Abstain Mask Retain Core: Time Series Prediction by Adaptive Masking Loss with Representation Consistency

Anonymous Author(s)
Affiliation
Address
email

Abstract

Time series forecasting plays a pivotal role in critical domains such as energy management and financial markets. Although deep learning-based approaches (e.g., MLP, RNN, Transformer) have achieved remarkable progress, the prevailing "long-sequence information gain hypothesis" exhibits inherent limitations. Through systematic experimentation, this study reveals a counterintuitive phenomenon: appropriately truncating historical data can paradoxically enhance prediction accuracy, indicating that existing models learn substantial redundant features (e.g., noise or irrelevant fluctuations) during training, thereby compromising effective signal extraction. Building upon information bottleneck theory, we propose an innovative solution termed Adaptive Masking Loss with Representation Consistency (AMRC), which features two core components: 1) Dynamic masking loss, which adaptively identified highly discriminative temporal segments to guide gradient descent during model training; 2) Representation consistency constraint, which stabilized the mapping relationships among inputs, labels, and predictions. Experimental results demonstrate that AMRC effectively suppresses redundant feature learning while significantly improving model performance. This work not only challenges conventional assumptions in temporal modeling but also provides novel theoretical insights and methodological breakthroughs for developing efficient and robust forecasting models. We have made our code available at `https://anonymous.4open.science/r/AMRC/`.

1 Introduction

Time series forecasting, as a pivotal technology in critical domains such as energy management and financial markets, directly influences decision-making quality and economic efficiency [11, 18, 12, 19, 22]. Recent breakthroughs in deep learning have driven revolutionary advancements in time series prediction. Contemporary frameworks including Multilayer Perceptron (MLP)-based architectures [17, 30, 7, 27, 4, 28], Recurrent Neural Networks (RNNs) with their variants [13, 21, 9], and attention mechanism-based models exemplified by the Transformer [20, 33, 32, 16, 34, 2, 6], have achieved remarkable breakthroughs in modeling complex temporal patterns through the construction of elaborate hierarchical temporal dependencies.

Current mainstream forecasting models predominantly adhere to the "long-sequence information gain hypothesis," which posits that extending historical data length enhances the availability of temporal dependencies [31, 15]. However, through systematic experimental analysis, this study challenges this conventional assumption. As shown in Table 1, we observed a counterintuitive phenomenon across multiple benchmark datasets and diverse model architectures: appropriately truncating early segments of input sequences can significantly improve prediction accuracy. This finding reveals a critical issue in modern predictive models: during training, models inadvertently capture a substantial number of redundant features. These features not only fail to enhance performance but also interfere with the learning process, thereby limiting the models' potential to achieve optimal results.

Through systematic analysis, we have identified two typical manifestations of redundant features and their underlying mechanisms. First, input truncation optimization experiments (as shown in Figure 2b and Table 1) demonstrate that selectively masking partial historical data can significantly improve model prediction performance. This phenomenon reveals the current model's inefficient utilization of long historical windows. Second, representation similarity analysis (as illustrated in Figure 2a) shows that both the model's prediction results and intermediate embeddings exhibit an abnormally concentrated distribution, which significantly deviates from the natural dispersion characteristics of the input and label. Collectively, these observations indicate that existing models exhibit low efficiency when processing long historical windows, often encoding substantial noise or irrelevant variables rather than truly predictive signals.

Building upon information bottleneck theory [24, 25, 23, 10], this study proposes an innovative method called Adaptive Masking Loss with Representation Consistency (AMRC). The core methodology comprises: 1) An adaptive masking mechanism that dynamically identifies key segments with high discriminative power in sequential data and leverages these informative segments to guide the gradient optimization process (as illustrated in Fig 3) ; 2) A representation consistency constraint that establishes stable mapping relationships among the input feature space, label space, and predicted outputs, thereby effectively enhancing the model's generalization capability. Experimental results (as shown in Table 2) demonstrate that the AMRC method significantly reduces the complexity of the training solution space by suppressing the model's reliance on redundant features, fully exploits the performance potential of the model architecture, and consequently improves prediction accuracy.

The primary contributions of this study include:

- **Theoretical Insight:** Through rigorous experimental validation, We demonstrate that existing time series forecasting models are prone to learning redundant features, which in turn constrain their performance. Building on the theory of information bottlenecks, we construct a novel theoretical framework for time series modeling and propose an innovative optimization pathway, offering a new theoretical perspective for advancing the field of time series forecasting.
- **Methodological Innovation:** We propose an optimization framework Adaptive Masking Loss with Representation Consistency. By dynamically selecting discriminative temporal segments to guide gradient descent (as illustrated in Figure 1) while enforcing input-label-prediction consistency, our method effectively suppresses redundant feature learning. Extensive experiments demonstrate consistent performance gains across diverse benchmarks and architectures.

Our work advances the understanding of temporal pattern learning mechanisms while offering a practical pathway to enhance the efficiency and reliability of time series forecasting systems.

2 Analysis of Redundant Feature Learning

Given a multivariate time series $\mathbf{X} \in \mathbb{R}^{T\times D}$, where T is the number of timesteps and D is the number of variables, the objective of time series forecasting is to learn a mapping function f_θ that transforms historical observations $\mathbf{X}_{t-L:t} \in \mathbb{R}^{L\times D}$ (where L denotes the input length) into future values $\mathbf{X}_{t+1:t+H} \in \mathbb{R}^{H\times D}$ (where H represents the forecasting horizon).

Conventional time series forecasting models follow the long-sequence information gain hypothesis[3, 33, 5, 29], which holds that increasing the input length L improves forecasting accuracy. However, our experiments (Table 1) on multiple standard benchmarks reveal a counterintuitive result: truncating the input—such as masking the first k timesteps—often improves forecasting performance, which is measured by Mean Squared Error (MSE). We found that models tend to learn redundant features, which degrade model performance even after convergence. This finding is supported by two key observations:

2.1 Input Truncation Optimization

Based on the baseline model configuration (input length $L = 48$, forecasting horizon $H = 48$), we design an input truncation comparative experiment by applying a masking operator $\mathcal{M}_k(\cdot)$ to the

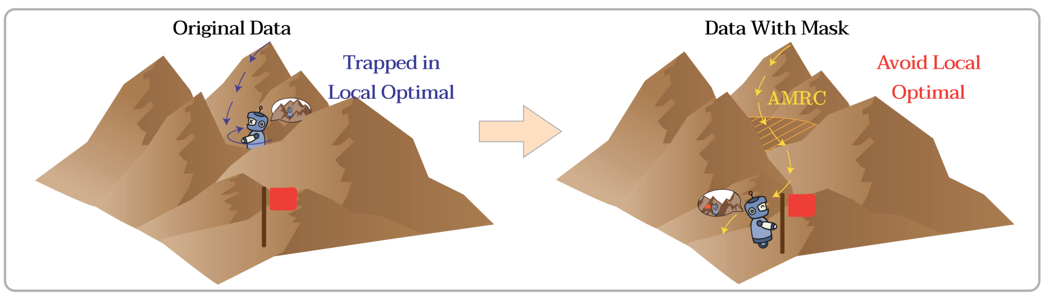


Figure 1: Illustration of the effect of AMRC method. Without regularization, the model tends to overfit redundant input features, leading to suboptimal convergence. By suppressing redundant input features, AMRC restructures the optimization landscape, promoting more efficient representation learning and facilitating better convergence.

input sequence. When we have an input sequence of length L at time step t, denoted as $\boldsymbol{X}_t^{(L)}$, the masking operator $\mathcal{M}_k(\cdot)$ is mathematically defined as:

$$\mathcal{M}_k(\boldsymbol{X}_t^{(L)}) = \begin{cases} 0 & \text{if } i \leq k \\ \boldsymbol{X}_t^{(L)} & \text{otherwise} \end{cases} \tag{1}$$

Here, $k \in \{1, \ldots, L\}$ denotes the masking step size.

To probe redundant features, we employ an Optimal Masking strategy: Given an input sequence of length L, we generate L masked variants $\{\mathcal{M}_k(\boldsymbol{X}_t^{(L)})\}_{k=1}^{L}$ (zero-padded to preserve dimensionality). For instance, $k = 5$ yields $L' = 43$ (first 5 positions zeroed). The optimal mask length k^* is selected as the configuration minimizing MSE, thereby defining the theoretical upper bound for redundancy elimination:

$$k^* = \underset{k \in \{1,2,\ldots,L\}}{\arg\min} \; \mathbb{E}\left[\left\| f_\theta\big(\mathcal{M}_k(\boldsymbol{X}_t^{(L)})\big) - \boldsymbol{Y}_t^{(H)} \right\|^2\right] \tag{2}$$

Table 1: Performance Gains via Optimal Masking Across Time Series Models. Ratio quantifies the percentage of training samples demonstrating prediction error reduction through Optimal Masking, calculated as *number of masked series/number of total series* $\times 100\%$.

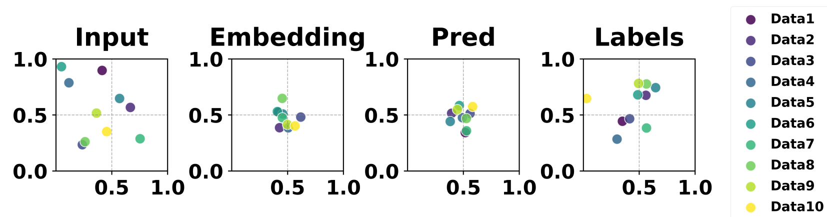
Model		ETTh1			ETTh2			Solar-Energy			Weather		
Metric		MSE	MSE*	Ratio	MSE	MSE*	Ratio	MSE	MSE*	Ratio	MSE	MSE*	Ratio
SOFTS	Train Set	0.278	**0.254**	56.54%	0.318	**0.259**	61.65%	0.182	**0.155**	11.80%	0.421	**0.400**	45.10%
	Test Set	0.408	**0.365**	64.24%	0.326	**0.303**	28.73%	0.293	**0.184**	41.58%	0.205	**0.185**	54.93%
iTransformer	Train Set	0.298	**0.270**	57.87%	0.315	**0.261**	64.19%	0.410	**0.281**	61.97%	0.436	**0.389**	62.98%
	Test Set	0.413	**0.289**	60.07%	0.329	**0.299**	32.16%	0.395	**0.271**	68.43%	0.209	**0.170**	80.26%
PatchTST	Train Set	0.343	**0.303**	65.57%	0.329	**0.269**	69.35%	0.366	**0.277**	35.89%	0.227	**0.180**	45.55%
	Test Set	0.424	**0.402**	65.51%	0.327	**0.298**	42.46%	0.374	**0.344**	51.66%	0.215	**0.180**	42.43%
TSMixer	Train Set	0.372	**0.342**	55.79%	0.544	**0.431**	73.96%	0.233	**0.195**	26.30%	0.363	**0.348**	37.57%
	Test Set	0.402	**0.372**	59.19%	0.324	**0.289**	42.13%	0.288	**0.250**	40.12%	0.222	**0.195**	70.88%
TimeMixer	Train Set	0.290	**0.262**	57.96%	0.309	**0.251**	59.36%	0.142	**0.112**	13.58%	0.403	**0.353**	63.93%
	Test Set	0.393	**0.366**	58.04%	0.318	**0.285**	44.52%	0.288	**0.253**	36.25%	0.197	**0.172**	66.13%

As demonstrated in Table 1, the experimental results confirm that masked models consistently achieve lower MSE, with more than 50% of samples exhibiting improved predictive performance (Ratio > 50%). Notably, the phenomenon of redundancy learning shows strong architecture-agnostic characteristics. On the Weather dataset, both iTransformer (a Transformer-based model) and TSMixer (an MLP-based model) demonstrate similar relative improvements: iTransformer achieves an MSE reduction from **0.209** to **0.170** (-18.7%), while TSMixer improves from **0.222** to **0.195** (-12.2%). These results indicate that the effectiveness of our masking strategy is not dependent on specific model architectures.

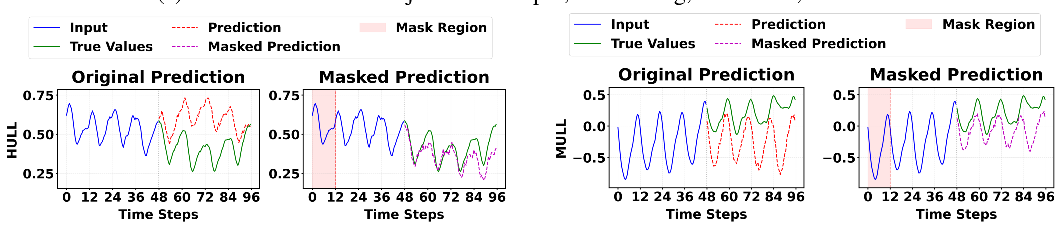
2.2 Representation Similarity Paradox

To further investigate the redundant feature learning phenomenon, we apply t-SNE to project the SOFTS model's high-dimensional representations of the input, embedding, prediction, and label onto a 2D plane (Fig. 2a), after normalizing all features to the $[0, 1]$ range.

As illustrated in Fig.2a, Normalized input ($\mathbf{Z}_{\text{in}} \in \mathbb{R}^L$) and output ($\mathbf{Z}_{\text{out}} \in \mathbb{R}^H$) embeddings show a clear contrast: inputs remain dispersed, while embeddings and preds cluster tightly despite large differences in their corresponding labels. This suggests that the model encodes redundant, task-irrelevant features that misrepresent semantic relationships and distort the input-output mapping.



(a) Normalized t-SNE Projections of Input, Embedding, Prediction, and Label



(b) Masked vs. Unmasked Prediction Performance

Figure 2: Embedding Distributions and Masking Effects of Our Method.

2.3 Information Bottleneck Constraints on Redundancy

According to the Information Bottleneck (IB) Theory [23], a neural network functions like a bottleneck that compresses input information during feature extraction. It discards irrelevant or noisy details and retains only the components most relevant to the overall task. For a time series forecasting model, let the input be denoted by X, the latent representation by Z, and the prediction target by Y. The model aims to learn a representation Z that maximally preserves information relevant to Y. This objective can be formally expressed as maximizing the mutual information between Z and Y:

$$I(Z, Y; \boldsymbol{\theta}) = \int dx\, dy\, p(z, y \mid \boldsymbol{\theta}) \log \frac{p(z, y \mid \boldsymbol{\theta})}{p(z \mid \boldsymbol{\theta}) p(y \mid \boldsymbol{\theta})}. \tag{3}$$

Due to inherent limitations in the data and model capacity, the amount of information that can be extracted and transmitted during training is bounded. Consequently, the representation capacity is subject to an upper information constraint I_c. Based on this, the objective of the time series prediction model can be equivalently formulated as the following constrained optimization problem:

$$\max_{\boldsymbol{\theta}} I(Z, Y; \boldsymbol{\theta}) \quad \text{s.t.} \quad I(X, Z; \boldsymbol{\theta}) \leq I_c. \tag{4}$$

This constrained optimization problem can be transformed into an unconstrained form using the method of Lagrange multipliers, leading to the maximization of the following objective[1]:

$$R_{\text{IB}}(\boldsymbol{\theta}) = I(Z; Y; \boldsymbol{\theta}) - \beta I(Z; X; \boldsymbol{\theta}). \tag{5}$$

There are two implementation paths under this objective: one is to maximize the mutual information $I(Z; Y)$ between Z and Y; the other is to minimize the mutual information $I(Z; X)$ between Z and X. Most current sequential prediction models focus on improving $I(Z; Y)$ through iterative training, but have not explicitly optimized performance by penalizing redundant features via minimizing $I(Z; X)$. Therefore, we propose an adaptive loss function that aims to minimize the mutual information between X and Z, offering a novel optimization path for improving the performance of sequential prediction models.

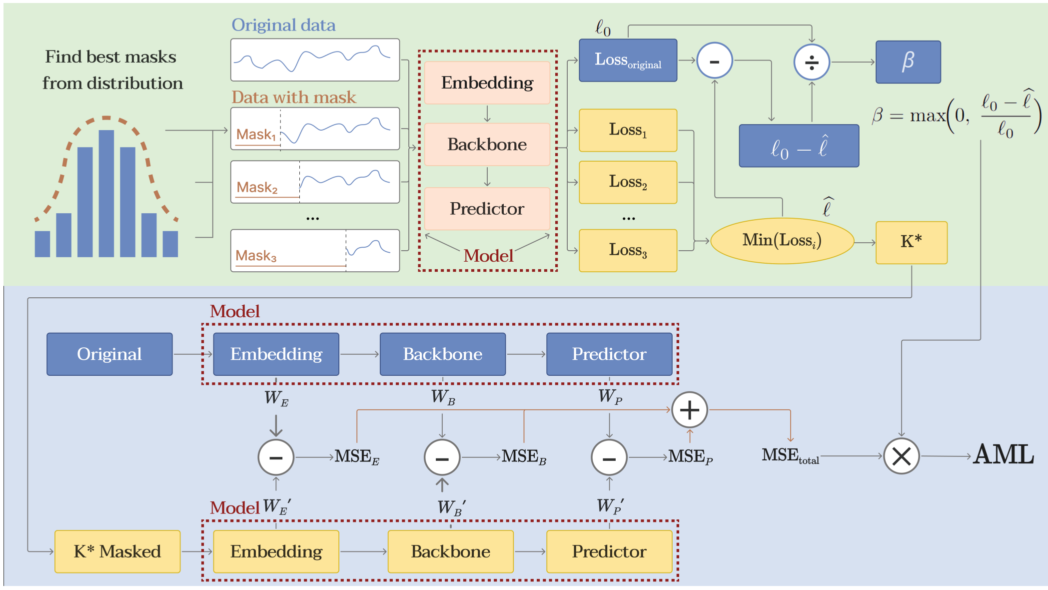


Figure 3: Overview of the Adaptive Masking Loss (AML) framework. The upper half illustrates how the optimal mask length K^* is selected by evaluating prediction losses over sampled masks. A weighting coefficient β is computed based on the gain over the unmasked loss. The lower half shows the AML loss, calculated as the sum of representation differences between the original input and the K^* masked input across embedding, backbone, and predictor layers.

3 Proposed Method

3.1 Adaptive Masking Loss (AML)

As discussed in Section 2.1, applying ideal masking to input data reduces the information $I(X)$ while improving prediction accuracy. This indicates that the representation Z_{k^*}, generated by encoder p_θ from masked features X_{t,k^*}, contains less redundancy and better approximates the minimal sufficient statistics (i.e., with smaller $I(X, Z_{k^*}; \theta)$). Based on this insight, we propose the **Adaptive Masking Loss (AML)** to explicitly reduce mutual information $I(X, Z; \theta)$ by guiding the encoder's output representation Z toward Z_{k^*}, thereby suppressing redundant feature learning and unleashing model potential. The overall framework of AML is illustrated in Figure 3.

3.1.1 Implementation

The exhaustive search for optimal mask k^* by enumerating all possible mask lengths $k \in \{1, ..., L\}$ results in prohibitive $O(L)$ time complexity for long sequences. We therefore adopt an efficient stochastic approximation strategy:

1. **Random Mask Generation**: Independently sample m mask indices $\{k_s\}_{s=1}^{m}$ from uniform distribution $d(k) = \text{Uniform}\{1, ..., L\}$, each generating a masked variant:
$$\widetilde{X}_{t,s}^{(L)} = \mathcal{M}_{k_s}(X_t^{(L)}) \tag{6}$$
2. **Loss Evaluation**: Compute prediction losses for both masked and original data:
$$\ell_s = \mathcal{L}(f_\theta(\widetilde{X}_{t,s}^{(L)}), Y_t^{(H)}) \tag{7}$$
$$\ell = \mathcal{L}(f_\theta(X_t^{(L)}), Y_t^{(H)}) \tag{8}$$
3. **Optimal Representation Selection**: If $\exists \ell_s < \ell$, the corresponding representation $\widetilde{Z}_s = p_\theta(\widetilde{X}_{t,s}^{(L)})$ satisfies $I(X_t^{(L)}, \widetilde{Z}_s) < I(X_t^{(L)}, Z)$, where $Z = p_\theta(X_t^{(L)})$ is the original representation. The optimal mask variant is selected by:
$$s^* = \arg\max_s (\ell - \ell_s) \tag{9}$$

3.1.2 Loss Formulation

To promote compact and informative representations, AML minimizes the distance between the original representation Z and the optimal masked variant $\widetilde{Z}_{s^*}$:

$$\mathcal{L}_{\mathrm{AML}} = \beta \cdot \frac{1}{D_1 \times D_2} \|Z - \widetilde{Z}_{s^*}\|^2 \tag{10}$$

where the adaptive weight $\beta = \max(0, (\ell - \ell_{s^*})/\ell)$ dynamically scales the optimization intensity, ensuring stronger influence from mask variants with greater loss reduction.

3.2 Embedding Similarity Penalty (ESP)

Time series forecasting models often encounter two issues: semantic inconsistency, where semantically similar inputs lead to substantially different predictions, and representation collapse, where dissimilar inputs result in nearly identical outputs. Both problems reduce the robustness and generalization ability of the model. To address these issues, we introduce a regularization strategy that compares, for each pair of samples within a mini-batch, the geometry of the embedding space with that of the output space.

Pairwise distances. For a batch $\mathcal{B} = \{(X_i, Y_i)\}_{i=1}^{n}$ we denote by $Z_i = f_{\mathrm{enc}}(X_i) \in \mathbb{R}^{L \times D}$ the encoder output and keep the ground-truth $Y_i \in \mathbb{R}^{P \times D}$. The (normalised) squared Frobenius distances are

$$\Delta_{ij}^{E} = \frac{1}{L \times D} \|Z_i - Z_j\|_F^2, \qquad \Delta_{ij}^{O} = \frac{1}{P \times D} \|Y_i - Y_j\|_F^2, \quad 1 \le i, j \le n. \tag{11}$$

Consistency penalty. Ideally Δ_{ij}^{E} and Δ_{ij}^{O} should match: semantically similar inputs ($\Delta_{ij}^{E} \approx 0$) ought to produce similar outputs ($\Delta_{ij}^{O} \approx 0$), and vice versa. Deviation is quantified element-wise through

$$P_{ij} = \mathrm{ReLU}\big(\Delta_{ij}^{E} - \Delta_{ij}^{O}\big) + \mathrm{ReLU}\big(\Delta_{ij}^{O} - \Delta_{ij}^{E}\big) = |\,\Delta_{ij}^{E} - \Delta_{ij}^{O}|_{+}, \tag{12}$$

where $\mathrm{ReLU}(x) = \max(0, x)$ and $|\cdot|_{+}$ denotes the non-negative part. The **Embedding-Similarity Penalty** then reads

$$\mathcal{L}_{\mathrm{ESP}} = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} P_{ij}. \tag{13}$$

Equation (13) back-propagates smooth, unbiased gradients that jointly reshape the encoder and the predictor so that input and output manifolds remain geometrically aligned. The detailed implementation of the Embedding Similarity Penalty (ESP) is provided as pseudocode in Appendix D.

3.3 Overall Training Objective

Section 3.1 introduced the Adaptive Masking Loss $\mathcal{L}_{\mathrm{AML}}$ that discourages the learning of redundant temporal prefixes, while Section 3.2 proposed the Embedding-Similarity Penalty $\mathcal{L}_{\mathrm{ESP}}$ to enforce semantic–behavioural consistency. Combined with the standard prediction loss $\mathcal{L}_{\mathrm{pred}}$ (*e.g.*, MSE between the forecast $\hat{Y}$ and the target Y), our final objective is

$$\mathcal{L}_{\text{total}} \;=\; \mathcal{L}_{\mathrm{pred}} \;+\; \lambda_{\mathrm{AML}}\,\mathcal{L}_{\mathrm{AML}} \;+\; \lambda_{\mathrm{ESP}}\,\mathcal{L}_{\mathrm{ESP}}, \tag{14}$$

where $\lambda_{\mathrm{AML}}, \lambda_{\mathrm{ESP}} > 0$ control the strength of each auxiliary term. Minimizing (14) jointly (i) identifies the informative prefix for every sequence, (ii) preserves the intrinsic topology of the data, and (iii) improves predictive accuracy and interpretability without adding inference-time overhead.

4 Experiment

4.1 Experiment Setup

Datasets. We evaluate our proposed method using seven widely recognized benchmark datasets for multivariate time series forecasting: **ETTh1**, **ETTh2**, **ETTm1**, **ETTm2**, **Solar-Energy**, **Electricity**, and **Weather**. These datasets encompass a variety of application scenarios with different temporal resolutions, seasonality patterns, and dynamic structures. Detailed descriptions of each dataset, including their specific characteristics and collection periods, are provided in the appendix F.

Task formulation. In our experimental setup, the forecasting task is formulated as a sequence-to-sequence regression problem, applicable to multivariate time series. Each model is trained to predict a future sequence $\boldsymbol{Y}_t^{(H)} \in \mathbb{R}^{H \times D}$ from a fixed-length historical input sequence $\boldsymbol{X}_t^{(48)} \in \mathbb{R}^{48 \times D}$, where H denotes the prediction length and D is the number of variables. We adopt multiple prediction horizons $H \in \{48, 72, 96, 120, 144, 168, 192\}$.

Baselines. Our method is compared against five diverse baseline models: **SOFTS** [8], **iTransformer** [14], **PatchTST** [16], **TSMixer** [7], and **TimeMixer** [26]. These baselines are implemented using their official codebases and recommended hyperparameters to ensure a fair comparison under consistent experimental conditions.

Implementation details. All models are implemented in PyTorch and trained on a single NVIDIA A100 80GB GPU. To ensure a fair comparison and allow both baseline models and those augmented with our proposed modules to fully exploit their capacity, we train each model for up to 100 epochs using the Adam optimizer with an initial learning rate of 1×10^{-4}, a cosine annealing scheduler, and a batch size of 32. Early stopping is applied based on validation loss with a patience of 20 epochs. The best-performing checkpoint on the validation set is selected for final evaluation on the test set.

Hyperparameter selection. For the AML, the input sequence prefix length is configured as $L = 48$, with the mask sampling cardinality parameterized as $m = 12$. We fix both λ_{AML} and λ_{ESP} to 1 for all experiments. These settings follow standard benchmark configurations commonly used in time series forecasting.

4.2 Forecasting Results

We present the forecasting performance of our method—Adaptive Masking Loss with Representation Consistency (AMRC)—in comparison with five representative baseline models across seven widely used time series benchmark datasets. Table 2 reports the Mean Squared Error (MSE) and Mean Absolute Error (MAE) for each model, both with and without the incorporation of AMRC.

Table 2: Performance Comparison of Time Series Forecasting Models With and Without AMRC. In the experimental results, we highlighted in bold the parts where the AMRC model improved by more than 0.05 in MSE and MAE metrics compared to the baseline model. The detailed hyperparameter configurations for each model can be found in Appendix C. Full results are listed in Appendix E

Model		ETTh1		ETTh2		ETTm1		ETTm2		Solar-Energy		Electricity		Weather	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
SOFTS	Original	0.408	0.414	0.326	0.359	0.484	0.434	0.210	0.285	0.293	0.314	0.169	0.255	0.205	0.234
	AMRC	**0.389**	**0.393**	**0.311**	0.362	**0.475**	**0.423**	**0.198**	**0.265**	0.290	0.309	**0.162**	**0.244**	**0.196**	**0.220**
iTransformer	Original	0.413	0.415	0.329	0.362	0.517	0.448	0.213	0.290	0.395	0.352	0.176	0.260	0.209	0.237
	AMRC	**0.402**	**0.399**	0.324	**0.356**	**0.502**	**0.447**	0.211	**0.280**	0.392	**0.342**	**0.163**	**0.239**	**0.201**	**0.221**
TimeMixer	Original	0.393	0.408	0.318	0.355	0.466	0.429	0.209	0.285	0.288	0.317	0.194	0.279	0.197	0.237
	AMRC	0.388	**0.401**	0.316	**0.339**	**0.447**	**0.405**	0.204	**0.269**	0.284	0.317	**0.188**	0.277	**0.186**	**0.228**
PatchTST	Original	0.424	0.424	0.327	0.358	0.461	0.422	0.211	0.287	0.374	0.382	0.211	0.283	0.215	0.280
	AMRC	**0.411**	**0.415**	**0.319**	0.356	0.456	**0.413**	**0.196**	**0.271**	**0.361**	0.376	0.207	0.285	0.210	**0.264**
TSMixer	Original	0.402	0.412	0.324	0.357	0.440	0.413	0.201	0.279	0.288	0.314	0.172	0.258	0.222	0.288
	AMRC	**0.386**	**0.397**	0.319	**0.340**	**0.432**	0.412	0.196	**0.257**	**0.280**	0.313	0.169	**0.247**	**0.212**	**0.281**

Consistent Performance Gains. Across all models and datasets, our method consistently yields performance improvements. For example, the MSE of the SOFTS model decreases from 0.408 to 0.389 on the ETTh1 dataset. Similar trends are observed in iTransformer, where the MSE on Electricity drops from 0.176 to 0.163. The enhancements demonstrate that AMRC effectively mitigates redundant or noisy temporal segments, thereby improving prediction stability and accuracy.

Architecture-Agnostic Effectiveness. AMRC delivers significant performance gains not only on Transformer-based architectures such as iTransformer and PatchTST, but also on MLP-based models including TimeMixer, SOFTS, and TSMixer. For instance, on the ETTm2 dataset, the MSE of PatchTST model decreases from 0.211 to 0.196 (a reduction of approximately 7.11%), while the MSE of SOFTS model drops from 0.210 to 0.198 (approximately 5.71% reduction). These results demonstrate the strong architecture-agnostic generalization ability of AMRC, highlighting its broad applicability across a wide range of time series forecasting models.

Table 3: Ablation Study Results on Different Model Components

Model		ETTh1		ETTh2		Solar-Energy		Weather	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
SOFTS	AML only	0.401	0.405	0.322	0.358	0.297	0.309	0.192	0.228
	ESP only	0.393	0.398	0.318	0.351	0.295	0.318	0.208	0.241
	AMRC	**0.389**	**0.393**	**0.311**	**0.362**	**0.290**	**0.309**	**0.196**	**0.220**
iTransformer	AML only	0.410	0.413	0.328	0.363	0.398	0.347	0.205	0.230
	ESP only	0.407	0.408	0.326	0.359	0.402	0.351	0.210	0.248
	AMRC	**0.402**	**0.399**	**0.324**	**0.356**	**0.392**	**0.342**	**0.201**	**0.221**
TimeMixer	AML only	0.395	0.412	0.319	0.351	0.287	0.319	0.189	0.232
	ESP only	0.391	0.406	0.317	0.347	0.293	0.325	0.202	0.248
	AMRC	**0.388**	**0.401**	**0.316**	**0.339**	**0.284**	**0.317**	**0.186**	**0.228**
PatchTST	AML only	0.419	0.420	0.325	0.361	0.369	0.379	0.214	0.274
	ESP only	0.417	0.418	0.323	0.357	0.375	0.384	0.217	0.281
	AMRC	**0.411**	**0.415**	**0.319**	**0.356**	**0.361**	**0.376**	**0.210**	**0.264**
TSMixer	AML only	0.396	0.404	0.324	0.356	0.285	0.317	0.216	0.283
	ESP only	0.390	0.399	0.322	0.352	0.291	0.323	0.224	0.292
	AMRC	**0.386**	**0.397**	**0.319**	**0.340**	**0.280**	**0.313**	**0.212**	**0.281**

Generalization on Low-Channel Datasets. On datasets with fewer input channels (ETTh1, ETTh2, ETTm1, ETTm2), AMRC effectively enhances model performance. For instance, on ETTm1, the MSE of iTransformer decreases from 0.517 to 0.502, and that of TSMixer drops from 0.440 to 0.432. These results demonstrate AMRC's ability to mitigate overfitting and improve prediction accuracy in low-dimensional time series forecasting tasks.

Robustness on High-Channel Datasets. For high-dimensional datasets such as Weather (21 channels) and Solar-Energy (137 channels) see in Appendix F, AMRC consistently improves robustness by reducing the impact of signal noise and inter-channel redundancy. On the Weather dataset, TimeMixer's MSE decreases from 0.197 to 0.186 and MAE from 0.237 to 0.228, while iTransformer sees an MAE drop from 0.237 to 0.221. On Solar-Energy, PatchTST's MSE drops from 0.374 to 0.361, and SOFTS sees a slight MAE reduction from 0.314 to 0.309. These enhancements highlight AMRC's effectiveness in managing complexity in multivariate time series with high channel counts.

Generalizable Training Framework. The consistent performance improvements observed across all models validate the strong scalability and integrability of AMRC. As a constraint-based optimization strategy, AMRC does not rely on any specific model architecture, making it highly generalizable. It serves as a versatile training framework for enhancing both the efficiency and accuracy of time series forecasting models.

4.3 Ablation Study

Setup. We evaluate ablation variants on four diverse datasets: ETTh1 and ETTh2, representing hourly electricity load with varying degrees of seasonality; Solar-Energy, which exhibits weather-driven variability and periodicity; and Weather, a multivariate meteorological dataset with complex inter-variable dependencies. We adopt a fixed input horizon following standard benchmarks.

Evaluation protocol. For each dataset, we apply the ablation study to five baseline models SOFTS, iTransformer, TimeMixer, PatchTST, and TSMixer under four configurations:1) baseline + AML, 2) baseline + ESP, and 3) baseline + both AML and ESP. This design allows us to assess the standalone effectiveness of each module as well as their combined synergy.

Findings. We evaluate the individual and joint effects of the AML and ESP components using five representative forecasting architectures across four datasets. As shown in Table 3, both components contribute measurable performance gains in isolation, while their combination AMRC consistently leads to the best forecasting accuracy in terms of MSE and MAE. AML provides stronger improvements across most settings, supporting its role in suppressing redundant prefixes during training. ESP, while often delivering smaller standalone gains, remains beneficial by promoting geometric alignment between embedding and output spaces. Together, these findings demonstrate that each component addresses a distinct source of generalization error.

Component impact across architectures. The benefits of AML and ESP are consistently observed across all backbone models, regardless of architectural differences. For instance, models with strong expressiveness, such as iTransformer and TimeMixer, benefit significantly from AML, achieving notable MSE reductions on datasets like Weather and ETTh2. Even architectures without attention mechanisms, such as SOFTS and TSMixer, exhibit consistent gains, highlighting the broad applicability of adaptive prefix masking. In contrast, the improvements from ESP are often more dataset-dependent, being particularly effective on high-dimensional multivariate inputs where representation alignment plays a critical role. For example, ESP yields non-trivial reductions in MAE on Weather, where multiple variables evolve under shared dynamics. Notably, we observe relatively smaller improvements on the Solar-Energy dataset for transformer-based models such as PatchTST and iTransformer, which may be attributed to their reliance on longer input sequences for stable attention computation.

Complementarity and synergy. The AMRC configuration, which jointly applies AML and ESP, consistently outperforms its ablated variants across all benchmarks. The performance improvement from combining both components generally exceeds the stronger of the two individual effects, indicating synergistic interaction. This complementarity can be attributed to their distinct operational scopes: AML operates on the input level by learning to suppress non-informative temporal segments, while ESP regularizes the latent space to align representations across semantically related inputs. As a result, AMRC improves both the quality of features learned from the data and the consistency of their usage in prediction. The robust gains observed across datasets and architectures suggest that jointly addressing input redundancy and representation inconsistency is critical for improving generalization in time series forecasting.

Table 4: AMRC Effectiveness Across Datasets and Models. Ratio is the percentage of training samples with reduced MSE under prefix masking. Ratio* is the same metric after training with AMRC, reflecting improved robustness. Results are from the ablation setting with input length set to 48. Detailed results are provided in Appendix E.

Model	ETTh1		ETTh2		Solar-Energy		Weather	
Metric	Ratio	Ratio*	Ratio	Ratio*	Ratio	Ratio*	Ratio	Ratio*
SOFTS	64%	**57.33%**	28.72%	**20.28%**	41.58%	**33.49%**	54.93%	**47.12%**
iTransformer	60.07%	**49.95%**	32.16%	**23.28%**	68.43%	**63.21%**	80.26%	**70.29%**
TimeMixer	58.04%	**46.29%**	44.52%	**34.17%**	36.25%	**27.90%**	66.13%	**52.28%**
PatchTST	65.51%	**51.63%**	42.46%	**26.19%**	51.66%	**47.64%**	42.43%	**30.78%**
TSMixer	59.19%	**46.62%**	42.13%	**27.98%**	40.12%	**28.36%**	70.88%	**58.23%**

Effectiveness of AMRC in Reducing Redundant Features We evaluate the model's robustness to redundant input by computing the proportion of training samples with improved MSE under prefix masking Ratio and compare it to the value after applying AMRC Ratio*. As shown in Table 4, AMRC consistently improves or maintains this ratio, indicating its effectiveness in suppressing the impact of redundant temporal information.

5 Conclusion

This study pioneers the investigation into the negative effects of redundant feature learning in time series forecasting and introduces AMRC, a plug-and-play solution that suppresses such learning without requiring architectural modifications. Unlike prior work focused on enhancing predictive features, AMRC improves accuracy by reducing reliance on redundant features while maintaining model flexibility. Its key advantages include: 1) seamless integration with existing models, 2) effective suppression of feature redundancy, and 3) strong generalization performance across benchmark tests. By addressing the long-overlooked issue of redundant learning, this research provides a novel and practical methodology for optimizing forecasting models.

References

[1] Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck, 2019. URL `https://arxiv.org/abs/1612.00410`.

[2] Peng Chen, Yingying Zhang, Yunyao Cheng, Yang Shu, Yihang Wang, Qingsong Wen, Bin Yang, and Chenjuan Guo. Pathformer: Multi-scale transformers with adaptive pathways for time series forecasting. *arXiv preprint arXiv:2402.05956*, 2024.

[3] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019.

[4] Abhimanyu Das, Weihao Kong, Andrew Leach, Shaan Mathur, Rajat Sen, and Rose Yu. Long-term forecasting with tide: Time-series dense encoder, 2024. URL `https://arxiv.org/abs/2304.08424`.

[5] Jiayu Ding, Shuming Ma, Li Dong, Xingxing Zhang, Shaohan Huang, Wenhui Wang, Nanning Zheng, and Furu Wei. Longnet: Scaling transformers to 1,000,000,000 tokens. *arXiv preprint arXiv:2307.02486*, 2023.

[6] Alexandre Drouin, Étienne Marcotte, and Nicolas Chapados. Tactis: Transformer-attentional copulas for time series. In *International Conference on Machine Learning*, pages 5447–5493. PMLR, 2022.

[7] Vijay Ekambaram, Arindam Jati, Nam Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. Tsmixer: Lightweight mlp-mixer model for multivariate time series forecasting. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, pages 459–469, 2023.

[8] Lu Han, Xu-Yang Chen, Han-Jia Ye, and De-Chuan Zhan. Softs: Efficient multivariate time series forecasting with series-core fusion. *arXiv preprint arXiv:2404.14197*, 2024.

[9] Hansika Hewamalage, Christoph Bergmeir, and Kasun Bandara. Recurrent neural networks for time series forecasting: Current status and future directions. *International Journal of Forecasting*, 37(1):388–427, 2021.

[10] Shizhe Hu, Zhengzheng Lou, Xiaoqiang Yan, and Yangdong Ye. A survey on information bottleneck. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[11] Nataliia Kashpruk, Cezary Piskor-Ignatowicz, and Jerzy Baranowski. Time series prediction in industry 4.0: a comprehensive review and prospects for future advancements. *Applied Sciences*, 13(22):12374, 2023.

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

[13] Shengsheng Lin, Weiwei Lin, Wentai Wu, Feiyu Zhao, Ruichao Mo, and Haotong Zhang. Segrnn: Segment recurrent neural network for long-term time series forecasting. *arXiv preprint arXiv:2308.11200*, 2023.

[14] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*, 2023.

[15] Chao Ma, Yikai Hou, Xiang Li, Yinggang Sun, and Haining Yu. Long input sequence network for long time series forecasting. *arXiv preprint arXiv:2407.15869*, 2024.

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

[17] Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv:1905.10437*, 2019.

[18] Asiye K Ozcanli, Fatma Yaprakdal, and Mustafa Baysal. Deep learning methods and applications for electrical power systems: A comprehensive review. *International Journal of Energy Research*, 44(9):7136–7157, 2020.

[19] Fotios Petropoulos, Daniele Apiletti, Vassilios Assimakopoulos, Mohamed Zied Babai, Devon K Barrow, Souhaib Ben Taieb, Christoph Bergmeir, Ricardo J Bessa, Jakub Bijak, John E Boylan, et al. Forecasting: theory and practice. *International Journal of forecasting*, 38(3):705–871, 2022.

[20] Yankun Ren, Longfei Li, Xinxing Yang, and Jun Zhou. Autotransformer: Automatic transformer architecture design for time series classification. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 143–155. Springer, 2022.

[21] Koushik Roy, Abtahi Ishmam, and Kazi Abu Taher. Demand forecasting in smart grid using long short-term memory. In *2021 International Conference on Automation, Control and Mechatronics for Industry 4.0 (ACMI)*, pages 1–5. IEEE, 2021.

[22] Zezhi Shao, Fei Wang, Yongjun Xu, Wei Wei, Chengqing Yu, Zhao Zhang, Di Yao, Tao Sun, Guangyin Jin, Xin Cao, et al. Exploring progress in multivariate time series forecasting: Comprehensive benchmarking and heterogeneity analysis. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[23] Noam Slonim and Naftali Tishby. Agglomerative information bottleneck. *Advances in neural information processing systems*, 12, 1999.

[24] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 ieee information theory workshop (itw)*, pages 1–5. Ieee, 2015.

[25] Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method, 2000. URL `https://arxiv.org/abs/physics/0004057`.

[26] Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y Zhang, and Jun Zhou. Timemixer: Decomposable multiscale mixing for time series forecasting. *arXiv preprint arXiv:2405.14616*, 2024.

[27] Zhijian Xu, Ailing Zeng, and Qiang Xu. Fits: Modeling time series with $10k$ parameters. *arXiv preprint arXiv:2307.03756*, 2023.

[28] Kun Yi, Qi Zhang, Wei Fan, Shoujin Wang, Pengyang Wang, Hui He, Ning An, Defu Lian, Longbing Cao, and Zhendong Niu. Frequency-domain mlps are more effective learners in time series forecasting. *Advances in Neural Information Processing Systems*, 36:76656–76679, 2023.

[29] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33: 17283–17297, 2020.

[30] 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, pages 11121–11128, 2023.

[31] 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, pages 11121–11128, 2023.

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

[33] 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, pages 11106–11115, 2021.

[34] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*, pages 27268–27286. PMLR, 2022.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims are clearly written in the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discussed the limitation of our method in Appendix A.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: All the theories and hypotheses we proposed are supported by experimental and mathematical derivations.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide detailed descriptions of the hyperparameters in the paper and appendices, along with an anonymous link to the experimental demo in the abstract.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We have included a link to an anonymous demo of our experiments in the abstract.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide the experimental setup details in both the main text and appendices.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The margin of error is reported in the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide sufficient computational resource details for each experiment in both the main text and appendices.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our methodology and implementation fully adhere to the ethical code standards set forth by NeurIPS.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: We have discussed the broader impact of time series forecasting in both abstract and introduction.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [No]

    Justification: This paper does not have this risk.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We included it in implementation details and appendix.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: N/A.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: N/A.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: N/A.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [No]

    Justification: We did not use any large language models (LLMs) in this work.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.