	0.169	0.274	0.156	0.247
	192	0.194	0.280	0.197	0.285	0.222	0.310	0.157	0.256	0.166	0.257	0.158	0.250	0.188	0.291	0.170	0.260
	336	0.207	0.297	0.209	0.298	0.242	0.327	0.172	0.269	0.181	0.273	0.166	0.263	0.200	0.301	0.187	0.278
	720	0.242	0.330	0.246	0.333	0.270	0.352	0.211	0.307	0.214	0.302	0.199	0.295	0.273	0.355	0.228	0.312
	Avg	0.209	0.291	0.212	0.299	0.240	0.324	0.170	0.269	0.178	0.269	0.166	0.261	0.208	0.305	0.185	0.274
Traffic	96	0.648	0.397	0.685	0.417	0.696	0.429	0.447	0.270	0.397	0.271	0.397	0.272	0.589	0.315	0.463	0.293
	192	0.598	0.370	0.630	0.386	0.646	0.407	0.465	0.279	0.416	0.279	0.412	0.269	0.618	0.323	0.507	0.301
	336	0.605	0.374	0.637	0.389	0.653	0.409	0.483	0.289	0.432	0.287	0.435	0.286	0.649	0.352	0.526	0.308
	720	0.645	0.395	0.660	0.410	0.724	0.442	0.524	0.314	0.465	0.304	0.462	0.295	0.678	0.367	0.565	0.324
	Avg	0.624	0.384	0.653	0.401	0.680	0.422	0.480	0.288	0.428	0.285	0.427	0.281	0.634	0.339	0.515	0.307

Table 10: Input length = 192.

Models		ReLinear+		ReLinear		DLinear		TimeXer		iTransfomer		TimeFilter		TimesNet		TimeMixer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.378	0.396	0.393	0.412	0.381	0.425	0.384	0.399	0.393	0.411	0.387	0.409	0.428	0.443	0.395	0.411
	192	0.422	0.422	0.449	0.452	0.450	0.448	0.435	0.429	0.439	0.438	0.429	0.427	0.455	0.453	0.431	0.429
	336	0.462	0.453	0.559	0.535	0.477	0.465	0.457	0.439	0.468	0.456	0.461	0.442	0.474	0.462	0.457	0.438
	720	0.488	0.497	0.541	0.536	0.496	0.507	0.528	0.501	0.498	0.497	0.513	0.490	0.482	0.485	0.497	0.481
	Avg	0.438	0.442	0.486	0.484	0.451	0.461	0.451	0.442	0.450	0.451	0.448	0.442	0.460	0.461	0.445	0.440
ETTh2	96	0.282	0.342	0.299	0.360	0.304	0.365	0.294	0.349	0.307	0.359	0.326	0.368	0.324	0.375	0.287	0.346
	192	0.362	0.399	0.389	0.419	0.403	0.428	0.366	0.399	0.377	0.402	0.385	0.412	0.421	0.421	0.396	0.413
	336	0.403	0.432	0.410	0.436	0.491	0.485	0.393	0.423	0.411	0.429	0.397	0.426	0.495	0.485	0.393	0.419
	720	0.488	0.524	0.554	0.520	0.742	0.616	0.425	0.449	0.430	0.449	0.432	0.451	0.459	0.468	0.443	0.461
	Avg	0.384	0.424	0.413	0.434	0.485	0.474	0.370	0.405	0.381	0.410	0.385	0.414	0.425	0.437	0.380	0.410
ETTm1	96	0.307	0.347	0.316	0.360	0.307	0.346	0.304	0.348	0.301	0.348	0.309	0.350	0.330	0.376	0.309	0.355
	192	0.341	0.367	0.353	0.376	0.342	0.369	0.345	0.375	0.343	0.374	0.344	0.379	0.380	0.400	0.352	0.383
	336	0.379	0.394	0.402	0.421	0.378	0.392	0.383	0.399	0.381	0.397	0.371	0.397	0.392	0.411	0.370	0.400
	720	0.443	0.437	0.488	0.483	0.446	0.438	0.439	0.433	0.439	0.431	0.434	0.434	0.438	0.440	0.448	0.437
	Avg	0.368	0.386	0.390	0.410	0.368	0.386	0.368	0.389	0.366	0.388	0.365	0.390	0.385	0.407	0.370	0.394
ETTm2	96	0.171	0.260	0.184	0.272	0.186	0.285	0.170	0.256	0.180	0.269	0.176	0.259	0.193	0.277	0.172	0.258
	192	0.231	0.298	0.269	0.344	0.246	0.323	0.229	0.296	0.246	0.311	0.224	0.292	0.241	0.310	0.237	0.301
	336	0.287	0.341	0.295	0.348	0.342	0.399	0.287	0.332	0.299	0.344	0.285	0.334	0.318	0.352	0.290	0.339
	720	0.385	0.399	0.402	0.414	0.487	0.483	0.378	0.390	0.383	0.394	0.377	0.390	0.406	0.408	0.372	0.387
	Avg	0.269	0.325	0.288	0.345	0.315	0.373	0.266	0.319	0.277	0.330	0.266	0.319	0.290	0.337	0.268	0.321
Weather	96	0.184	0.251	0.189	0.255	0.184	0.243	0.155	0.205	0.168	0.214	0.153	0.202	0.180	0.231	0.152	0.199
	192	0.224	0.282	0.251	0.318	0.226	0.285	0.198	0.245	0.213	0.252	0.198	0.249	0.225	0.266	0.197	0.243
	336	0.269	0.318	0.307	0.368	0.271	0.322	0.253	0.285	0.268	0.292	0.257	0.291	0.287	0.306	0.249	0.282
	720	0.333	0.371	0.385	0.430	0.335	0.373	0.328	0.337	0.341	0.345	0.329	0.337	0.353	0.355	0.327	0.336
	Avg	0.253	0.306	0.283	0.343	0.254	0.306	0.234	0.268	0.248	0.276	0.234	0.270	0.261	0.290	0.231	0.265
ECL	96	0.152	0.247	0.154	0.251	0.177	0.283	0.136	0.235	0.138	0.234	0.143	0.238	0.169	0.274	0.133	0.226
	192	0.163	0.258	0.166	0.263	0.188	0.293	0.155	0.251	0.157	0.251	0.158	0.252	0.185	0.290	0.153	0.244
	336	0.179	0.276	0.183	0.281	0.203	0.309	0.176	0.273	0.173	0.268	0.173	0.268	0.196	0.299	0.172	0.262
	720	0.213	0.308	0.217	0.313	0.236	0.338	0.210	0.300	0.208	0.300	0.216	0.306	0.281	0.359	0.209	0.297
	Avg	0.177	0.272	0.180	0.277	0.201	0.306	0.169	0.265	0.169	0.263	0.173	0.266	0.208	0.306	0.167	0.257
Traffic	96	0.450	0.302	0.456	0.313	0.451	0.304	0.395	0.272	0.368	0.261	0.361	0.253	0.588	0.316	0.397	0.273
	192	0.458	0.306	0.465	0.317	0.461	0.309	0.419	0.284	0.393	0.271	0.389	0.266	0.611	0.328	0.422	0.287
	336	0.473	0.314	0.482	0.327	0.478	0.321	0.438	0.286	0.404	0.278	0.400	0.268	0.633	0.335	0.446	0.324
	720	0.503	0.333	0.512	0.347	0.508	0.340	0.476	0.304	0.437	0.295	0.431	0.286	0.646	0.340	0.470	0.297
	Avg	0.471	0.314	0.479	0.326	0.475	0.319	0.432	0.287	0.401	0.276	0.395	0.268	0.620	0.330	0.434	0.295

Table 11: Input length = 336.

Models		ReLinear+		ReLinear		DLinear		TimeXer		iTransfomer		TimeFilter		TimesNet		TimeMixer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.373	0.395	0.411	0.435	0.397	0.417	0.381	0.401	0.393	0.413	0.389	0.410	0.444	0.455	0.397	0.415
	192	0.406	0.416	0.456	0.465	0.435	0.441	0.421	0.425	0.428	0.435	0.431	0.436	0.477	0.472	0.413	0.423
	336	0.436	0.442	0.466	0.468	0.457	0.463	0.493	0.487	0.449	0.451	0.449	0.450	0.489	0.478	0.447	0.449
	720	0.479	0.497	0.587	0.568	0.477	0.499	0.550	0.469	0.509	0.508	0.556	0.520	0.533	0.518	0.538	0.506
	Avg	0.424	0.438	0.480	0.484	0.442	0.455	0.461	0.446	0.445	0.452	0.456	0.454	0.486	0.481	0.449	0.448
ETTh2	96	0.275	0.338	0.304	0.365	0.285	0.351	0.290	0.351	0.302	0.361	0.298	0.358	0.392	0.426	0.288	0.351
	192	0.347	0.391	0.381	0.413	0.386	0.419	0.359	0.398	0.368	0.401	0.362	0.401	0.426	0.446	0.384	0.406
	336	0.382	0.416	0.436	0.455	0.459	0.471	0.378	0.415	0.394	0.422	0.402	0.428	0.396	0.430	0.395	0.419
	720	0.515	0.509	0.598	0.553	0.735	0.609	0.420	0.445	0.419	0.447	0.426	0.448	0.449	0.470	0.430	0.452
	Avg	0.380	0.414	0.430	0.447	0.466	0.463	0.362	0.402	0.371	0.408	0.372	0.409	0.416	0.443	0.374	0.407
ETTm1	96	0.301	0.347	0.315	0.361	0.301	0.344	0.303	0.353	0.301	0.352	0.301	0.355	0.331	0.380	0.301	0.350
	192	0.336	0.367	0.362	0.399	0.335	0.365	0.348	0.382	0.336	0.374	0.349	0.382	0.373	0.395	0.339	0.375
	336	0.371	0.389	0.392	0.415	0.380	0.398	0.380	0.402	0.373	0.395	0.366	0.393	0.389	0.413	0.386	0.402
	720	0.428	0.426	0.511	0.500	0.437	0.434	0.437	0.434	0.438	0.433	0.429	0.430	0.487	0.465	0.448	0.440
	Avg	0.359	0.382	0.395	0.419	0.363	0.385	0.367	0.393	0.362	0.389	0.361	0.390	0.395	0.413	0.369	0.392
ETTm2	96	0.164	0.257	0.175	0.269	0.170	0.266	0.169	0.258	0.178	0.268	0.166	0.254	0.189	0.274	0.171	0.263
	192	0.219	0.301	0.248	0.327	0.231	0.313	0.226	0.296	0.237	0.308	0.238	0.306	0.245	0.315	0.234	0.302
	336	0.273	0.334	0.310	0.361	0.304	0.366	0.281	0.331	0.288	0.341	0.293	0.345	0.316	0.353	0.294	0.343
	720	0.359	0.394	0.403	0.417	0.441	0.451	0.371	0.393	0.380	0.396	0.370	0.394	0.396	0.406	0.381	0.395
	Avg	0.255	0.322	0.284	0.344	0.287	0.349	0.262	0.320	0.271	0.328	0.267	0.325	0.287	0.337	0.270	0.326
Weather	96	0.174	0.235	0.187	0.254	0.175	0.236	0.152	0.205	0.165	0.213	0.152	0.204	0.172	0.229	0.149	0.200
	192	0.215	0.274	0.224	0.283	0.217	0.277	0.194	0.243	0.204	0.252	0.197	0.250	0.213	0.262	0.190	0.240
	336	0.260	0.310	0.296	0.355	0.264	0.318	0.244	0.282	0.252	0.287	0.248	0.287	0.267	0.304	0.243	0.281
	720	0.324	0.362	0.372	0.420	0.326	0.366	0.317	0.333	0.327	0.339	0.322	0.342	0.340	0.351	0.320	0.331
	Avg	0.243	0.295	0.270	0.328	0.246	0.299	0.227	0.266	0.237	0.273	0.230	0.271	0.248	0.287	0.226	0.263
ECL	96	0.140	0.237	0.145	0.246	0.157	0.263	0.140	0.241	0.135	0.232	0.137	0.232	0.168	0.265	0.132	0.226
	192	0.154	0.250	0.159	0.259	0.165	0.272	0.160	0.256	0.155	0.251	0.150	0.243	0.185	0.290	0.151	0.242
	336	0.169	0.268	0.175	0.277	0.185	0.291	0.181	0.280	0.173	0.272	0.168	0.261	0.201	0.305	0.169	0.260
	720	0.204	0.301	0.211	0.311	0.219	0.321	0.215	0.308	0.204	0.296	0.194	0.285	0.224	0.320	0.205	0.293
	Avg	0.167	0.264	0.173	0.273	0.182	0.287	0.174	0.271	0.167	0.263	0.162	0.255	0.195	0.295	0.164	0.255
Traffic	96	0.410	0.283	0.419	0.300	0.412	0.287	0.374	0.262	0.361	0.262	0.351	0.250	0.592	0.319	0.399	0.288
	192	0.423	0.289	0.431	0.304	0.424	0.291	0.399	0.273	0.383	0.270	0.378	0.262	0.616	0.337	0.406	0.284
	336	0.436	0.296	0.446	0.314	0.439	0.301	0.412	0.281	0.394	0.275	0.393	0.270	0.618	0.342	✗	✗
	720	0.466	0.316	0.475	0.330	0.469	0.320	0.443	0.299	0.449	0.306	0.426	0.287	0.654	0.351	✗	✗
	Avg	0.434	0.296	0.443	0.312	0.436	0.300	0.407	0.279	0.397	0.278	0.387	0.267	0.620	0.337	✗	✗

Table 12: Input length = 720.

Models		ReLinear+		ReLinear		DLinear		TimeXer		iTransfomer		TimeFilter		TimesNet		TimeMixer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.370	0.399	0.398	0.429	0.376	0.404	0.395	0.423	0.396	0.427	0.387	0.414	0.455	0.464	0.386	0.417
	192	0.406	0.423	0.429	0.442	0.409	0.424	0.419	0.438	0.443	0.458	0.430	0.441	0.457	0.470	0.425	0.445
	336	0.442	0.452	0.525	0.513	0.443	0.450	0.471	0.472	0.474	0.462	0.464	0.460	0.501	0.503	0.596	0.549
	720	0.469	0.495	0.507	0.517	0.477	0.500	0.512	0.510	0.546	0.537	0.666	0.581	0.613	0.563	0.797	0.839
	Avg	0.421	0.442	0.465	0.475	0.426	0.445	0.449	0.461	0.465	0.471	0.487	0.474	0.507	0.500	0.551	0.563
ETTh2	96	0.269	0.337	0.320	0.377	0.288	0.356	0.293	0.357	0.307	0.366	0.293	0.353	0.427	0.451	0.298	0.356
	192	0.333	0.384	0.384	0.418	0.420	0.445	0.354	0.395	0.376	0.407	0.389	0.414	0.422	0.439	0.525	0.501
	336	0.416	0.447	0.474	0.474	0.518	0.501	0.389	0.425	0.416	0.438	0.413	0.430	0.423	0.449	0.383	0.422
	720	0.453	0.491	0.619	0.563	0.805	0.634	0.435	0.459	0.439	0.465	0.458	0.476	0.447	0.468	✗	✗
	Avg	0.368	0.415	0.449	0.458	0.508	0.484	0.368	0.409	0.385	0.419	0.388	0.418	0.430	0.452	✗	✗
ETTm1	96	0.305	0.347	0.326	0.375	0.308	0.351	0.311	0.363	0.312	0.365	0.307	0.362	0.345	0.392	0.312	0.363
	192	0.327	0.359	0.375	0.414	0.337	0.368	0.353	0.391	0.345	0.382	0.336	0.382	0.388	0.408	0.342	0.376
	336	0.370	0.392	0.397	0.421	0.372	0.393	0.374	0.404	0.379	0.403	0.372	0.406	0.401	0.421	0.380	0.393
	720	0.425	0.428	0.520	0.514	0.426	0.425	0.430	0.430	0.425	0.430	0.466	0.447	0.466	0.459	0.426	0.423
	Avg	0.359	0.383	0.405	0.431	0.361	0.384	0.367	0.397	0.365	0.395	0.370	0.399	0.400	0.420	0.365	0.389
ETTm2	96	0.160	0.254	0.208	0.308	0.161	0.253	0.179	0.264	0.178	0.271	0.185	0.271	0.200	0.292	0.182	0.274
	192	0.216	0.293	0.265	0.347	0.217	0.295	0.248	0.309	0.236	0.313	0.254	0.313	0.260	0.332	0.238	0.307
	336	0.269	0.332	0.313	0.374	0.291	0.352	0.317	0.357	0.285	0.344	0.360	0.376	0.305	0.361	0.286	0.343
	720	0.359	0.394	0.365	0.396	0.401	0.428	0.389	0.405	0.388	0.409	0.423	0.428	0.398	0.421	0.386	0.404
	Avg	0.252	0.318	0.288	0.356	0.268	0.332	0.283	0.334	0.272	0.334	0.306	0.347	0.291	0.352	0.273	0.332
Weather	96	0.168	0.225	0.194	0.274	0.171	0.234	0.156	0.213	0.166	0.218	0.148	0.203	0.174	0.233	0.148	0.204
	192	0.209	0.264	0.225	0.287	0.215	0.277	0.203	0.252	0.209	0.257	0.205	0.254	0.218	0.269	0.201	0.242
	336	0.251	0.301	0.260	0.312	0.256	0.305	0.251	0.292	0.260	0.295	0.263	0.307	0.275	0.309	0.250	0.287
	720	0.310	0.349	0.361	0.408	0.316	0.356	0.314	0.337	0.324	0.344	0.321	0.343	0.339	0.354	0.319	0.334
	Avg	0.234	0.277	0.260	0.320	0.240	0.293	0.231	0.274	0.240	0.279	0.234	0.277	0.252	0.291	0.230	0.267
ECL	96	0.133	0.231	0.139	0.241	0.144	0.243	0.142	0.247	0.136	0.234	0.139	0.234	0.187	0.289	0.148	0.204
	192	0.148	0.245	0.152	0.253	0.158	0.256	0.160	0.261	0.160	0.259	0.152	0.249	0.195	0.296	0.267	0.359
	336	0.163	0.262	0.170	0.274	0.173	0.272	0.177	0.280	0.180	0.280	0.166	0.263	0.205	0.307	✗	✗
	720	0.197	0.295	0.204	0.303	0.208	0.295	0.203	0.294	0.202	0.299	0.199	0.294	0.221	0.281	✗	✗
	Avg	0.160	0.258	0.166	0.268	0.171	0.267	0.171	0.271	0.170	0.268	0.164	0.260	0.202	0.293	✗	✗
Traffic	96	0.386	0.272	0.397	0.289	0.386	0.273	0.362	0.261	0.351	0.257	0.348	0.251	0.595	0.322	✗	✗
	192	0.397	0.276	0.406	0.294	0.397	0.277	0.380	0.272	0.370	0.266	0.363	0.255	0.606	0.322	✗	✗
	336	0.410	0.284	0.421	0.301	0.412	0.286	0.403	0.283	0.393	0.278	0.380	0.264	0.628	0.333	✗	✗
	720	0.450	0.307	0.462	0.323	0.455	0.312	0.440	0.302	0.429	0.297	0.420	0.284	0.647	0.352	✗	✗
	Avg	0.411	0.285	0.422	0.302	0.413	0.287	0.396	0.280	0.386	0.275	0.378	0.264	0.619	0.332	✗	✗

Table 13: Comparison of several deep learning models trained with ReAP and the official settings. The input length is 96. The + suffix denotes models trained with ReAP, Red denotes no improvement.

Models		TimeMixer		TimeMixer+		TimeXer		TimeXer+		iTransfomer		iTransfomer+		TimeFilter		TimeFilter+		TimesNet		TimesNet+	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.383	0.399	0.376	0.392	0.390	0.400	0.383	0.398	0.378	0.397	0.377	0.398	0.382	0.396	0.380	0.392	0.436	0.445	0.394	0.409
	192	0.441	0.430	0.423	0.421	0.445	0.432	0.437	0.431	0.435	0.433	0.430	0.431	0.438	0.431	0.436	0.422	0.473	0.484	0.438	0.437
	336	0.495	0.458	0.479	0.444	0.482	0.452	0.478	0.456	0.476	0.451	0.471	0.453	0.487	0.452	0.480	0.446	0.525	0.488	0.476	0.453
	720	0.496	0.490	0.479	0.469	0.501	0.491	0.496	0.479	0.511	0.490	0.495	0.480	0.491	0.467	0.471	0.463	0.528	0.502	0.484	0.469
	Avg	0.454	0.444	0.439	0.431	0.455	0.444	0.449	0.441	0.450	0.443	0.443	0.441	0.450	0.436	0.442	0.431	0.491	0.480	0.448	0.442
ETTh2	96	0.294	0.347	0.289	0.340	0.296	0.343	0.288	0.341	0.294	0.345	0.297	0.350	0.288	0.339	0.286	0.338	0.330	0.368	0.309	0.356
	192	0.391	0.404	0.367	0.389	0.379	0.401	0.370	0.392	0.379	0.399	0.370	0.396	0.379	0.397	0.368	0.388	0.395	0.409	0.383	0.400
	336	0.423	0.436	0.421	0.432	0.415	0.425	0.422	0.431	0.434	0.436	0.420	0.432	0.416	0.432	0.411	0.424	0.457	0.456	0.423	0.433
	720	0.448	0.454	0.424	0.442	0.441	0.450	0.409	0.429	0.471	0.469	0.422	0.443	0.419	0.441	0.415	0.437	0.451	0.459	0.421	0.439
	Avg	0.389	0.410	0.375	0.401	0.383	0.405	0.372	0.398	0.395	0.412	0.377	0.405	0.376	0.402	0.370	0.397	0.408	0.423	0.384	0.407
ETTm1	96	0.320	0.360	0.310	0.352	0.330	0.367	0.319	0.358	0.340	0.370	0.320	0.358	0.319	0.358	0.319	0.356	0.341	0.378	0.328	0.367
	192	0.367	0.388	0.359	0.381	0.380	0.394	0.361	0.386	0.399	0.401	0.367	0.383	0.361	0.380	0.353	0.380	0.382	0.400	0.374	0.391
	336	0.391	0.404	0.389	0.401	0.406	0.414	0.394	0.408	0.436	0.427	0.403	0.407	0.390	0.402	0.386	0.400	0.415	0.422	0.409	0.416
	720	0.470	0.452	0.449	0.438	0.458	0.444	0.445	0.439	0.506	0.465	0.466	0.444	0.459	0.446	0.449	0.441	0.482	0.457	0.475	0.450
	Avg	0.387	0.401	0.377	0.393	0.394	0.405	0.380	0.398	0.420	0.416	0.389	0.398	0.382	0.396	0.377	0.394	0.405	0.414	0.397	0.406
ETTm2	96	0.176	0.259	0.172	0.255	0.177	0.258	0.170	0.253	0.185	0.266	0.176	0.260	0.171	0.257	0.169	0.255	0.187	0.265	0.179	0.259
	192	0.238	0.298	0.238	0.298	0.247	0.305	0.237	0.300	0.253	0.311	0.241	0.302	0.236	0.299	0.235	0.298	0.251	0.308	0.248	0.304
	336	0.292	0.335	0.292	0.334	0.303	0.343	0.295	0.338	0.325	0.358	0.304	0.343	0.295	0.337	0.293	0.335	0.322	0.349	0.310	0.342
	720	0.392	0.395	0.392	0.395	0.410	0.403	0.394	0.398	0.418	0.409	0.400	0.398	0.394	0.395	0.391	0.394	0.417	0.404	0.410	0.401
	Avg	0.275	0.322	0.273	0.320	0.284	0.327	0.274	0.322	0.295	0.336	0.280	0.326	0.274	0.322	0.272	0.320	0.294	0.332	0.287	0.327
Weather	96	0.161	0.209	0.161	0.207	0.159	0.206	0.157	0.205	0.172	0.211	0.177	0.215	0.161	0.206	0.160	0.205	0.172	0.223	0.169	0.219
	192	0.208	0.252	0.207	0.249	0.207	0.250	0.205	0.249	0.227	0.259	0.224	0.256	0.217	0.257	0.208	0.248	0.229	0.269	0.212	0.255
	336	0.264	0.294	0.262	0.289	0.262	0.290	0.263	0.292	0.281	0.299	0.279	0.297	0.272	0.295	0.264	0.290	0.277	0.301	0.276	0.299
	720	0.344	0.344	0.340	0.340	0.346	0.348	0.343	0.345	0.357	0.349	0.357	0.348	0.349	0.345	0.342	0.340	0.358	0.353	0.354	0.349
	Avg	0.244	0.275	0.242	0.271	0.244	0.274	0.242	0.273	0.259	0.280	0.259	0.279	0.250	0.276	0.243	0.271	0.259	0.287	0.253	0.280
ECL	96	0.156	0.247	0.158	0.250	0.140	0.242	0.139	0.242	0.150	0.242	0.150	0.240	0.141	0.235	0.142	0.238	0.169	0.274	0.168	0.272
	192	0.170	0.260	0.172	0.263	0.157	0.256	0.159	0.259	0.166	0.257	0.165	0.253	0.158	0.250	0.155	0.246	0.188	0.291	0.185	0.288
	336	0.187	0.278	0.187	0.277	0.172	0.269	0.171	0.273	0.181	0.273	0.181	0.271	0.166	0.263	0.161	0.251	0.200	0.301	0.197	0.299
	720	0.228	0.312	0.227	0.310	0.211	0.307	0.210	0.307	0.214	0.302	0.216	0.302	0.199	0.295	0.195	0.291	0.273	0.355	0.212	0.305
	Avg	0.185	0.274	0.186	0.275	0.170	0.269	0.170	0.270	0.178	0.269	0.178	0.267	0.166	0.261	0.163	0.256	0.208	0.305	0.190	0.291
Traffic	96	0.463	0.293	0.473	0.307	0.447	0.270	0.436	0.278	0.397	0.271	0.393	0.266	0.397	0.272	0.393	0.261	0.589	0.315	0.591	0.314
	192	0.507	0.301	0.486	0.306	0.465	0.279	0.450	0.285	0.416	0.279	0.416	0.275	0.412	0.269	0.408	0.260	0.618	0.323	0.618	0.325
	336	0.526	0.308	0.496	0.312	0.483	0.289	0.480	0.482	0.432	0.287	0.431	0.281	0.435	0.286	0.432	0.280	0.649	0.352	0.633	0.344
	720	0.565	0.324	0.528	0.333	0.524	0.314	0.508	0.319	0.465	0.304	0.462	0.298	0.462	0.295	0.463	0.302	0.678	0.367	0.662	0.350
	Avg	0.515	0.307	0.496	0.314	0.480	0.288	0.468	0.341	0.428	0.285	0.425	0.280	0.427	0.281	0.424	0.276	0.634	0.339	0.626	0.333

E ALIGNMENT OF SCALING LAW

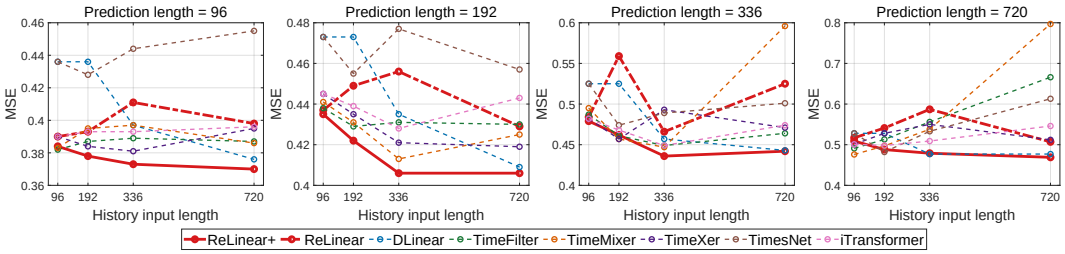


Figure 4: MSE versus historical input length in `ETTh1`. Each subplot represents a fixed prediction window, with input length on the x-axis and MSE on the y-axis.

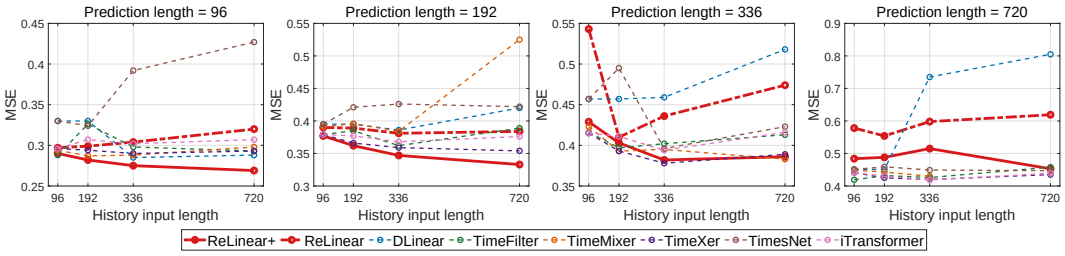


Figure 5: MSE versus historical input length in `ETTh2`. Each subplot represents a fixed prediction window, with input length on the x-axis and MSE on the y-axis.

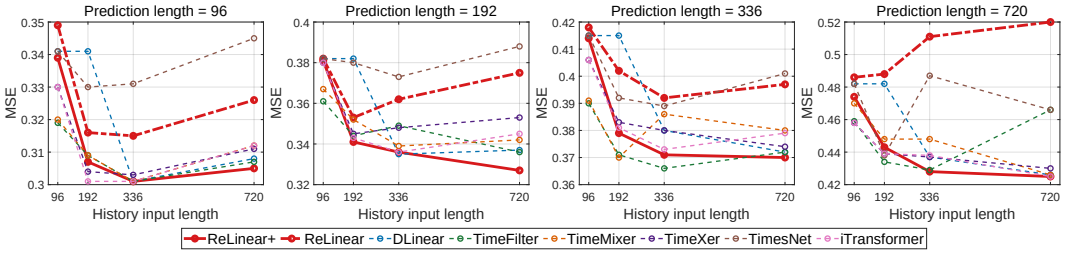


Figure 6: MSE versus historical input length in `ETTm1`. Each subplot represents a fixed prediction window, with input length on the x-axis and MSE on the y-axis.

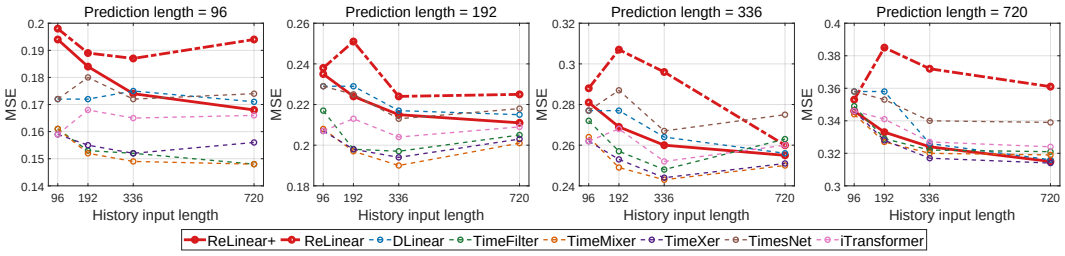


Figure 7: MSE versus historical input length in `weather`. Each subplot represents a fixed prediction window, with input length on the x-axis and MSE on the y-axis.

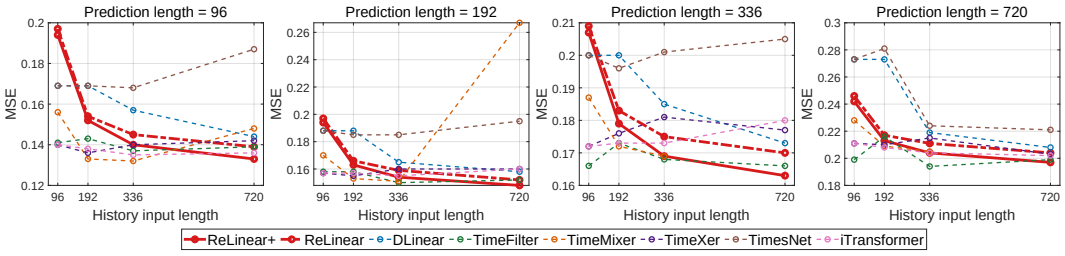


Figure 8: MSE versus historical input length in `electricity`. Each subplot represents a fixed prediction window, with input length on the x-axis and MSE on the y-axis.

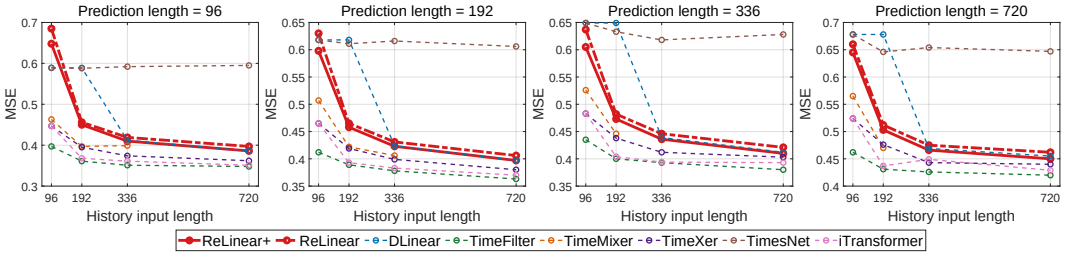


Figure 9: MSE versus historical input length in `traffic`. Each subplot represents a fixed prediction window, with input length on the x-axis and MSE on the y-axis.

F GENERATIVE AI DISCLOSURE

In this work, we used generative AI tools for the following tasks (all fall under the category of **recommended disclosure**):

- **Drafting paper text** (e.g., generating initial wording and rephrasing, followed by authors' manual revision and confirmation)
- **Summarizing literature** (extracting key points, followed by authors' verification against the original sources)
- **Finding related work and identifying research gaps** (using AI-assisted suggestions to guide search and synthesis, followed by authors' validation)
- **Improving readability and editing** (language polishing and structural/wording edits)
- **Formatting references** (formatting reference entries and bibliographic style; final citation details were checked by the authors)
- **Suggesting titles and keywords** (generating candidate options; final selection was made by the authors)

We did **not** use generative AI tools for the **required disclosure** tasks, including (but not limited to) generating synthetic datasets, developing theoretical models or conceptual frameworks, formulating/assisting with mathematical claims or proofs, proposing/refining hypotheses, designing or providing feedback on methodology/experiments, implementing methods, translation, qualitative/thematic data analysis, or interpreting results.