MULTI-MASK ONLINE ENSEMBLE TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Time series forecasting is increasingly required to operate on non-stationary data streams, where models must adapt online as temporal patterns evolve. Although Transformers have achieved strong performance in offline forecasting by relying on flexible all-to-all attention, such weak inductive bias can be less reliable in online settings. We identify three core mismatches: Over-extended receptive fields that capture spurious correlations. Single static architectures that fail to decouple heterogeneous temporal patterns. Slow adaptation to abrupt concept drift. To address these challenges, we propose MiMe, a Multi-Mask online ensemble framework for time series forecasting. MiMe introduces Pattern-induced Masking to impose diverse temporal inductive biases on self-attention, thereby constraining receptive fields for heterogeneous temporal patterns. It further employs Shared Backbone Training to integrate multiple masked forecasters within a single Transformer backbone, enabling parameter-efficient decoupling of heterogeneous patterns. Finally, online ensembling dynamically reweights masked forecasters through online convex optimization, enabling rapid strategy-level adaptation to concept drift with an $O(\log T)$ regret guarantee. Extensive experiments demonstrate MiMe's strong forecasting performance and robustness against state-of-the-art baselines. Code is available at https://github.com/anonymous7538/AnonProj.

1 INTRODUCTION

Time series forecasting is fundamental to intelligent decision-making systems and has been driven by high-value scenarios: from stock price fluctuation warnings and financial risk management Arsenault et al. (2025); Dong et al. (2024), to sequential recommendation Wang et al. (2025); Zhang et al. (2025), and then to online resource estimation in cloud platforms Chen et al. (2023a); Luo et al. (2024a); Gao et al. (2025); Ye et al. (2025); Gao et al. (2026), enabling auto-scaling of containers Wang et al. (2023); Luo et al. (2024b); Yu et al. (2025). Despite their diverse application domains, they share a common objective: to achieve highly accurate future predictions in dynamic environments.

Recently, Transformer architectures Zhou et al. (2021); Wu et al. (2021a); Nie et al. (2023) based on self-attention Vaswani et al. (2017) have become dominant backbones for time series modeling. The key advantage of Transformers is the significant relaxation of structural assumptions about data, known as inductive biases Wu et al. (2021b). Unlike the locality assumptions of Convolutional Neural Networks Wu et al. (2023) or the time-invariance constraints of Recurrent Neural Network Mienye et al. (2024), Transformer's all-to-all self-attention imposes minimal restrictions on the dependency range, enabling the flexible capture of multi-scale dependencies. In offline settings where the complete historical data is accessible, this minimal inductive bias facilitates substantial performance gains through global correlation modeling. A detailed discussion of related work is in Appendix. B

However, online forecasting operates on continuous streams whose distributions evolve over time, requiring models to adapt to distribution shifts, i.e., concept drift Chen et al. (2023a); Zhang et al. (2023). More importantly, concept drift is often accompanied by changes in the dominant temporal patterns. As illustrated in Fig. 1, streaming time series may switch among long-range dependencies, periodicity, contextual-dependencies, directional constraints, and short-range dependencies, each favoring a different **receptive field**, i.e., a different range of historical observations that should influence the forecast. Therefore, an effective online forecaster should not rely on a fixed receptive field, but adapt its information integration strategy as temporal patterns evolve.

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.

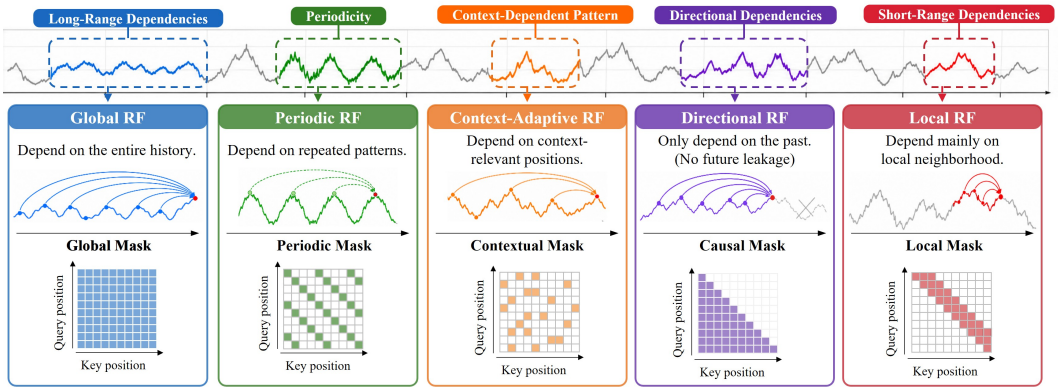


Figure 1: Temporal pattern switching in online time series and the corresponding receptive-field (RF) induced by Transformer relation masks. Different temporal patterns favor different receptive fields, motivating complementary masks for online adaptation.

In Transformers, relation masks provide a natural mechanism for constraining receptive fields by controlling pairwise connectivity in self-attention. However, conventional Transformer fails to switch among multiple receptive-field biases under evolving data streams. We further identify Transformer's limitations in online forecasting at three levels: the attention operator, model architecture, and online adaptation paradigm: **Over-extended receptive field**: unconstrained all-to-all attention may capture noise and spurious correlations, diluting informative temporal signals. **Static architecture vs. heterogeneous features**: Transformer uses a single parameter set to fit diverse temporal features, failing to decouple and adapt to evolving temporal patterns. **Online adaptation bottleneck**: the conventional "offline warm-up + online update" paradigm relies mainly on gradient-based parameter updates that respond slowly to concept drift.

To this end, we propose MiMe, an online ensemble forecasting framework based on multi-masked self-attention, containing three key components: **Pattern-Induced Masking** introduces several attention masks correspond to diverse temporal patterns, reintroducing inductive biases to constrain receptive fields and suppress spurious interferences. **Shared Backbone Training** integrates multiple masked forecasters within a shared Transformer backbone for parallel of heterogeneous temporal patterns extraction with extremely low parameter increments. **Online Ensembling** dynamically reweights masked forecasters through online convex optimization with a theoretical regret guarantee, allowing MiMe to rapidly adapt to concept drift. The contributions are summarized as follows:

- We propose an online ensemble framework that addresses the weak inductive bias of online Transformers by complementing parameter-level updates with strategy-level fast switching.
- We design pattern-induced masking to reintroduce structured inductive biases for heterogeneous temporal streams, integrating multiple masked forecasters into a single Transformer backbone via parameter-efficient shared training.
- We provide $O(\log T)$ regret bound for our online ensemble method. Experiments on 7 benchmarks demonstrate that MiMe consistently achieves state-of-the-art performance.

2 Problem Formulation and Preliminaries

We formalize online forecasting problem and weighted online ensembling, together with the regret criterion, then introduce the two-phase learning protocol and the masked Transformer used in MiMe.

2.1 Online Time Series Forecasting

Definition 1 (Online Data Stream)**.** An online time series is a sequence of observations arriving at successive time steps. At time step t, the observed stream is represented as $\{x_1, x_2, \cdots, x_{t-1}, x_t\}$.

Definition 2 (Multi-Step Time Series Forecasting, MSTF)**.** Given an online data stream within T time steps (which can be sufficiently large), at each time step $t \in [T]$, The purpose of MSTF is to

forecast the values of the next H steps $\{x_{t+1}, \cdots, x_{t+H}\}$, where $H \in \mathbb{Z}_{\geq 1}$ is a predefined parameter. For notational convenience, let $Y_t = \{y_1, \cdots, y_H\}$ denote this ground truth sequence.

Definition 3 (ML-based Online Forecaster)**.** An online forecaster is a machine-learning model f parameterized by θ_t at time step t. Given the latest L observations $X_t = \{x_{t-L+1}, \cdots, x_t\}$, it predicts the next H values: $f(X_t; \theta_t) = \widehat{Y}_t = \{\widehat{y}_1, \cdots, \widehat{y}_H\}$. After Y_t is revealed, the loss $\ell(Y_t, \widehat{Y}_t)$ is used to update the model parameters.

For simplicity, we can use $f(X)$ as a substitute for $f(X; \theta)$ to represent the learning model if we focus on the output of f rather than the variation of θ in this process.

2.2 Online Ensembling

At each time step, multiple forecasters produce individual predictions. We can ensemble these results using *Online Convex Optimization* (OCO) to improve robustness.

Definition 4 (Weighted Online Ensembling)**.** At each time step t, given multiple forecasters $\boldsymbol{f} = \{f_1, \cdots, f_\Phi\}$, each with a forecast result $\widehat{Y}_t^\varphi$, define $w_t^\varphi \in \mathbb{R}^+$ as the weight of the φ-th forecaster. Let $\boldsymbol{w}_t = \{w_t^1, \cdots, w_t^\Phi\}$. The ensembling result is $\widetilde{Y}_t = \sum_{\varphi=1}^{\Phi} w_t^\varphi \widehat{Y}_t^\varphi$.

The performance of an online algorithm is evaluated by *regret* Hoi et al. (2021), defined as the cumulative loss difference between the algorithm and the best fixed forecaster in hindsight across the entire online process (over T sequential time steps).

Definition 5 (Regret)**.**

$$\text{Regret}_T = \sum_{t=1}^{T} \ell(Y_t, \widetilde{Y}_t) - \inf_{\boldsymbol{w}} \sum_{t=1}^{T} \ell\left(Y_t, \sum_{\varphi=1}^{\Phi} w^\varphi \widehat{Y}_t^\varphi\right).$$

The second term represents the loss incurred by the optimal parameter set $\boldsymbol{w}^*$ chosen after observing the entire data stream. An online algorithm is considered efficient if its regret is *sub-linear* w.r.t. T, i.e., $\text{Regret}_T = o(T)$. This implies that $\lim_{T\to\infty} \frac{\text{Regret}_T}{T} = 0$, indicating that the ensemble model performs as well as the best fixed weight configuration chosen after observing the entire data stream.

Thus, its average loss asymptotically approaches that of the best fixed ensemble in hindsight.

2.3 Two-Phase Online Learning

We next define the historical dataset used for offline training and the two-phase training strategy adopted in online forecasting.

Definition 6 (Historical Dataset)**.** Let $\mathcal{X} = \{(X^1, Y^1), \cdots, (X^S, Y^S)\}$ denote a historical dataset containing S input–target pairs. Each $X^s = \{x_1^s, \cdots, x_L^s\}$ is a historical sequence of length L, and $Y^s = \{y_1^s, \cdots, y_H^s\}$ is its corresponding future sequence of length H.

Definition 7 (Two-Phase Online Learning Strategy)**.** The learning process of f consists of offline training and online updating, both using the loss function $\ell(\cdot, \cdot)$.

The offline phase uses $\mathcal{X}$ to warm up the parameters θ_0. The objective is to minimize the average loss $\frac{1}{S}\left(\sum_{s=1}^{S} \ell\left(Y^s, f(X^s; \theta_0)\right)\right)$.

The online phase sequentially updates the model parameters based on real-time data stream with the following steps: **Forecasting:** The model generates $\widehat{Y}_t$ based on X_t. **Revelation:** The true value Y_t is revealed. **Update:** The model updates its parameters from θ_t to θ_{t+1} using $\ell(Y_t, \widehat{Y}_t)$ to improve future performance.

For simplicity, during offline training, we omit the sample superscript s and use $X = \{x_1, \cdots, x_L\}$ to denote an arbitrary historical sequence. We also omit the initialization subscript and write $\theta = \theta_0$.

2.4 Mask-Based Transformer

We consider a Transformer encoder-based forecaster f with five components: ① an embedding layer $\text{EMB} : \mathbb{R}^L \to \mathbb{R}^{L\times D}$; ② self-attention ATT; ③ batch normalization BN; ④ a feed-forward network

FFN, where the latter three map $\mathbb{R}^{L\times D}$ to $\mathbb{R}^{L\times D}$; and ⑤ a forecasting head $\text{HEAD} : \mathbb{R}^{L\times D} \to \mathbb{R}^{H}$. Here, D denotes the representation dimension.

Definition 8 (Relation Masks)**.** Given a time series $X \in \mathbb{R}^{L}$, we define a relation mask $M = (m_{ij})_{L\times L}$, where

$$m_{ij} = \begin{cases} 0, & x_i \text{ relates to } x_j, \forall i, j \in \{1, \cdots, L\}; \\ -\infty, & \text{otherwise.} \end{cases}$$

Let Φ be the number of masks, define $\boldsymbol{M} = \{M_1, \cdots, M_\Phi\}$ as a mask set.

Definition 9 (Masked Transformer Forecaster)**.** Given X and $M_\varphi \in \boldsymbol{M}$, the masked forecaster f_φ first maps X into an embedding:

$$Z = \text{EMB}(X). \tag{1}$$

After projecting Z into $Q, K, V \in \mathbb{R}^{L\times D}$, masked self-attention is computed as

$$\text{ATT}(Z \mid M_\varphi) = \text{softmax}\left(\frac{QK^\top}{\sqrt{D}} + M_\varphi\right) V, Z' = \text{BN}\left(Z + \text{ATT}(Z \mid M_\varphi)\right), Z'' = \text{BN}\left(Z' + \text{FFN}(Z')\right). \tag{2}$$

After repeating Eqn. 2 multiple times, we get $Z_O \in \mathbb{R}^{L\times D}$. Finally, $\text{HEAD}(Z_O)$ outputs a result.

3 MiMe Model Structure

This section presents the architecture and training procedure of MiMe (shown in Fig.2). We first transform the input series into patch embeddings and construct relation masks to constrain self-attention for heterogeneous temporal patterns. We then integrate the resulting masked forecasters through a shared Transformer backbone, followed by offline warm-up, online updating, and online ensembling for streaming adaptation.

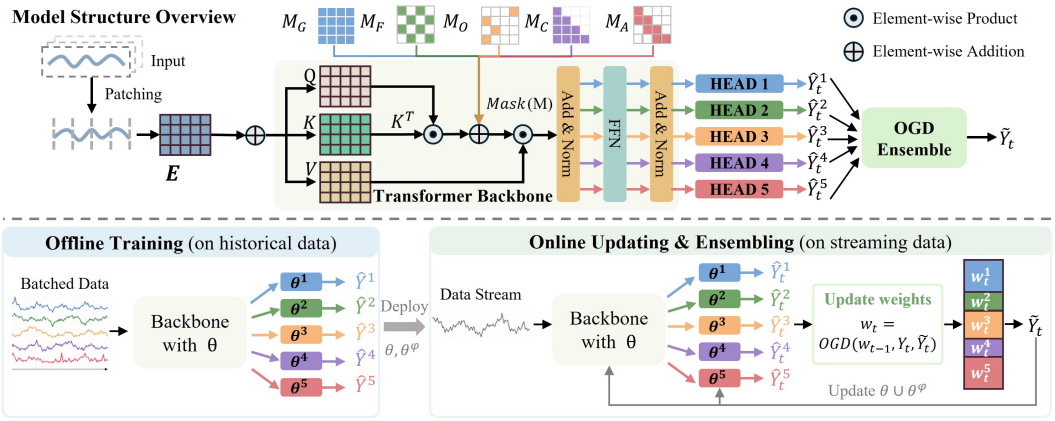


Figure 2: Overview of MiMe. Multiple mask-specific forecasters with complementary receptive-field biases share a Transformer backbone. The model is first trained offline on batched historical data and then updated online on streaming data, while OGD dynamically reweights the forecasters to produce the final ensemble prediction.

3.1 Patching & Embedding

We patch the input time series $X \in \mathbb{R}^{L}$ into P-length patches with stride λ Nie et al. (2023). the number of patches is $N = \lfloor \frac{L-P}{\lambda} \rfloor + 2$. We get the patch series is $\mathbf{X} = \{X_1, X_2, \ldots, X_N\} \in \mathbb{R}^{N\times P}$, where the n-th patch is $X_n = \{x_{(p-1)P+1}, x_{(p-1)P+2}, \ldots, x_{pP}\} \in \mathbb{R}^{P}$. Then we use $\text{EMB} : \mathbb{R}^{N\times P} \to \mathbb{R}^{N\times D}$ to get the X's embedding $\mathbf{E} = \{E_1, E_2, \ldots, E_N\} \in \mathbb{R}^{N\times D}$.

3.2 Mask Construction

To capture heterogeneous temporal patterns, we construct relation masks that induce complementary receptive fields by constraining patch-wise connectivity in self-attention. Specifically, we consider:

Global Mask: $M_G \in \mathbb{R}^{N\times N}$ ensures full connectivity, enabling the model to capture long-range dependencies. Formally, $m_{ij}^G = 0, \quad \forall i, j \in \{1, \ldots, N\}$.

Periodic Mask: M_F targets time series with strong periodicity. Each step attends exclusively to positions separated by integer multiples of the dominant periods, modeling recurring behaviors while ignoring irrelevant connections. We first use the Fourier Transform on the series X

$$\overline{X} = \text{FFT}(X) = \langle \overline{x}_1, \overline{x}_2, \ldots, \overline{x}_{\frac{L}{2}+1} \rangle \in \mathbb{C}^{\frac{L}{2}+1}.$$

TOPK function is used to select the Υ factors with the biggest amplitude in $\overline{X}$, and return the factor's frequency bin $\boldsymbol{\nu}$, where $\boldsymbol{\nu} = \text{TOPK}(\overline{X}) = \langle \nu_1, \ldots, \nu_\Upsilon \rangle \in \mathbb{R}^\Upsilon$.

We can further get the Λ dominant periods in patch series, i.e., $\boldsymbol{\tau} = \langle \tau_1, \ldots, \tau_\Upsilon \rangle \in \mathbb{R}^\Upsilon$, where the κ-th dominant period in the patch series is $\tau_\kappa = \lceil \frac{1}{\nu_\kappa \lambda} \rceil$. The Periodic Mask is calculated as:

$$m_{ij}^F = \begin{cases} 0, & \text{if } |i-j| \pmod{\tau_\kappa} = 0, \text{ for } \kappa = 1, \ldots, \Upsilon. \\ -\infty, & \text{otherwise.} \end{cases}$$

Contextual Mask: M_O adapts to dynamic dependencies, selecting relevant positions based on semantic similarity rather than fixed structures. After getting $Q, K, V \in \mathbb{R}^{N\times D}$, Kullback-Leibler Divergence Measurement KLM Zhou et al. (2021) is used to measure the "importance" of queries:

$$\text{KLM}(\mathbf{q}_i, K) = \ln \sum_{j=1}^{N} e^{\frac{\mathbf{q}_i \mathbf{k}_j^T}{\sqrt{D}}} - \frac{1}{N} \sum_{j=1}^{N} \frac{\mathbf{q}_i \mathbf{k}_j^T}{\sqrt{D}},,$$

where $\mathbf{q}_i = Q_{i,:}$ denotes the i-th row of Q. We define the TOP-KLM(Q, K) function to get the top-β factors in Q with the biggest KLM$(\mathbf{q}_i, K)$ for each K. The Contextual mask is calculated as:

$$m_{ij}^O = \begin{cases} 0, & \text{if } \quad \mathbf{q}_i \in \text{TOP-KLM}(Q, K). \\ -\infty, & \text{otherwise.} \end{cases}$$

Causal Mask: M_C restricts the receptive field to past and present steps, preventing future information leakage. It is implemented as a lower triangular matrix:

$$m_{ij}^C = \begin{cases} 0, & \text{if } i \geq j; \\ -\infty, & \text{otherwise.} \end{cases}$$

Local Mask: M_A focuses on immediate temporal neighborhoods within a radius w, improving robustness to concept drift by emphasizing recent context, forming a band mask:

$$m_{ij}^A = \begin{cases} 0, & \text{if } |i-j| \leq w; \\ -\infty, & \text{otherwise.} \end{cases}$$

We get the mask set $\mathbf{M} = \{M_G, M_F, M_O, M_C, M_A\} \in \mathbb{R}^{\Phi\times N\times N}$, each mask can be represented as $M_\varphi \in \mathbf{M}$, $\varphi = 1, \cdots, \Phi$ (here $\Phi = |\mathbf{M}|$).

3.3 Shared Backbone Setting

Inspired by Havasi et al. (2021), we leverage the redundancy of Transformer parameters to train multiple models within the same Transformer backbone, reducing computational cost.

The forecaster set $\boldsymbol{f} = \{f_1, \cdots, f_\Phi\}$ consists of Φ identical Masked Transformer. We decompose the parameters of each f_φ into two distinct parts: the shared parameters θ in **Backbone**, corresponds to the parameters employed in Eqn. 1 and Eqn. 2, and the mask-specific parameters θ^φ for prediction **Head**, corresponds to those employed in HEAD component in Def. 9.

3.4 Offline Training

During offline training, we aim to warm-up Φ forecasters in $\boldsymbol{f}$ with the historical dataset $\mathcal{X}$. Inspired by Bootstrap Sampling Alpaydin (2020), we adopt multi-round random shuffling, where the training data are randomly reordered and traversed for Φ rounds. In each round, the dataset is partitioned into batches and processed sequentially. For each sample, masks are constructed to constrain the receptive field according to its temporal patterns. Masked self-attention is applied to extract representations, which are then fed into the corresponding mask-specific prediction head.

Algorithm 1: Batch-Level Multi-Head Training

Input: Loss function $\ell(\cdot,\cdot)$; model number Φ; batch size B; historical dataset $\mathcal{X}$ with S pair of samples.
Output: Updated parameter set $\boldsymbol{\theta}$.

for 1 **to** Φ **do** // Repeat Φ times
  Shuffle the dataset $\mathcal{X}$ and reindexing;
  Partition $\mathcal{X}$ into $\frac{S}{B}$ batches as $\mathcal{X}^1, \cdots, \mathcal{X}^{\frac{S}{B}}$;
  Initialize Masks M_G, M_L, M_A; Set total loss $\mathcal{L} \leftarrow 0$;
  for $i \leftarrow 1$ **to** $\frac{S}{B}$ **do** // Dataset
    for $j \leftarrow 1$ **to** B **do** // Each Batch
      Compute Masks M_F, M_O;
      $s \leftarrow (i-1)\cdot B + j$; $\varphi \leftarrow i \bmod \Phi$;
      $\widehat{Y}^s \leftarrow f_\varphi(X^s; \theta \cup \theta^\varphi)$;
      $\mathcal{L} \leftarrow \mathcal{L} + \ell(Y^s, \widehat{Y}^s)$;
    if $i \bmod \Phi = 0$ **then**
      Update $\boldsymbol{\theta}$ by backpropagating $\frac{\mathcal{L}}{\Phi B}$;
      $\mathcal{L} \leftarrow 0$;

Parameter updates are performed in an accumulated manner. Prediction heads are alternately activated, and gradients are accumulated across Φ heads. Once all heads are trained once, a unified backpropagation step updates both the shared backbone θ and all head parameters $\{\theta^\varphi\}_{\varphi=1}^{\Phi}$.

Offline training runs for up to e epochs with early stopping based on validation loss. In each epoch, Alg. 1 is executed, where parameter updates are performed at the batch level.

3.5 Online Updating & Ensembling

Unlike offline training, which learns generalized representations via shuffled batch updates, the online updating phase performs sequential updates according to the data stream X_t to track evolving temporal patterns. At each timestamp t, the forecaster f operates on a historical window of length L and outputs a prediction $\widehat{Y}_t \in \mathbb{R}^H$. The online updating procedure is summarized in Alg. 2. We denote the parameters initialized from offline training as $\boldsymbol{\theta}_0$.

The ensemble prediction $\widetilde{Y}_t$ at time step t is obtained by aggregating Φ forecasters:

$$\widetilde{Y}_t = \sum_{\varphi=1}^{\Phi} w_{t-1}^{\varphi} \widehat{Y}_t^{\varphi} = \sum_{\varphi=1}^{\Phi} w_{t-1}^{\varphi} \cdot f_\varphi(X_t; \theta_{t-1} \cup \theta_{t-1}^{\varphi}).$$

Here, we apply the Simple Additive Weighting (SAW) method Chong et al. (2023), which satisfies $\sum_{\varphi=1}^{\Phi} w_t^{\varphi} = 1$. To initialize the process, let $\boldsymbol{w}_0$ be a randomly initialized weight. Similar to Alg. 2, $\boldsymbol{w}_t$ will be updated according to $\boldsymbol{w}_{t-1}$ at the end of time step t.

We consider $\ell_t(\cdot,\cdot)$ as the Mean Squared Error (MSE) corresponds to $\boldsymbol{w}_t$, where

$$\ell_t(Y_t, \widetilde{Y}_t) = |Y_t - \widetilde{Y}_t|^2 = |Y_t - \sum_{\varphi=1}^{\Phi} w_{t-1}^{\varphi} \widehat{Y}_t^{\varphi}|^2.$$

We adopt the Online Gradient Descent Algorithm (Zinkevich's online version of Stochastic Gradient Descent) Hazan et al. (2006) to update $\boldsymbol{w}_t$, shown in Alg. 3.

3.6 Regret Analysis

According to classical OGD results Hazan et al. (2006), logarithmic regret can be achieved when the loss function is strongly convex and its gradients are uniformly bounded.

At time step t, the ensemble loss can be expressed as:

$$\ell_t(\boldsymbol{w}_t) = \|F_t \boldsymbol{w}_t - Y_t\|_2^2,$$

Algorithm 2: Online Updating

Input: Time step t; data stream X_t; truth Y_t; loss function $\ell(\cdot,\cdot)$; parameter $\boldsymbol{\theta}_{t-1}$.
Output: Updated parameters $\boldsymbol{\theta}_t$.

Update Masks M_F, M_O;
foreach $\varphi = 1, \ldots, \Phi$ **do**
  // Forward Inference
  $\widehat{Y}_t^{\varphi} \leftarrow f_{\varphi}(X_t; \theta_{t-1} \cup \theta_{t-1}^{\varphi})$;
  $\mathcal{L} \leftarrow \ell(Y_t, \widehat{Y}_t^{\varphi})$;
  Update $\theta_{t-1} \cup \theta_{t-1}^{\varphi}$ by backpropagating $\mathcal{L}$;

Algorithm 3: Online Gradient Descent

Input: Loss function $\ell(\cdot,\cdot)$; $\boldsymbol{w}_{t-1}$; step size constant η
Output: $\boldsymbol{w}_t$.

// Global Setting
Compute $\nabla^2 \ell_t(\boldsymbol{w}_{t-1}) \in \mathbb{R}^{\Phi \times \Phi}$ (Hessian Matrix);
Compute lower bound Λ: $\nabla^2 \ell(\boldsymbol{w}) \succeq \Lambda I_{\Phi}, \forall \boldsymbol{w} \in [0,1]^{\Phi}$;
// Stepped Updating
$\eta_t = \eta/(\Lambda t)$;
$\boldsymbol{w}_t = \arg\min_{\boldsymbol{w}} \| \boldsymbol{w} - (\boldsymbol{w}_{t-1} - \eta_t \nabla \ell_{t-1}(\boldsymbol{w}_{t-1})) \|$;

where $F_t \in \mathbb{R}^{H \times \Phi}$ stacks the forecasting vectors produced by all Φ forecasters. However, since predictions from different forecasters may be linearly dependent, the Hessian matrix $(F_t)^\top F_t$ is not guaranteed to be positive definite, which violates the strong convexity requirement of OGD.

To guarantee strong convexity, we adopt an augmentation technique inspired by Tikhonov regularization Tikhonov & Arsenin (1977), we augment the prediction matrix and target vector as

$$\tilde{F}_t = \begin{bmatrix} F_t \\ \frac{\sigma}{\sqrt{\eta}} I \end{bmatrix}, \quad \tilde{Y}_t = \begin{bmatrix} Y_t \\ 0 \end{bmatrix},$$

where $\sigma > 0$ is a regularization constant. Under this formulation, the loss becomes $\ell_t(\boldsymbol{w}_t) = \|\tilde{F}_t \boldsymbol{w}_t - \tilde{Y}_t\|_2^2$, whose Hessian is given by

$$\nabla^2 \ell_t(\boldsymbol{w}_t) = (F_t)^\top F_t + \frac{\sigma^2}{\eta} I.$$

The loss function is Λ-strongly convex with $\Lambda \geq \frac{\sigma^2}{\eta}$.

Moreover, under mild boundedness assumptions on the data stream and forecasters' outputs, the gradient norm admits a uniform upper bound. Let Γ denote a constant such that $\|\nabla \ell_t(\boldsymbol{w})\|_2 \leq \Gamma$ for all $\boldsymbol{w} \in \mathcal{W}$ and all t.

With the step size schedule $\eta_t = \eta/(\Lambda t)$, the OGD algorithm guarantees the following regret bound:

$$\sum_{t=1}^{T} \ell_t(\boldsymbol{w}_t) - \min_{\boldsymbol{w} \in \mathcal{W}} \sum_{t=1}^{T} \ell_t(\boldsymbol{w}) \;\leq\; \frac{\Gamma^2}{\Lambda} \log(T+1),$$

where $\mathcal{W} = \{\boldsymbol{w} \in \mathbb{R}_{\geq 0}^{\Phi} : \|\boldsymbol{w}\|_1 = 1\}$. This establishes an $O(\log T)$ regret for the proposed online ensembling. Detailed assumptions and proofs are in Appendix A, symbol table is in Appendix C.

4 Experiments

4.1 Experiment Setup

Datasets. We adopt 7 datasets: ETT (15-min/1-h) and Weather (10-min) are high-frequency time series. Traffic provides large-scale hourly records from sensors and customers with complex dependencies. Exchange contains daily economic data marked by high noise and weak periodicity. Detailed statistics of the dataset are provided in the Appendix D.

Baselines. We adapt 14 baselines from 3 categories: Transformer-based models: FEDformer Zhou et al. (2022), Pyraformer Liu et al. (2022), Crossformer Zhang & Yan (2022), iTransformer Liu et al. (2024), and PatchTST Nie et al. (2023). Attention-free models: LightTS Campos et al. (2023), TSMixer Chen et al. (2023b), TimeMixer Wang et al. (2024), WPMixer Murad et al. (2025), PAttn Tan et al. (2024), and TimesNet Wu et al. (2023). Online forecasting models: FSNet Pham et al. (2023), Time-FSNet, and OneNet Zhang et al. (2023). Implementation details are provided in the Appendix E.

4.2 Comparative Results

Tab. 1 reports the overall forecasting performance. Across all datasets and forecasting horizons, MiMe achieves the best overall average performance, reducing MSE and MAE by 3.83% and 3.53%,

Table 1: Performance comparison under forecasting horizon $H = \{12, 24\}$.The best is in **boldface** and the second best is underlined.

Model	Metric	ETTh1			ETTh2			ETTm1			ETTm2			Weather			Traffic			Exchange		
		12	24	Avg	12	24	Avg	12	24	Avg	12	24	Avg	12	24	Avg	12	24	Avg	12	24	Avg
FEDformer	MSE	1.458	1.545	1.502	8.175	8.359	8.267	1.294	1.462	1.378	2.731	2.802	2.766	2.384	2.619	2.502	0.784	0.852	0.818	1.686	1.818	1.752
	MAE	0.831	0.851	0.841	1.099	1.115	1.107	0.754	0.796	0.775	0.724	0.728	0.726	0.996	1.053	1.025	0.534	0.580	0.557	0.932	0.982	0.957
Pyraformer	MSE	1.592	1.709	1.650	13.047	15.353	14.200	0.801	1.204	1.002	9.293	11.051	10.172	1.334	2.099	1.717	0.742	0.760	0.751	2.284	4.095	3.189
	MAE	0.842	0.893	0.867	1.384	1.722	1.553	0.588	0.722	0.655	0.889	1.050	0.970	0.642	0.833	0.737	0.447	0.444	0.446	1.130	1.516	1.323
Crossformer	MSE	1.482	1.443	1.462	12.331	13.729	13.030	0.960	1.215	1.088	9.063	10.177	9.620	1.134	1.732	1.433	0.755	1.060	0.908	1.921	3.283	2.602
	MAE	0.838	0.839	0.838	1.314	1.452	1.383	0.641	0.714	0.677	0.916	0.990	0.953	0.589	0.771	0.680	0.476	0.666	0.571	1.006	1.386	1.196
iTransformer	MSE	0.906	0.993	0.949	2.974	4.233	3.603	0.493	0.654	0.573	0.704	0.998	0.851	0.854	1.097	0.976	0.480	0.566	0.523	0.527	0.780	0.653
	MAE	0.620	0.660	0.640	0.711	0.800	0.755	0.436	0.506	0.471	0.422	0.476	0.449	0.471	0.565	0.518	0.367	0.424	0.396	0.482	0.587	0.534
PatchTST	MSE	1.151	1.185	1.168	3.482	4.158	3.820	0.649	0.760	0.705	0.963	1.097	1.030	1.040	1.135	1.088	0.498	0.502	0.500	0.524	0.790	0.657
	MAE	0.686	0.703	0.694	0.779	0.878	0.829	0.493	0.543	0.518	0.451	0.495	0.473	0.516	0.562	0.539	0.381	0.376	0.379	0.471	0.579	0.525
LightTS	MSE	1.025	1.110	1.067	2.357	3.281	2.819	0.503	0.664	0.583	0.587	0.787	0.687	0.853	1.110	0.981	0.514	0.547	0.531	0.531	0.790	0.660
	MAE	0.677	0.716	0.696	0.736	0.852	0.794	0.451	0.526	0.489	0.437	0.488	0.463	0.473	0.595	0.534	0.401	0.414	0.407	0.534	0.632	0.583
TSMixer	MSE	0.982	1.044	1.013	2.348	3.288	2.818	0.467	0.593	0.530	0.712	1.028	0.870	0.843	1.098	0.971	0.465	0.510	0.488	0.538	1.133	0.835
	MAE	0.664	0.695	0.680	0.832	0.955	0.894	0.446	0.508	0.477	0.494	0.566	0.530	0.494	0.591	0.543	0.373	0.401	0.387	0.555	0.782	0.669
TimeMixer	MSE	0.945	1.078	1.011	2.420	4.062	3.241	**0.431**	0.589	0.510	0.575	0.732	0.654	0.719	0.896	0.807	**0.358**	**0.380**	**0.369**	0.434	0.719	0.576
	MAE	0.621	0.675	0.648	0.700	0.840	0.770	0.414	0.490	0.452	0.398	0.430	0.414	0.403	0.477	0.440	**0.279**	**0.285**	**0.282**	0.427	0.554	0.491
WPMixer	MSE	0.893	0.893	0.893	2.039	2.605	2.322	0.446	0.569	0.507	0.555	0.726	0.641	0.717	0.860	0.789	0.403	0.419	0.411	0.310	0.487	0.398
	MAE	0.606	0.606	0.606	0.641	0.706	0.674	0.421	0.486	0.454	0.388	0.422	0.405	0.405	0.459	0.432	0.333	0.337	0.335	**0.368**	0.459	0.413
PAttn	MSE	0.950	1.000	0.975	2.770	3.376	3.073	0.453	0.575	0.514	0.697	0.950	0.824	0.724	0.877	0.801	0.407	0.416	0.411	0.319	0.614	0.466
	MAE	0.618	0.642	0.630	0.732	0.830	0.781	0.420	0.478	0.449	0.397	0.454	0.426	0.402	0.469	0.435	0.331	0.328	0.329	0.379	0.515	0.447
TimesNet	MSE	0.978	1.043	1.010	3.463	4.613	4.038	0.532	0.663	0.598	1.315	1.447	1.381	1.083	1.183	1.133	0.714	0.858	0.786	0.604	1.004	0.804
	MAE	0.645	0.675	0.660	0.907	0.966	0.937	0.460	0.513	0.486	0.586	0.587	0.587	0.576	0.607	0.591	0.498	0.579	0.538	0.566	0.669	0.618
FSNet	MSE	1.339	1.344	1.342	3.149	6.881	5.015	0.513	0.686	0.599	0.767	1.521	1.144	0.756	0.983	0.870	0.550	0.559	0.555	0.540	1.210	0.875
	MAE	0.718	0.742	0.730	0.868	1.133	1.001	0.457	0.559	0.508	0.503	0.653	0.578	0.450	0.540	0.495	0.399	0.378	0.389	0.548	0.820	0.684
Time-FSNet	MSE	2.061	2.149	2.105	10.588	12.446	11.517	1.437	1.708	1.572	8.032	8.313	8.172	2.976	3.286	3.131	0.767	0.773	0.770	3.230	3.174	3.202
	MAE	0.967	0.974	0.970	1.518	1.594	1.556	0.836	0.901	0.868	1.220	1.232	1.226	1.086	1.139	1.112	0.501	0.496	0.499	1.158	1.189	1.174
OneNet	MSE	1.017	0.949	0.983	2.178	2.722	2.450	0.501	0.689	0.595	0.540	0.741	0.641	0.746	0.885	0.815	0.451	0.480	0.465	0.663	0.960	0.811
	MAE	0.629	0.641	0.635	0.712	0.783	0.748	0.444	0.530	0.487	0.408	0.460	0.434	0.431	0.485	0.458	0.337	0.345	0.341	0.569	0.701	0.635
MiMe	MSE	**0.893**	**0.887**	**0.890**	**1.745**	**2.254**	**2.000**	0.446	**0.568**	**0.507**	**0.533**	**0.686**	**0.610**	**0.716**	**0.830**	**0.773**	0.396	0.403	0.399	**0.304**	**0.472**	**0.388**
	MAE	**0.588**	**0.588**	**0.588**	**0.593**	**0.674**	**0.634**	**0.408**	**0.469**	**0.439**	**0.368**	**0.412**	**0.390**	**0.388**	**0.440**	**0.414**	0.320	0.316	0.318	0.366	**0.442**	**0.404**

respectively, over the second-best baselines. On high-frequency datasets like ETT and Weather, MiMe achieves consistently leading performance, showing its ability to capture rapidly changing temporal patterns. On the challenging Exchange dataset, which is noisy and weakly periodic, MiMe also achieves the lowest MSE and MAE, indicating robustness to noisy financial series. For the high-dimensional Traffic dataset, TimeMixer obtains the best results, while MiMe ranks second and still outperforms the remaining baselines. This suggests that although high-dimensional multivariate forecasting remains challenging, the proposed multi-mask design and online ensembling mechanism provide strong robustness across diverse temporal scenarios. Overall, the results demonstrate that MiMe effectively balances pattern-specific temporal modeling and adaptive online aggregation.

4.3 Ablation Study and Further Discussion

RQ1: Does mask diversity improve online forecasting?

Constraining the receptive field of Transformers with diverse relation masks is a core design of MiMe. We therefore investigate how progressively incorporating heterogeneous masks affects forecasting performance. As shown in Tab. 2, increasing mask diversity generally leads to consistent performance gains across all datasets. Configurations with 4 masks already achieve the second-best performance, while the specific four-mask combination varies across datasets, reflecting their different dominant temporal characteristics. This observation suggests that no fixed subset of inductive biases is universally optimal for heterogeneous time series. When all five masks are incorporated, MiMe achieves the best performance on every dataset, providing further improvements over the strongest four-mask configurations. These results indicate that a more diverse set of masks provides complementary receptive

Table 2: Ablation results averaged over forecasting horizon $H = \{12, 24\}$, demonstrating the effect of diverse masks.

Config	Metric	ETTh1	ETTh2	ETTm1	ETTm2	Weather	Traffic	Exchange
M_G	MSE	1.1327	3.5684	0.7024	0.9489	1.0335	0.5030	0.5829
	MAE	0.6739	0.7984	0.5115	0.4516	0.5066	0.3817	0.4888
M_G+M_F	MSE	0.9839	2.8699	0.5958	0.7710	0.8902	0.4379	0.4591
	MAE	0.6265	0.7359	0.4771	0.4222	0.4657	0.3466	0.4396
M_G+M_C	MSE	0.9827	2.7825	0.5937	0.7933	0.8821	0.4375	0.4564
	MAE	0.6254	0.7268	0.4758	0.4261	0.4630	0.3464	0.4371
M_G+M_O	MSE	0.9790	2.9437	0.5885	0.7368	0.8629	0.4371	0.4629
	MAE	0.6239	0.7354	0.4738	0.4138	0.4569	0.3455	0.4439
M_G+M_A	MSE	0.9844	2.9460	0.5958	0.7703	0.8947	0.4377	0.4592
	MAE	0.6265	0.7429	0.4770	0.4235	0.4661	0.3465	0.4406
M_G+M_F+M_C	MSE	0.9341	2.5979	0.5559	0.7356	0.8402	0.4166	0.4167
	MAE	0.6080	0.7078	0.4616	0.4140	0.4477	0.3319	0.4188
M_G+M_F+M_O	MSE	0.9312	2.6833	0.5556	0.7147	0.8319	0.4164	0.4197
	MAE	0.6076	0.7132	0.4611	0.4088	0.4447	0.3312	0.4216
M_G+M_F+M_A	MSE	0.9358	2.7308	0.5588	0.7238	0.8432	0.4167	0.4171
	MAE	0.6089	0.7200	0.4629	0.4131	0.4488	0.3319	0.4210
M_G+M_F+M_C+M_O	MSE	0.9127	2.5401	0.5374	0.6867	0.8120	0.4061	0.3956
	MAE	0.5993	0.6993	0.4538	0.4030	0.4356	0.3230	0.4081
M_G+M_F+M_C+M_A	MSE	0.9139	2.5393	0.5395	0.6960	0.8191	0.4059	0.3956
	MAE	0.5992	0.6962	0.4546	0.4056	0.4384	0.3232	0.4083
M_G+M_F+M_O+M_A	MSE	0.9127	2.6150	0.5379	0.6725	0.8141	0.4062	0.3976
	MAE	0.6000	0.7061	0.4539	0.4013	0.4362	0.3230	0.4110
M_G+M_C+M_O+M_A	MSE	0.9114	2.5553	0.5350	0.6858	0.8103	0.4060	0.3958
	MAE	0.5994	0.7015	0.4525	0.4028	0.4346	0.3230	0.4091
M_G+M_F+ M_C+M_O+M_A	MSE	**0.8899**	**1.9997**	**0.5069**	**0.6098**	**0.7727**	**0.3994**	**0.3885**
	MAE	**0.5882**	**0.6336**	**0.4386**	**0.3899**	**0.4144**	**0.3176**	**0.4039**

fields and a broader set of temporal modeling strategies, improving the robustness of forecasting across heterogeneous temporal patterns. The detailed results are in Appendix F.1.

RQ2: Is the gain from mask diversity or merely from more experts?

Mixture-of-Experts (MoE) have shown that increasing the number of experts can improve model capacity and forecasting performance Fedus et al. (2022). Therefore, the gain of MiMe may also come from using more experts rather than from heterogeneous masks. To isolate the effect of mask diversity, we fix the number of experts to five and compare MiMe with homogeneous expert ensembles, where all experts use the same mask, i.e., $5 \times M_G$, $5 \times M_F$, $5 \times M_C$, $5 \times M_O$, and $5 \times M_A$. Tab. 3 shows the results, details are in Appendix F.2.

Table 3: Expert diversity comparison. Results are averaged over forecasting horizon $H = \{12, 24\}$.

Config	Metric	ETTh1	ETTh2	ETTm1	ETTm2	Weather	Traffic	Exchange
$5\times M_G$	MSE	0.895	2.271	0.520	0.637	0.789	0.415	0.411
	MAE	0.590	0.663	0.446	0.401	0.423	0.329	0.417
$5\times M_F$	MSE	0.893	2.212	0.522	0.647	0.793	0.416	0.416
	MAE	0.590	0.658	0.448	0.398	0.426	0.330	0.418
$5\times M_C$	MSE	0.890	2.006	0.523	0.693	0.783	0.415	0.418
	MAE	**0.588**	0.649	0.449	0.406	0.425	0.329	0.420
$5\times M_O$	MSE	**0.885**	2.204	0.510	0.629	**0.771**	0.424	0.421
	MAE	**0.588**	0.650	0.445	**0.387**	0.422	0.338	0.423
$5\times M_A$	MSE	0.893	2.054	0.529	0.651	0.783	0.416	0.426
	MAE	0.591	0.641	0.452	0.399	0.423	0.330	0.423
MiMe	MSE	0.890	**2.000**	**0.507**	**0.610**	0.773	**0.399**	**0.388**
	MAE	**0.588**	**0.634**	**0.439**	0.390	**0.414**	**0.318**	**0.404**

Homogeneous experts can achieve the best result on some datasets when their mask bias matches the dominant temporal pattern. However, no homogeneous configuration remains consistently optimal. In contrast, MiMe achieves the best performance in most cases and second-best results in the remaining cases. This demonstrates that the improvement does not simply result from increasing the expert number or model capacity. Instead, heterogeneous masks provide complementary temporal inductive biases, enabling the ensemble to adapt more robustly to diverse online data streams.

RQ3: What is the role of the online ensemble strategy?

We investigate how the aggregation strategy affects performance. We adopt several ensemble methods: **Custom** directly averages the outputs of the five masked forecasters. FTPL Kalai & Vempala (2003) is a non-convex method that chooses one expert instead of forming a weighted combination; we use two variants with different perturbation distributions, i.e., **FTPL-Gumbel** with standard Gumbel noise and **FTPL-Exp** with standard exponential noise. We also compare with Exponentiated Gradient Descent **EGD** Kivinen & Warmuth (1997), the convex online aggregation method used in OneNet Zhang et al. (2023). Tab. 4 reports the results. It shows that FTPL methods perform worst in most cases, since selecting a single masked forecaster imposes a strong temporal-pattern assumption and discards complementary mask information. Convex aggregation is more stable by combining multiple forecasters. Compared with EGD, which has an $O(\sqrt{T})$ regret guarantee, our OGD achieves a tighter $O(\log T)$ regret under the regularized strongly convex objective, and consequently obtains the best performance on most datasets and metrics. The detailed results are in Appendix F.3.

Table 4: Ensemble method comparison averaged over prediction lengths $H = \{12, 24\}$.

Ensemble Method	Metric	ETTh1	ETTh2	ETTm1	ETTm2	Weather	Traffic	Exchange
Custom	MSE	0.894	2.148	0.517	0.645	0.785	0.406	0.444
	MAE	0.591	0.651	0.443	0.396	0.421	0.323	0.434
FTPL_{Exp}	MSE	1.074	2.785	0.658	0.846	0.963	0.501	0.578
	MAE	0.650	0.721	0.487	0.430	0.471	0.380	0.485
$\text{FTPL}_{\text{Gumbel}}$	MSE	1.072	2.745	0.655	0.866	0.962	0.502	0.573
	MAE	0.650	0.718	0.486	0.430	0.471	0.380	0.485
EGD	MSE	0.894	2.174	0.517	0.649	0.785	0.406	0.443
	MAE	0.591	0.654	0.443	0.396	0.421	0.323	0.434
OGD	MSE	**0.890**	**2.000**	**0.507**	**0.610**	**0.773**	**0.399**	**0.388**
	MAE	**0.588**	**0.634**	**0.439**	**0.390**	**0.414**	**0.318**	**0.404**

More experimental discussions on the efficiency of the shared backbone, the visualization of ensemble weights, enhanced online adaptation, and hyperparameter sensitivity are provided in the Appendix G.

5 Conclusion

This paper studies online time series forecasting for data streams and identifies key mismatches between Transformers and online adaptation requirements. We propose MiMe, a mask-ensembling framework that integrates heterogeneous temporal inductive biases through masked attention and adapts them online via principled ensemble optimization. MiMe integrates offline training with online updating and ensembling, enabling efficient adaptation to concept drift.

AI USE STATEMENT

Generative AI tools were used to assist with language editing, manuscript organization, figure presentation, and result interpretation. All AI-assisted suggestions were critically reviewed by the authors. The proposed method, experimental results, mathematical claims, and final manuscript were independently verified and finalized by the authors. The authors take full responsibility for the content of this work.

REPRODUCIBILITY STATEMENT

We provide complete assumptions and proofs for the regret analysis in Appendix A, detailed dataset statistics in Appendix D, and preprocessing, architecture and training configurations in Appendix E. Detailed ablation results and additional analyses are reported in Appendix F and Appendix G. Anonymous source code is also provided with the submission to facilitate reproduction of the experimental results: https://github.com/anonymous7538/AnonProj.

REFERENCES

Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2020.

Pierre-Daniel Arsenault, Shengrui Wang, and Jean-Marc Patenaude. A survey of explainable artificial intelligence (xai) in financial time series forecasting. *ACM Computing Surveys*, 57(10):1–37, 2025.

David Campos, Miao Zhang, Bin Yang, Tung Kieu, Chenjuan Guo, and Christian S Jensen. Lightts: Lightweight time series classification with adaptive ensemble distillation. *Proceedings of the ACM on Management of Data*, 2023.

Jiadong Chen, Yang Luo, Xiuqi Huang, Fuxin Jiang, Yangguang Shi, Tieying Zhang, and Xiaofeng Gao. IPOC: an adaptive interval prediction model based on online chasing and conformal inference for large-scale systems. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 202–212, 2023a.

Si-An Chen, Chun-Liang Li, Nathanael C Yoder, Sercan O Arik, and Tomas Pfister. Tsmixer: An all-mlp architecture for time series forecast-ing. *arXiv preprint arXiv:2303.06053*, 2023b.

Edwin KP Chong, Wu-Sheng Lu, and Stanislaw H Zak. *An Introduction to Optimization: With Applications to Machine Learning*. John Wiley & Sons, 2023.

Zihan Dong, Xinyu Fan, and Zhiyuan Peng. Fnspid: A comprehensive financial news dataset in time series. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4918–4927, 2024.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *J. Mach. Learn. Res.*, 23:120:1–120:39, 2022.

Mohan Gao, Kexin Xu, Xiaofeng Gao, Tengwei Cai, and Haoyuan Ge. Spatial-temporal heterogenous graph contrastive learning for microservice workload prediction. In *Conference on Artificial Intelligence, AAAI*, pp. 11681–11689, 2025.

Mohan Gao, Zhemeng Yu, Yang Luo, Lintao Ma, Yinbo Sun, Yuchen Fang, and Xiaofeng Gao. Macro-micro collaborative learning for logical data center microservice indicators forecasting. In *ACM Web Conference , WWW*, pp. 5010–5020, 2026.

Zhihan Gao, Xingjian Shi, Hao Wang, Yi Zhu, Yuyang Bernie Wang, Mu Li, and Dit-Yan Yeung. Earthformer: Exploring space-time transformers for earth system forecasting. *Advances in Neural Information Processing Systems, NeurIPS*, 35:25390–25403, 2022.

Marton Havasi, Rodolphe Jenatton, Stanislav Fort, Jeremiah Zhe Liu, Jasper Snoek, Balaji Lakshminarayanan, Andrew Mingbo Dai, and Dustin Tran. Training independent subnetworks for robust prediction. In *International Conference on Learning Representations, ICLR*, 2021.

Elad Hazan, Adam Kalai, Satyen Kale, and Amit Agarwal. Logarithmic regret algorithms for online convex optimization. In *Annual Conference on Learning Theory, COLT*, volume 4005, pp. 499–513, 2006.

Kareem Hegazy, Michael W. Mahoney, and N. Benjamin Erichson. Powerformer: A transformer with weighted causal attention for time-series forecasting. *CoRR*, abs/2502.06151, 2025.

Steven CH Hoi, Doyen Sahoo, Jing Lu, and Peilin Zhao. Online learning: A comprehensive survey. *NeuroComputing*, 459:249–289, 2021.

Charles C Holt. Forecasting seasonals and trends by exponentially weighted moving averages. *International journal of forecasting*, 20(1):5–10, 2004.

Yajin Huang, Yuncan Chen, Shimin Xu, Dongyan Wu, and Xunyi Wu. Self-supervised learning with adaptive frequency-time attention transformer for seizure prediction and classification. *Brain Sciences*, 15(4), 2025.

Adam Kalai and Santosh S. Vempala. Efficient algorithms for online decision problems. In *Annual Conference on Computational Learning Theory, COLT*, volume 2777, pp. 26–40, 2003.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *International Conference on Learning Representations, ICLR*, 2020.

Jyrki Kivinen and Manfred K Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and computation*, 132(1):1–63, 1997.

Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang, and Xifeng Yan. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. In *Annual Conference on Neural Information Processing Systems, NeurIPS*, pp. 5244–5254, 2019.

Yuxuan Liang, Yutong Xia, Songyu Ke, Yiwei Wang, Qingsong Wen, Junbo Zhang, Yu Zheng, and Roger Zimmermann. Airformer: Predicting nationwide air quality in china with transformers. In *AAAI conference on artificial intelligence*, volume 37, pp. 14329–14337, 2023.

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

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, ICLR*, 2024.

Zhipeng Liu, Peibo Duan, Xuan Tang, Baixin Li, Yongsheng Huang, Mingyang Geng, Changsheng Zhang, Bin Zhang, and Binwu Wang. Timeformer: Transformer with attention modulation empowered by temporal characteristics for time series forecasting. *CoRR*, abs/2510.06680, 2025.

Yang Luo, Mohan Gao, Zhemeng Yu, Haoyuan Ge, Xiaofeng Gao, Tengwei Cai, and Guihai Chen. Integrating system state into spatio temporal graph neural network for microservice workload prediction. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024a.

Yang Luo, Shiyu Wang, Zhemeng Yu, Wei Lu, Xiaofeng Gao, Lintao Ma, and Guihai Chen. Adaptive two-stage cloud resource scaling via hierarchical multi-indicator forecasting and bayesian decision-making. *CoRR*, 2024b.

Ibomoiye Domor Mienye, Theo G. Swart, and George Obaido. Recurrent neural networks: A comprehensive review of architectures, variants, and applications. *Information*, 15(9):517, 2024.

Md Mahmuddun Nabi Murad, Mehmet Aktukmak, and Yasin Yilmaz. Wpmixer: Efficient multi-resolution mixing for long-term time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 19581–19588, 2025.

Zelin Ni, Hang Yu, Shizhan Liu, Jianguo Li, and Weiyao Lin. Basisformer: Attention-based time series forecasting with learnable and interpretable basis. *Advances in Neural Information Processing Systems, NeurIPS*, 36:71222–71241, 2023.

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

Quang Pham, Chenghao Liu, Doyen Sahoo, and Steven C. H. Hoi. Learning fast and slow for online time series forecasting. In *International Conference on Learning Representations, ICLR*, 2023.

Mingtian Tan, Mike A Merrill, Vinayak Gupta, Tim Althoff, and Thomas Hartvigsen. Are language models actually useful for time series forecasting? *Advances in Neural Information Processing Systems, NeurIPS*, 37:60162–60191, 2024.

Andrey Nikolayevich Tikhonov and Vasiliy Yakovlevich Arsenin. *Solutions of Ill-posed Problems*. W. H. Winston & Sons, Washington, D.C., 1977. Translated from the Russian; preface by Fritz John.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Annual Conference on Neural Information Processing Systems, NeurIPS*, pp. 5998–6008, 2017.

Hao Wang, Wei Guo, Luankang Zhang, Jin Yao Chin, Yufei Ye, Huifeng Guo, Yong Liu, Defu Lian, Ruiming Tang, and Enhong Chen. Generative large recommendation models: Emerging trends in llms for recommendation. In *Companion Proceedings of the ACM on Web Conference 2025*, pp. 49–52, 2025.

Shiyu Wang, Yinbo Sun, Xiaoming Shi, Shiyi Zhu, Lintao Ma, James Zhang, Yangfei Zheng, and Liu Jian. Full scaling automation for sustainable development of green data centers. In *International Joint Conference on Artificial Intelligence IJCAI*, pp. 6264–6271, 2023.

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, ICLR*, volume 2024, pp. 38626–38652, 2024.

Billy M Williams and Lester A Hoel. Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results. *Journal of transportation engineering*, 129 (6):664–672, 2003.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: decomposition transformers with auto-correlation for long-term series forecasting. In *International Conference on Neural Information Processing Systems, NeurIPS*, 2021a.

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

Qiang Wu, Gechang Yao, Zhixi Feng, and Yang Shuyuan. Peri-midformer: Periodic pyramid transformer for time series analysis. *Advances in Neural Information Processing Systems, NeurIPS*, 37:13035–13073, 2024.

Yuhuai Wu, Markus N. Rabe, Wenda Li, Jimmy Ba, Roger B. Grosse, and Christian Szegedy. LIME: learning inductive bias for primitives of mathematical reasoning. In *International Conference on Machine Learning, ICML*, volume 139, pp. 11251–11262, 2021b.

Hengyu Ye, Jiadong Chen, Fuxin Jiang, Xiao He, Tieying Zhang, Jianjun Chen, and Xiaofeng Gao. Fremer: Lightweight and effective frequency transformer for workload forecasting in cloud services. *Proceedings of the VLDB Endowment*, 18(11):3812–3825, 2025.

Zhemeng Yu, Yang Luo, Yucen Gao, Yinbo Sun, Xiaofeng Gao, Lintao Ma, and Guihai Chen. Causalscaler: A causality-driven autoscaling framework for the cloud. In *International Conference on Database Systems for Advanced Applications DASFAA*, 2025.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big bird: Transformers for longer sequences. In *Annual Conference on Neural Information Processing Systems, NeurIPS*, 2020.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Conference on Artificial Intelligence, AAAI*, pp. 11121–11128, 2023.

Pengchuan Zhang, Xiyang Dai, Jianwei Yang, Bin Xiao, Lu Yuan, Lei Zhang, and Jianfeng Gao. Multi-scale vision longformer: A new vision transformer for high-resolution image encoding. In *International Conference on Computer Vision, ICCV*, pp. 2978–2988, 2021.

Qianru Zhang, Peng Yang, Junliang Yu, Haixin Wang, Xingwei He, Siu-Ming Yiu, and Hongzhi Yin. A survey on point-of-interest recommendation: Models, architectures, and security. *IEEE Transactions on Knowledge and Data Engineering*, 2025.

Yifan Zhang, Qingsong Wen, Xue Wang, Weiqi Chen, Liang Sun, Zhang Zhang, Liang Wang, Rong Jin, and Tieniu Tan. Onenet: Enhancing time series forecasting models under concept drift by online ensembling. In *Annual Conference on Neural Information Processing Systems, NeurIPS*, 2023.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *International Conference on Learning Representations, ICLR*, 2022.

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 *Conference on Artificial Intelligence, AAAI*, pp. 11106–11115, 2021.

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, ICML*, 2022.

A Regret Analysis

Suppose that each sample x_t of the online data stream is drawn (by an adversary) from an unknown set $\mathcal{X} \subseteq \mathcal{R}$, while every element $\hat{y}_h^i$ in the prediction vector generated by each forecaster i belongs to a set $\mathcal{Y}$.

Assumption A.1 (Constant upper bound assumption)**.** We assume that there exists an absolute positive constant $R_{\max} > 0$ such that

$$R_{\max} \geq \max\Big\{ \max\{|x| : x \in \mathcal{X}\}, \max\{|y| : y \in \mathcal{Y}\} \Big\}.$$

Assumption A.2 (Addition closed assumption)**.** Let $\hat{Y}$ and $\hat{Y}'$ be two arbitrary prediction vectors. We assume that for any real number $w \in [0, 1]$, the weighted sum $w \cdot \hat{Y} + (1 - w)\hat{Y}'$ is a feasible prediction.

With slight abuse of notation, we use $\ell_t : \mathbb{R}^{\Phi}_{\geq 0} \mapsto \mathbb{R}_{\geq 0}$ to represent the loss incurred by a weight vector $w_t \in \mathbb{R}^{\Phi}_{\geq 0}$ at step t with respect to the given input X_t and Y_t and the set of forcasters $\mathbf{f} = \{f_\varphi\}^{\varphi \in \Phi}$. Define $\Gamma = \max_{t\in[T]} \Big\{ \|\nabla \ell_t(w_t)\|_2 \Big\}$, and $\Lambda = \min_{\varphi\in[\Phi], t\in[T]} \Big\{ \lambda_\varphi\big(\nabla^2 \ell_t(w_t)\big) \Big\}$, where $\lambda_\varphi(\cdot)$ is the operator of taking the φ-th eigenvalue. The following proposition uses Γ and Λ to specify the regret of the OGD algorithm under our setting.

Lemma A.3 (Hazan et al. (2006))**.** *By taking the step size $\eta_t = \frac{\eta}{\Lambda t}$, the OGD algorithm guarantees that for any $T \geq 1$*

$$\sum_{t\in[T]} \ell_t(w_t) - \min_{w\in\mathcal{W}} \sum_{t\in[T]} \ell_t(w) \leq \frac{\Gamma^2}{\Lambda} \cdot \log(T + 1),$$

where $\mathcal{W} = \{w \in \mathbb{R}^{\Phi}_{\geq 0} : \|w\|_1 = 1\}$ represents the feasible set of w.

By Lemma A.3, to specify a valid step size and ensure well-bounded regret, we need to guarantee $\Lambda > 0$. Next, we show that this can be achieved through our *augmentation* technique.

Formally, for an arbitrary matrix A, we use $A_{p,*}$ and $A_{*,q}$ to represent the p-th row and q-th column of A, respectively. Let $F^t \in \mathbb{R}^{H\times\Phi}$ be a $H \times \Phi$-dimensional matrix such that $F^t_{*,\varphi} = f(X_t; \theta_t^\varphi)$ for each $\varphi \in [\Phi]$ and $t \in [T]$. Then the MSE loss function can be rewritten as

$$\ell_t(w_t) = \|F^t w_t - Y_t\|_2^2 .$$

Thus, we have

$$\nabla \ell_t(w_t) \;= (F^t)^\top \big(F^t w_t - Y_t\big) \tag{3}$$

$$\nabla^2 \ell_t(w_t) \;= (F^t)^\top F^t \tag{4}$$

One obstacle here is that the $\Phi \times \Phi$-dimensional matrix $(F^t)^\top F^t$ is not guaranteed to be positive-definite, as F^t may not have the full column rank when the prediction vectors generated by different forecasters are linearly dependent. To address this issue, we develop an augmentation technique that works as follows. At each step t, we concatenate the matrix F^t with a $\Phi \times \Phi$-dimensional scaled identity matrix $\frac{\sigma}{\sqrt{\eta}} I$, and augment Y_t to an $H + \Phi$-dimensional vector by appending Φ zeros when computing the loss. This provides the following augmented system.

$$\tilde{F}^t = \begin{bmatrix} F^t \\ \frac{\sigma}{\sqrt{\eta}} I \end{bmatrix} \in \mathbb{R}^{(H+\Phi)\times\Phi}, \quad \tilde{Y}_t = \begin{bmatrix} Y_t \\ 0 \end{bmatrix} \in \mathbb{R}^{H+\Phi}. \tag{5}$$

Such an augmentation can be viewed as an implementation of Tikhonov regularization Tikhonov & Arsenin (1977), shifting the Hessian eigenvalues by a constant factor and providing a strictly positive lower bound for Λ.

Lemma A.4. *With the augmentation technique, we ensure that $\Lambda \geq \frac{\sigma^2}{\eta}$.*

Proof. By combining Eq. equation 4 with Eq. equation 5, we have

$$\nabla^2 \ell_t(w_t) \;=\; (\tilde{F}^t)^\top \tilde{F}^t \;=\; \begin{bmatrix} (F^t)^\top & \frac{\sigma}{\sqrt{\eta}} I \end{bmatrix} \begin{bmatrix} F^t \\ \frac{\sigma}{\sqrt{\eta}} I \end{bmatrix} \;=\; (F^t)^\top F^t + \frac{\sigma^2}{\eta} I$$

Therefore, for any $t \in [T]$ and $\varphi \in [\Phi]$,

$$\lambda_\varphi\Big(\nabla^2 \ell_t(w_t)\Big) \;=\; \lambda_\varphi\Big((F^t)^\top F^t + \frac{\sigma^2}{\eta} I\Big) \;=\; \lambda_\varphi\Big((F^t)^\top F^t\Big) + \frac{\sigma^2}{\eta}\,,$$

where the last equality follows from the definition of the eigenvalues. Since the matrix $(F^t)^\top F^t$ is positive semi-definite, we always have $\lambda_\varphi\Big((F^t)^\top F^t\Big) \geq 0$. Therefore, this proposition holds. □

Lemma A.5. *The parameter Γ has a constant upper bound $2H\Phi(R_{\max})^2 + \frac{\sigma^2}{\eta}$.*

Proof. Putting Eq. equation 3 and Eq. equation 5 together, we have

$$\begin{aligned}
\nabla \ell_t(w_t) \;&= (\tilde{F}^t)^\top \big(\tilde{F}^t w_t - \tilde{Y}_t\big) \\
&= \begin{bmatrix} (F^t)^\top & \frac{\sigma}{\sqrt{\eta}} I \end{bmatrix} \begin{bmatrix} F^t \\ \frac{\sigma}{\sqrt{\eta}} I \end{bmatrix} w_t - \begin{bmatrix} (F^t)^\top & \frac{\sigma}{\sqrt{\eta}} I \end{bmatrix} \begin{bmatrix} Y_t \\ 0 \end{bmatrix} \\
&= (F^t)^\top F^t w_t + \frac{\sigma^2}{\eta} w_t - (F^t)^\top Y^t\,.
\end{aligned}$$

Using the triangle inequality based on the definition of the norm, we have

$$\begin{aligned}
\|\nabla \ell_t(w_t)\|_2 \;\leq& \|(F^t)^\top F^t w_t\|_2 + \|\frac{\sigma^2}{\eta} w\|_2 + \|(F^t)^\top Y_t\|_2 \\
&\leq \|(F^t)^\top F^t w_t\|_1 + \frac{\sigma^2}{\eta} + \|(F^t)^\top Y_t\|_1\,,
\end{aligned}$$

where the second transition follows from the property that for any vector p, $\|p\|_2 \leq \|p\|_1$ and the fact that $\|w\| = 1$. Let $\mathcal{F}^t = (F^t)^\top F^t$. Then for any $\varphi, \varphi' \in [\Phi]$,

$$\mathcal{F}^t_{\varphi,\varphi'} = F^t_{*,\varphi} \cdot F^t_{*,\varphi'} \leq H \cdot (R_{\max})^2\,.$$

Therefore,

$$\begin{aligned}
\|(F^t)^\top F^t w_t\|_1 \;&= \sum_{\varphi \in [\Phi]} \sum_{\varphi' \in [\Phi]} \mathcal{F}^t_{\varphi,\varphi'} w_t^{\varphi'} \\
&\leq \sum_{\varphi \in [\Phi]} H(R_{\max})^2 \\
&= H\Phi(R_{\max})^2
\end{aligned}$$

where the second transition holds because $w_t \geq 0$ and $\|w_t\|_1 = 1$. It can be proved in a similar way that $\|(F^t)^\top Y_t\|_1 \leq H\Phi(R_{\max})^2$. Therefore, this proposition holds. □

By plugging Lemma A.5 and Lemma A.4 into Lemma A.3, we get the following result.

Theorem A.6. *Through using the augmentation technique and taking the step size $\eta_t = \frac{\eta}{\Lambda t}$, the regret of the OGD algorithm is bounded by $\frac{\eta^2}{\sigma^2}\Big(2H\Phi(R_{\max})^2 + 1\Big)^2 \cdot \log(T+1)$.*

B Related Works

B.1 Transformer-Based Time-Series Forecasting.

Transformer architectures have been extensively adapted to time series forecasting by incorporating temporal structures into self-attention. Informer Zhou et al. (2021) and Pyraformer Liu et al. (2022) improve the modeling efficiency of long-range dependencies through sparse or hierarchical attention. Autoformer Wu et al. (2021a), FEDformer Zhou et al. (2022), and BasisFormer Ni et al. (2023) introduce decomposition or basis representations to disentangle complex temporal components. Other studies focus on multivariate dependencies Zhang & Yan (2022); Liu et al. (2024); Gao et al. (2022); Liang et al. (2023) or frequency-domain characteristics Wu et al. (2024); Ye et al. (2025). These methods advance offline forecasting, but generally employ a fixed temporal modeling architecture after training.

B.2 Masked Attention and Temporal Inductive Bias.

Masked or sparse attention introduces structural constraints into self-attention by selectively restricting pairwise interactions. Position-based methods employ predefined connectivity patterns. LogTrans Li et al. (2019) adopts logarithmically sparse attention, BigBird Zaheer et al. (2020) combines local, random, and global connections, and Longformer Zhang et al. (2021) uses sliding-window attention for long-sequence modeling. For temporal modeling, TimeFormer Liu et al. (2025) incorporates causal constraints with temporal decay, while Powerformer Hegazy et al. (2025) introduces weighted causal attention. Content-dependent approaches determine sparse connectivity according to input representations. Reformer Kitaev et al. (2020) uses locality-sensitive hashing to identify relevant tokens, Informer Zhou et al. (2021) selects informative queries through probabilistic sparse attention, and FTTransformer Huang et al. (2025) incorporates time-frequency masking for temporal representation learning.

These studies demonstrate that restricting attention connectivity can introduce useful structural inductive biases and suppress irrelevant interactions. However, most methods employ a single predefined or input-based masking mechanism within a fixed model. In contrast, MiMe constructs multiple masks with heterogeneous receptive fields and treats them as complementary forecasting strategies for online adaptation.

B.3 Online Time-Series Forecasting.

Existing online forecasting approaches can be broadly grouped into three categories. **Classical statistical models**, such as ARIMA Williams & Hoel (2003) and ETS Holt (2004), naturally support sequential estimation and forecasting, but their fixed parametric structures and distributional assumptions can limit their flexibility under complex concept drift. **Conventional deep forecasting models** can be deployed online through repeated retraining or fine-tuning as new observations arrive. Representative architectures include DLinear Zeng et al. (2023), TimesNet Wu et al. (2023), PatchTST Nie et al. (2023), and iTransformer Liu et al. (2024). **Deep learning models with specially designed online update modules.** FSNet Pham et al. (2023) introduces fast and slow learning components for parameter- and representation-level adaptation. OneNet Zhang et al. (2023) employs Exponentiated Gradient Descent (EGD) Kivinen & Warmuth (1997) to adaptively combine predictive components through online convex optimization, while IPOC Chen et al. (2023a) adopts Follow-the-Perturbed-Leader (FTPL) to aggregate multiple base forecasters under a non-convex online optimization setting.

These methods enable forecasts to respond to newly observed data without relying solely on online parameter updates in Def. 7. However, their predictors and modeling structures are typically predefined, leaving expert temporal inductive biases largely fixed. When experts capture similar dependencies, reweighting offers limited flexibility under changing temporal patterns and may suffer delayed switching after abrupt performance changes. MiMe addresses this limitation by jointly constructing and aggregating diverse experts: relation masks induce heterogeneous receptive fields and distinct temporal inductive biases, which are dynamically combined through online convex optimization. Unlike existing methods that adapt or aggregate predefined predictors, MiMe couples structural expert diversity with online strategy adaptation, enabling efficient switching among complementary temporal patterns within a shared Transformer backbone.

C SYMBOLS AND NOTATIONS

To facilitate reference, Table 5 summarizes the symbols and notations used throughout the paper.

Table 5: Summary of Symbols in MiMe.

Symbol	Definition
x_t	Observation at time step t
$X_t = \{x_{t-L+1}, \ldots, x_t\}$	Historical input window of length L
$Y_t = \{y_1, \ldots, y_H\}$	Ground-truth future series with horizon H
$\hat{Y}_t$	Model prediction at time step t
$\ell(\cdot, \cdot)$	Loss function
$\ell_t(\cdot)$	Loss incurred at time step t
$\mathcal{X} = \{(X^s, Y^s)\}_{s=1}^{S}$	Historical dataset with S samples for offline training
$X^s = \{x_1^s, \ldots, x_L^s\}$	Input series of length L in sample s
$Y^s = \{y_1^s, \ldots, y_H^s\}$	Corresponding future sequence of sample s
θ_0	Model parameters initialized by offline training
f	Online forecaster
f_φ	φ-th masked forecaster
Φ	Number of forecasters / masks
θ_t	Shared backbone parameters at time step t
θ_t^φ	Mask-specific φ head parameters at time step t
$\hat{Y}_t^\varphi$	Prediction produced by forecaster f_φ
w_t^φ	Ensemble weight of forecaster φ at time t
$\mathbf{w}_t \in \mathbb{R}^\Phi$	Weight vector at time step t
$\tilde{Y}_t$	Ensemble prediction
D	Hidden representation dimension
$Z \in \mathbb{R}^{L \times D}$	Embedded input representations
$Q, K, V \in \mathbb{R}^{L \times D}$	Query, Key, and Value matrices
P	Patch length
λ	Patch stride
N	Number of patches
$X_n \in \mathbb{R}^P$	n-th patch
$E_n \in \mathbb{R}^D$	Embedded patch representation
Υ	Number of selected frequencies
ν_κ	Selected dominant frequency, $\kappa = 1, \ldots, \Upsilon$
τ_κ	Dominant period at patch level, $\kappa = 1, \ldots, \Upsilon$
$M \in \mathbb{R}^{N \times N}$	Relation mask
M_φ	Mask used by forecaster f_φ
$\mathcal{M} = \{M_1, \ldots, M_\Phi\}$	Set of all masks
m_{ij}	Mask entry between positions i and j
M_G	Global mask
M_F	Periodic mask
M_O	Contextual (content-based) mask
M_C	Causal mask
M_A	Local window mask
w	Local window radius
η	Learning rate
η_t	Time-dependent step size
$F_t \in \mathbb{R}^{H \times \Phi}$	Prediction matrix at time step t
$\tilde{F}_t$	Augmented prediction matrix
Λ	Strong convexity constant
Γ	Upper bound on gradient norm
σ	Regularization coefficient
$\mathcal{W}$	Feasible set of ensemble weights

D SUMMARY OF DATASETS

Table 6: Summary of datasets used in experiments.

Category	Dataset	Domain	Freq	Var	Len
High Frequency	ETTh1 / h2	Energy (Industrial)	1 hour	7	14,400
	ETTm1 / m2	Energy (Industrial)	15 mins	7	14,400
	Weather	Meteorology	10 mins	21	52,696
High Dimension	Traffic	Transportation	1 hour	862	17,544
Unstable	Exchange	Finance	1 day	8	7,588

E IMPLEMENTATION DETAILS

Input Preprocessing. All input series are normalized using the mean and standard deviation computed from the training set. We adopt the channel-independent strategy, where each variate is processed separately while sharing the same embedding weights and backbone parameters across all variates.

Patching. We segment the input time series into overlapping patches. Given an input length of $L = 720$, patch length $P = 16$, and stride $\lambda = 8$, we pad the input by replicating the last 8 time steps before segmentation to ensure the tail of the sequence is fully covered. This yields an effective length of 728 and produces a total of $N = (728 - 16)/8 + 1 = 90$ patches per variate. Learnable positional encodings are added to the patch embeddings to preserve temporal order information.

Model Architecture. The encoder consists of 2 stacked Transformer encoder layers. Each layer employs a multi-head attention mechanism with 4 heads, where the model dimension is set to 32, and each attention head operates on a dimension of $d_k = d_v = 32/4 = 8$. The FFN within each layer expands the hidden dimension to 128, followed by the GELU activation function. Dropout with a rate of 0.1 is applied to the attention output before the residual connection, as well as to the outputs of both FFN linear layers. Layer normalization is applied in a post-norm manner. After encoding, the patch representations are aggregated and fed into a single linear head, which directly projects to the forecasting horizon $H = \{12, 24\}$.

Training metrics. We use two widely adopted metrics in series forecasting, Mean Absolute Error (MAE) and Mean Squared Error (MSE). Given ground-truth values $\mathbf{y} = \{y_1, \ldots, y_N\}$ and predictions $\hat{\mathbf{y}} = \{\hat{y}_1, \ldots, \hat{y}_N\}$, the Mean Squared Error is computed as:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2.$$

MSE is sensitive to outliers because large prediction errors (e.g., when $|y_i - \hat{y}_i| > 1$) are squared, which amplifies their contribution to the overall loss. However, this property also allows MSE to reflect the dispersion and variance of prediction errors, providing insight into the error distribution.

The Mean Absolute Error is defined as the average absolute difference between predictions and ground-truth values:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|.$$

MAE is robust to outliers since it treats all errors linearly, without squaring. However, this also means it does not capture the variability or distributional characteristics of the prediction errors—only their average magnitude.

Training Configuration. The model is trained using the MSE as the loss function. We employ the AdamW optimizer with an initial learning rate of 1×10^{-4} and weight decay of 1×10^{-4}. The batch size is set to 32. All models are trained for a maximum of 3 epochs with an early stopping strategy applied to prevent overfitting.

F Detailed Experimental Results

This section presents the complete experimental results and additional comparisons that complement the main-text analysis.

F.1 Detailed Ablation Results

Table 7: Ablation study of module combinations with forecasting horizon $H = \{12, 24\}$ on multiple datasets. The best is in **boldface** and the second best is underlined. Config notation: M_G=Global Mask, M_F=Periodic Mask, M_C=Causal Mask, M_O=Contextual Mask, M_A=Local Mask.

Config	Metric	ETTh1			ETTh2			ETTm1			ETTm2			Weather			Traffic			Exchange		
		12	24	Avg	12	24	Avg	12	24	Avg	12	24	Avg	12	24	Avg	12	24	Avg	12	24	Avg
M_G	MSE	1.1330	1.1323	1.1327	3.2483	3.8885	3.5684	0.6494	0.7554	0.7024	0.8403	1.0575	0.9489	0.9920	1.0750	1.0335	0.5017	0.5043	0.5030	0.4917	0.6740	0.5829
	MAE	0.6723	0.6756	0.6739	0.7572	0.8396	0.7984	0.4871	0.5360	0.5115	0.4275	0.4757	0.4516	0.4860	0.5273	0.5066	0.3844	0.3789	0.3817	0.4524	0.5253	0.4888
M_G+M_F	MSE	0.9781	0.9898	0.9839	2.5408	3.1990	2.8699	0.5362	0.6555	0.5958	0.6770	0.8650	0.7710	0.8320	0.9484	0.8902	0.4356	0.4402	0.4379	0.3631	0.5550	0.4591
	MAE	0.6235	0.6295	0.6265	0.6901	0.7818	0.7359	0.4502	0.5040	0.4771	0.3980	0.4463	0.4222	0.4396	0.4919	0.4657	0.3490	0.3442	0.3466	0.3989	0.4803	0.4396
M_G+M_C	MSE	0.9769	0.9885	0.9827	2.4418	3.1232	2.7825	0.5327	0.6546	0.5937	0.6980	0.8887	0.7933	0.8233	0.9409	0.8821	0.4351	0.4399	0.4375	0.3715	0.5413	0.4564
	MAE	0.6226	0.6283	0.6254	0.6796	0.7741	0.7268	0.4481	0.5036	0.4758	0.4011	0.4511	0.4261	0.4366	0.4894	0.4630	0.3487	0.3441	0.3464	0.4025	0.4716	0.4371
M_G+M_O	MSE	0.9767	0.9813	0.9790	2.5844	3.3030	2.9437	0.5288	0.6481	0.5885	0.6575	0.8161	0.7368	0.7891	0.9367	0.8629	0.4343	0.4399	0.4371	0.3746	0.5512	0.4629
	MAE	0.6222	0.6255	0.6239	0.6902	0.7806	0.7354	0.4461	0.5015	0.4738	0.3918	0.4358	0.4138	0.4262	0.4875	0.4569	0.3470	0.3441	0.3455	0.4058	0.4819	0.4439
M_G+M_A	MSE	0.9798	0.9890	0.9844	2.5871	3.3049	2.9460	0.5366	0.6551	0.5958	0.6831	0.8574	0.7703	0.8413	0.9480	0.8947	0.4355	0.4399	0.4377	0.3663	0.5521	0.4592
	MAE	0.6239	0.6290	0.6265	0.6955	0.7902	0.7429	0.4504	0.5037	0.4770	0.4004	0.4465	0.4235	0.4409	0.4914	0.4661	0.3490	0.3440	0.3465	0.4011	0.4801	0.4406
M_G+M_F+M_C	MSE	0.9316	0.9367	0.9341	2.2425	2.9533	2.5979	0.4942	0.6176	0.5559	0.6417	0.8295	0.7356	0.7827	0.8977	0.8402	0.4132	0.4201	0.4166	0.3364	0.4971	0.4167
	MAE	0.6065	0.6095	0.6080	0.6616	0.7541	0.7078	0.4329	0.4904	0.4616	0.3882	0.4398	0.4140	0.4216	0.4737	0.4477	0.3335	0.3302	0.3319	0.3857	0.4520	0.4188
M_G+M_F+M_O	MSE	0.9238	0.9385	0.9312	2.3458	3.0209	2.6833	0.4945	0.6166	0.5556	0.6406	0.7887	0.7147	0.7669	0.8970	0.8319	0.4134	0.4195	0.4164	0.3286	0.5109	0.4197
	MAE	0.6049	0.6102	0.6076	0.6654	0.7611	0.7132	0.4333	0.4889	0.4611	0.3860	0.4317	0.4088	0.4161	0.4733	0.4447	0.3331	0.3293	0.3312	0.3816	0.4616	0.4216
M_G+M_F+M_A	MSE	0.9334	0.9383	0.9358	2.3923	3.0693	2.7308	0.4978	0.6198	0.5588	0.6169	0.8307	0.7238	0.7857	0.9006	0.8432	0.4132	0.4202	0.4167	0.3335	0.5008	0.4171
	MAE	0.6075	0.6104	0.6089	0.6780	0.7620	0.7200	0.4349	0.4910	0.4629	0.3857	0.4405	0.4131	0.4230	0.4745	0.4488	0.3336	0.3302	0.3319	0.3848	0.4571	0.4210
M_G+M_F+M_C +M_O	MSE	0.9076	0.9179	0.9127	2.2219	2.8583	2.5401	0.4774	0.5973	0.5374	0.6029	0.7704	0.6867	0.7508	0.8732	0.8120	0.4032	0.4091	0.4061	0.3148	0.4764	0.3956
	MAE	0.5973	0.6014	0.5993	0.6515	0.7472	0.6993	0.4251	0.4824	0.4538	0.3801	0.4259	0.4030	0.4080	0.4632	0.4356	0.3250	0.3211	0.3230	0.3713	0.4448	0.4081
M_G+M_F+M_C +M_A	MSE	0.9072	0.9205	0.9139	2.1742	2.9044	2.5393	0.4765	0.6025	0.5395	0.6216	0.7705	0.6960	0.7587	0.8796	0.8191	0.4033	0.4085	0.4059	0.3132	0.4780	0.3956
	MAE	0.5972	0.6013	0.5992	0.6512	0.7411	0.6962	0.4251	0.4841	0.4546	0.3825	0.4288	0.4056	0.4111	0.4658	0.4384	0.3253	0.3212	0.3232	0.3711	0.4455	0.4083
M_G+M_F+M_O +M_A	MSE	0.9089	0.9165	0.9127	2.3154	2.9145	2.6150	0.4770	0.5988	0.5379	0.5806	0.7643	0.6725	0.7519	0.8764	0.8141	0.4035	0.4088	0.4062	0.3113	0.4839	0.3976
	MAE	0.5982	0.6017	0.6000	0.6611	0.7511	0.7061	0.4250	0.4828	0.4539	0.3774	0.4252	0.4013	0.4082	0.4641	0.4362	0.3249	0.3212	0.3230	0.3715	0.4505	0.4110
M_G+M_C+M_O +M_A	MSE	0.9084	0.9144	0.9114	2.2315	2.8791	2.5553	0.4743	0.5956	0.5350	0.5936	0.7780	0.6858	0.7479	0.8728	0.8103	0.4034	0.4087	0.4060	0.3162	0.4754	0.3958
	MAE	0.5980	0.6008	0.5994	0.6540	0.7490	0.7015	0.4235	0.4815	0.4525	0.3785	0.4271	0.4028	0.4066	0.4626	0.4346	0.3248	0.3212	0.3230	0.3738	0.4444	0.4091
M_G+M_F+M_C +M_O+M_A	MSE	**0.8929**	**0.8870**	**0.8899**	**1.7455**	**2.2540**	**1.9997**	**0.4461**	**0.5677**	**0.5069**	**0.5331**	**0.6865**	**0.6098**	**0.7158**	**0.8295**	**0.7727**	**0.3962**	**0.4026**	**0.3994**	**0.3045**	**0.4725**	**0.3885**
	MAE	**0.5883**	**0.5881**	**0.5882**	**0.5927**	**0.6745**	**0.6336**	**0.4084**	**0.4687**	**0.4386**	**0.3677**	**0.4121**	**0.3899**	**0.3883**	**0.4405**	**0.4144**	**0.3195**	**0.3156**	**0.3176**	**0.3660**	**0.4417**	**0.4039**

F.2 Detailed Results of Expert Diversity Comparison

Table 8: Expert diversity comparison with different forecasting horizon $H = \{12, 24\}$ across multiple datasets.The best is in **boldface** and the second best is underlined.

Config	Metric	ETTh1			ETTh2			ETTm1			ETTm2			Weather			Traffic			Exchange		
		12	24	Avg	12	24	Avg	12	24	Avg	12	24	Avg	12	24	Avg	12	24	Avg	12	24	Avg
5×M_G	MSE	0.901	0.888	0.895	1.982	2.560	2.271	0.461	0.579	0.520	0.553	0.721	0.637	0.729	0.849	0.789	0.413	0.418	0.415	0.313	0.509	0.411
	MAE	0.592	0.589	0.590	0.625	0.701	0.663	0.418	0.474	0.446	0.378	0.424	0.401	0.394	0.451	0.423	0.332	0.327	0.329	0.372	0.462	0.417
5×M_F	MSE	0.897	0.890	0.893	1.923	2.501	2.212	0.460	0.585	0.522	0.565	0.729	0.647	0.731	0.854	0.793	0.413	0.419	0.416	0.313	0.519	0.416
	MAE	0.591	0.590	0.590	0.615	0.701	0.658	0.418	0.477	0.448	0.373	0.423	0.398	0.398	0.454	0.426	0.332	0.327	0.330	0.372	0.465	0.418
5×M_C	MSE	0.893	**0.886**	0.890	1.750	2.261	2.006	0.463	0.583	0.523	0.615	0.771	0.693	0.725	0.841	0.783	0.412	0.418	0.415	0.335	0.502	0.418
	MAE	0.589	**0.587**	0.588	0.600	0.699	0.649	0.420	0.477	0.449	0.381	0.431	0.406	0.401	0.449	0.425	0.332	0.327	0.329	0.384	0.455	0.420
5×M_O	MSE	**0.880**	0.889	**0.885**	1.930	2.478	2.204	0.451	0.569	0.510	0.538	0.720	0.629	**0.709**	0.834	**0.771**	0.423	0.425	0.424	0.321	0.522	0.421
	MAE	**0.587**	0.589	**0.588**	0.606	0.693	0.650	0.415	0.475	0.445	**0.363**	**0.412**	**0.387**	0.398	0.447	0.422	0.343	0.333	0.338	0.378	0.469	0.423
5×M_A	MSE	0.897	0.889	0.893	1.817	2.290	2.054	0.462	0.595	0.529	0.565	0.737	0.651	0.721	0.844	0.783	0.414	0.419	0.416	0.310	0.541	0.426
	MAE	0.592	0.589	0.591	0.604	0.679	0.641	0.421	0.484	0.452	0.372	0.427	0.399	0.397	0.450	0.423	0.333	0.328	0.330	0.371	0.476	0.423
MiMe	MSE	0.893	0.887	0.890	**1.745**	**2.254**	**2.000**	**0.446**	**0.568**	**0.507**	**0.533**	**0.686**	**0.610**	0.716	**0.830**	0.773	**0.396**	**0.403**	**0.399**	**0.304**	**0.472**	**0.388**
	MAE	0.588	0.588	0.588	**0.593**	**0.674**	**0.634**	**0.408**	**0.469**	**0.439**	0.368	0.412	0.390	**0.388**	**0.440**	**0.414**	**0.320**	**0.316**	**0.318**	**0.366**	**0.442**	**0.404**

F.3 Detailed Results of Ensemble Method Comparison

Table 9: Ensemble method comparison averaged over forecasting horizon $H = \{12, 24\}$. The best is in **boldface** and the second best is underlined.

Ensemble Method	Metric	ETTh1			ETTh2			ETTm1			ETTm2			Weather			Traffic			Exchange		
		12	24	Avg	12	24	Avg	12	24	Avg	12	24	Avg	12	24	Avg	12	24	Avg	12	24	Avg
Custom	MSE	0.893	0.895	0.894	1.852	2.445	2.148	0.457	0.577	0.517	0.574	0.717	0.645	0.725	0.845	0.785	0.404	0.408	0.406	0.340	0.547	0.444
	MAE	0.589	0.592	0.591	0.610	0.692	0.651	0.414	0.472	0.443	0.375	0.417	0.396	0.393	0.449	0.421	0.325	0.320	0.323	0.389	0.479	0.434
$FTPL_{Exp}$	MSE	1.081	1.066	1.074	2.564	3.006	2.785	0.598	0.717	0.658	0.753	0.939	0.846	0.925	1.000	0.963	0.499	0.503	0.501	0.496	0.659	0.578
	MAE	0.650	0.650	0.650	0.680	0.762	0.721	0.459	0.515	0.487	0.408	0.452	0.430	0.449	0.493	0.471	0.383	0.377	0.380	0.452	0.518	0.485
$FTPL_{Gumbel}$	MSE	1.078	1.066	1.072	2.456	3.034	2.745	0.596	0.713	0.655	0.803	0.930	0.866	0.922	1.002	0.962	0.500	0.504	0.502	0.492	0.654	0.573
	MAE	0.650	0.651	0.650	0.676	0.759	0.718	0.458	0.514	0.486	0.410	0.450	0.430	0.449	0.493	0.471	0.383	0.377	0.380	0.453	0.516	0.485
EGD	MSE	0.893	0.895	0.894	1.898	2.449	2.174	0.457	0.577	0.517	0.581	0.718	0.649	0.725	0.845	0.785	0.404	0.408	0.406	0.340	0.547	0.443
	MAE	0.589	0.592	0.591	0.616	0.693	0.654	0.414	0.472	0.443	0.375	0.417	0.396	0.393	0.449	0.421	0.325	0.320	0.323	0.389	0.479	0.434
OGD	MSE	**0.893**	**0.887**	**0.890**	**1.745**	**2.254**	**2.000**	**0.446**	**0.568**	**0.507**	**0.533**	**0.686**	**0.610**	**0.716**	**0.830**	**0.773**	**0.396**	**0.403**	**0.399**	**0.304**	**0.472**	**0.388**
	MAE	**0.588**	**0.588**	**0.588**	**0.593**	**0.674**	**0.634**	**0.408**	**0.469**	**0.439**	**0.368**	**0.412**	**0.390**	**0.388**	**0.440**	**0.414**	**0.320**	**0.316**	**0.318**	**0.366**	**0.442**	**0.404**

G More Discussion

RQ4: Can backbone sharing improve efficiency while preserving forecasting performance?

To evaluate the effectiveness of backbone sharing, we compare MiMe with an independent-backbone variant, where each masked forecaster uses a separate Transformer while the online ensembling procedure remains unchanged. As shown in Tab. 10, the shared design reduces the parameter count from 0.303M to 0.199M, corresponding to a 34.44% reduction. Despite the substantially lower model complexity, it does not sacrifice forecasting accuracy. Instead, the shared backbone consistently achieves lower MSE across all seven datasets and comparable or lower MAE. On average, backbone sharing reduces MSE and MAE by approximately 2.41% and 1.62%, respectively.

Table 10: Performance and efficiency comparison between shared and independent backbones. Metrics are averaged over forecasting horizon $H = \{12, 24\}$. Red indicates relative forecasting improvement, while green indicates parameter reduction.

Metric	Backbone	ETTh1	ETTh2	ETTm1	ETTm2	Weather	Traffic	Exchange
MSE	Shared	0.890	2.000	0.507	0.610	0.773	0.399	0.388
	Independent	0.910	2.081	0.520	0.621	0.783	0.412	0.397
	Improvement	↑ 2.22%	↑ 3.90%	↑ 2.47%	↑ 1.78%	↑ 1.26%	↑ 3.10%	↑ 2.17%
MAE	Shared	0.588	0.634	0.439	0.390	0.414	0.318	0.404
	Independent	0.600	0.635	0.446	0.390	0.425	0.327	0.412
	Improvement	↑ 2.01%	↑ 0.28%	↑ 1.60%	↑ 0.06%	↑ 2.45%	↑ 2.88%	↑ 2.09%
Params (M)	Shared	0.199	0.199	0.199	0.199	0.199	0.199	0.199
	Independent	0.303	0.303	0.303	0.303	0.303	0.303	0.303
	Reduction	↓ 34.44%	↓ 34.44%	↓ 34.44%	↓ 34.44%	↓ 34.44%	↓ 34.44%	↓ 34.44%

These results indicate that multi-mask strategies do not require independent Transformer backbones. Sharing backbone effectively removes parameter redundancy while preserving mask-specific inductive biases, yielding a favorable balance between forecasting performance and model efficiency.

RQ5: How does MiMe adapt its ensemble weights to evolving temporal patterns?

As shown in Fig. 3, to examine the strategy-level adaptation of MiMe, we select time series exhibiting evident pattern changes and visualize the online ensemble weights of the five masked forecasters.

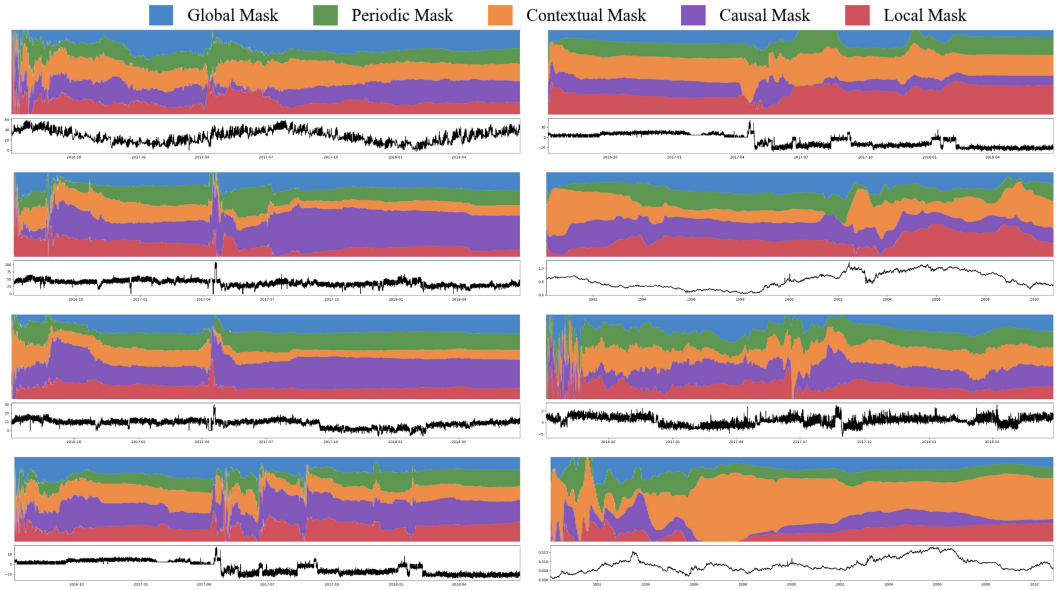


Figure 3: Evolution of ensemble weights on representative time series with evident regime changes.

The ensemble weights are clearly time-varying rather than converging to a fixed combination. When the underlying series undergoes noticeable changes in level, trend, or fluctuation pattern, MiMe

correspondingly reallocates weights among the heterogeneous masked forecasters, whereas the weights evolve more smoothly during relatively stable periods. Moreover, different series exhibit distinct dominant experts and weight-transition patterns, indicating that no single receptive-field bias is consistently preferred across heterogeneous streams. These observations provide qualitative evidence that MiMe dynamically adjusts the contribution of different temporal inductive biases according to sequential feedback, enabling strategy-level adaptation to evolving temporal patterns.

RQ6: Can MiMe rapidly adapt to distribution shifts across datasets?

To further evaluate the adaptability of online forecasters under distribution shifts, we conduct online transfer experiments in which models are first trained on a source dataset and then continuously updated on a target stream. We consider two representative transfer scenarios:

(1) Cross-object transfer. We transfer models from ETTh1 to ETTh2 and from ETTm1 to ETTm2, where the sampling frequency remains unchanged while the monitored transformer unit changes. This setting evaluates whether the model can reuse learned temporal representations while adapting to object-specific distribution shifts.

(2) Cross-frequency transfer. We transfer ETTh1 to ETTm1 and ETTh2 to ETTm2, corresponding to the same transformer unit observed at different temporal resolutions. This setting introduces a substantial change in temporal granularity, from hourly to 15-minute observations, and therefore evaluates the model's adaptability to frequency-induced distribution shifts.

Table 11: Transfer forecasting results with different forecasting horizon $H = \{12, 24\}$ on multiple transfer directions. The best is in **boldface** and the second best is underlined.

Model	Metric	ETTh1→ETTh2			ETTm1→ETTm2			ETTh1→ETTm1			ETTh2→ETTm2		
		12	24	Avg	12	24	Avg	12	24	Avg	12	24	Avg
FEDformer	MSE	8.184	8.364	8.274	2.746	2.820	2.783	1.328	1.534	1.431	2.752	2.829	2.790
	MAE	1.101	1.115	1.108	0.733	0.740	0.737	0.769	0.824	0.796	0.737	0.747	0.742
Pyraformer	MSE	12.999	15.174	14.087	9.475	11.039	10.257	0.897	1.462	1.179	9.537	11.295	10.416
	MAE	1.381	1.701	1.541	0.895	1.071	0.983	0.621	0.788	0.704	0.915	1.100	1.008
Crossformer	MSE	12.337	13.717	13.027	9.117	10.280	9.699	1.228	1.442	1.335	9.288	10.542	9.915
	MAE	1.314	1.452	1.383	0.932	1.017	0.974	0.727	0.782	0.754	0.968	1.084	1.026
iTransformer	MSE	2.991	4.218	3.605	0.707	1.004	0.855	0.549	0.747	0.648	0.800	1.196	0.998
	MAE	0.710	0.799	0.754	0.421	0.476	0.448	0.461	0.540	0.501	0.446	0.515	0.480
PatchTST	MSE	3.521	4.294	3.907	0.984	1.134	1.059	0.692	0.822	0.757	1.043	1.245	1.144
	MAE	0.784	0.881	0.833	0.457	0.503	0.480	0.515	0.571	0.543	0.477	0.528	0.502
LightTS	MSE	2.367	3.341	2.854	0.671	0.765	0.718	0.544	0.717	0.630	0.662	0.911	0.787
	MAE	0.735	0.857	0.796	0.466	0.483	0.475	0.476	0.555	0.515	0.464	0.525	0.494
TSMixer	MSE	2.299	3.383	2.841	0.699	1.067	0.883	0.515	0.671	0.593	0.744	1.091	0.918
	MAE	0.829	0.970	0.899	0.492	0.581	0.536	0.472	0.541	0.506	0.511	0.593	0.552
TimeMixer	MSE	2.931	5.274	4.103	0.616	0.893	0.754	**0.440**	0.622	0.531	0.623	0.979	0.801
	MAE	0.747	0.892	0.820	0.413	0.473	0.443	**0.420**	0.512	0.466	0.419	0.505	0.462
WPMixer	MSE	2.039	2.918	2.479	**0.565**	0.755	0.660	0.459	**0.587**	**0.523**	**0.579**	0.804	**0.692**
	MAE	0.638	0.747	0.693	0.394	0.439	0.416	0.427	0.493	0.460	0.397	0.459	0.428
PAttn	MSE	2.758	3.516	3.137	0.707	0.969	0.838	0.498	0.642	0.570	0.817	1.069	0.943
	MAE	0.735	0.838	0.787	0.400	0.460	0.430	0.445	0.507	0.476	0.431	0.497	0.464
TimesNet	MSE	3.087	4.035	3.561	1.022	1.348	1.185	0.693	0.777	0.735	1.087	1.594	1.341
	MAE	0.859	0.924	0.891	0.554	0.551	0.553	0.527	0.554	0.541	0.582	0.630	0.606
FSNet	MSE	2.895	6.272	4.583	0.694	1.183	0.939	0.536	0.772	0.654	0.904	1.843	1.373
	MAE	0.862	1.163	1.013	0.494	0.628	0.561	0.473	0.597	0.535	0.533	0.700	0.617
Time-FSNet	MSE	10.582	12.425	11.503	8.332	8.806	8.569	1.473	1.761	1.617	8.067	8.521	8.294
	MAE	1.518	1.595	1.557	1.234	1.310	1.272	0.838	0.917	0.877	1.190	1.247	1.218
OneNet	MSE	2.189	3.449	2.819	0.591	0.810	0.700	0.530	0.749	0.640	0.620	1.059	0.839
	MAE	0.693	0.824	0.759	0.427	0.475	0.451	0.457	0.556	0.507	0.439	0.537	0.488
MiMe	MSE	**1.884**	**2.613**	**2.249**	0.593	**0.721**	**0.657**	0.477	0.596	0.537	0.610	**0.795**	0.703
	MAE	**0.611**	**0.713**	**0.662**	**0.376**	**0.421**	**0.398**	0.429	**0.486**	**0.458**	**0.389**	**0.440**	**0.414**

As shown in Tab. 11. MiMe exhibits consistently strong transfer performance under both scenarios. For cross-object transfer, MiMe achieves the best averaged MSE and MAE on both ETTh1→ETTh2 and ETTm1→ETTm2. Under the more substantial cross-frequency shifts, MiMe remains highly competitive: it achieves the best averaged MAE on both transfer directions, while ranking among the top-performing methods in terms of MSE. Notably, MiMe obtains the best averaged MAE across all four transfer directions, indicating stable adaptation despite changes in either the monitored object or temporal resolution.

Compared with existing online forecasting methods, the advantage is also consistent. MiMe outperforms FSNet, Time-FSNet, and OneNet in both averaged MSE and MAE across all four transfer directions. In particular, compared with OneNet, the strongest of these online baselines, MiMe reduces the average MSE and MAE across the four transfer settings by approximately 17.0% and 12.4%, respectively. These results suggest that the heterogeneous receptive fields and online ensemble mechanism of MiMe provide a flexible strategy space for adapting to previously unseen temporal distributions rather than relying on a fixed temporal inductive bias.

RQ7: How sensitive is MiMe to its hyperparameters?

We investigate the sensitivity of MiMe to three key hyperparameters: the patch length P in patch embedding, and the step-size constant η and regularization coefficient σ in online ensembling.

(1) Effect of Patch Length

The patch length P determines the temporal granularity of input tokens and the effective temporal span represented by each patch, thereby affecting the granularity at which relation masks constrain receptive fields. We vary $P \in \{16, 32, 64, 128\}$ while keeping the other settings fixed, with results reported in Tab. 12.

Table 12: Effect of patch length P on forecasting performance (MSE/MAE) under prediction horizons $H \in \{12, 24\}$.

Dataset	Pred Len	Patch=16		Patch=32		Patch=64		Patch=128	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	12	0.893	0.588	**0.847**	**0.573**	0.880	0.591	0.865	0.585
	24	0.888	0.589	**0.845**	**0.572**	0.863	0.582	0.855	0.580
	avg	0.890	0.588	**0.846**	**0.572**	0.871	0.587	0.860	0.583
ETTh2	12	1.745	0.593	1.733	0.572	1.791	0.571	**1.697**	**0.559**
	24	2.271	0.675	2.374	0.657	2.373	0.622	**2.143**	**0.606**
	avg	2.008	0.634	2.053	0.614	2.082	0.597	**1.920**	**0.583**
ETTm1	12	0.447	0.409	**0.423**	0.391	0.427	0.393	0.425	**0.390**
	24	0.568	0.469	0.549	**0.457**	0.552	0.462	**0.541**	0.461
	avg	0.507	0.439	0.486	**0.424**	0.489	0.427	**0.483**	0.426
ETTm2	12	0.538	0.369	0.491	0.357	0.487	0.355	**0.471**	**0.350**
	24	0.690	0.414	0.634	0.398	0.640	0.396	**0.625**	**0.394**
	avg	0.614	0.392	0.563	0.377	0.563	0.376	**0.548**	**0.372**
Weather	12	0.716	0.388	0.646	0.358	0.641	0.354	**0.641**	**0.351**
	24	0.830	0.441	0.807	0.428	**0.798**	**0.425**	0.805	0.428
	avg	0.773	0.415	0.727	0.393	**0.720**	**0.389**	0.723	0.389
Traffic	12	0.396	0.320	**0.389**	**0.309**	0.401	0.316	0.422	0.330
	24	0.403	0.316	**0.403**	**0.311**	0.415	0.318	0.440	0.335
	avg	0.400	0.318	**0.396**	**0.310**	0.408	0.317	0.431	0.333
Exchange	12	**0.304**	**0.366**	0.329	0.375	0.440	0.432	0.462	0.436
	24	**0.476**	**0.443**	0.484	0.447	0.557	0.476	0.573	0.482
	avg	**0.390**	**0.405**	0.406	0.411	0.498	0.454	0.517	0.459

The optimal patch length varies across datasets. Moderate patch lengths perform better on ETTh1 and Traffic, with $P = 32$ achieving the lowest average errors, whereas larger patches are preferred on ETTh2 and ETTm2. Weather also benefits from relatively coarse temporal granularity, achieving its best performance around $P = 64$–128. In contrast, Exchange favors the finest granularity ($P = 16$), and its performance deteriorates noticeably as the patch length increases. ETTm1 remains relatively stable across $P = 32$ to 128.

These results indicate that no single temporal granularity is universally optimal across heterogeneous time series. Fine-grained patches better preserve rapidly varying local information, while larger patches provide more aggregated temporal context. The appropriate trade-off therefore depends on the temporal characteristics of each dataset. Notably, MiMe uses a fixed patch length in the main experiments rather than dataset-specific tuning, while the sensitivity results suggest that further gains can be obtained by adapting the input granularity to individual data streams.

(2) Effect of η and σ

We examine the two hyperparameters involved in online ensembling. Specifically, we vary η and σ independently over $\{10, 5, 1, 0.5, 0.1, 0.05, 0.01, 0.001\}$ while keeping the other hyperparameters fixed. The results are shown in Fig. 4.

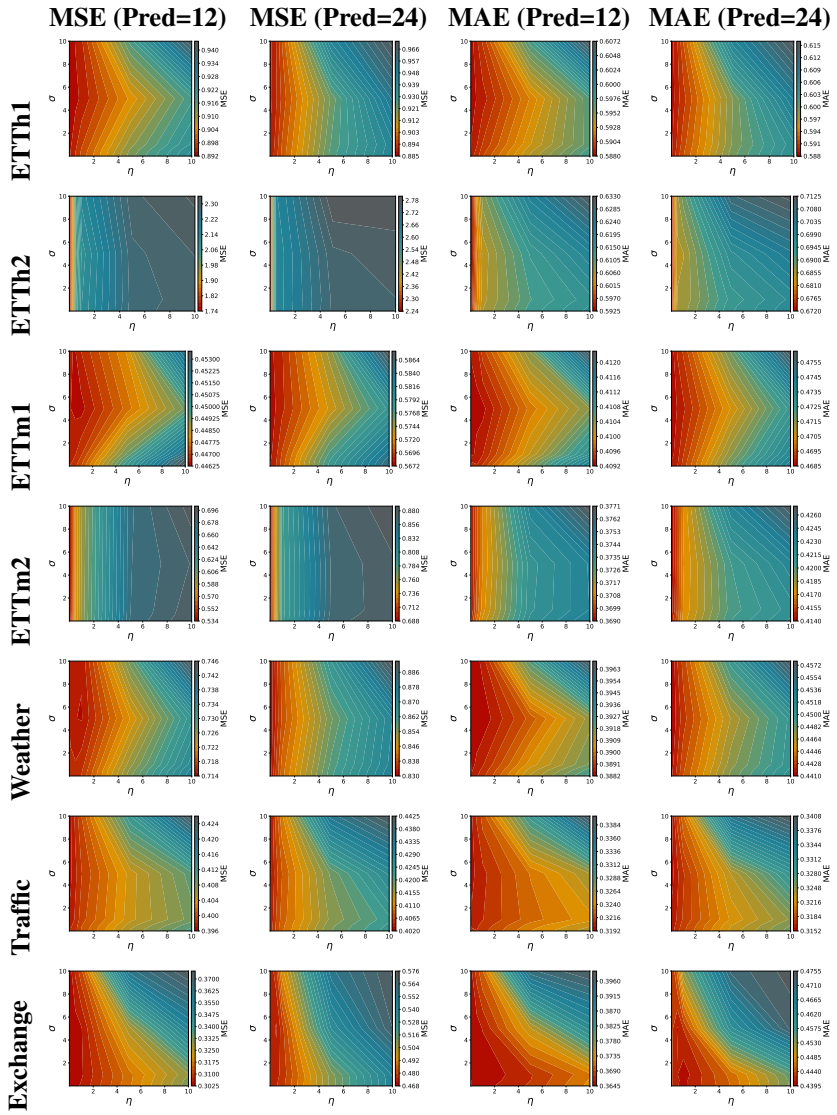


Figure 4: MSE and MAE results over η and λ across different datasets and prediction lengths.

Overall, MiMe is more sensitive to the step-size constant η than to the regularization coefficient σ. In most cases, smaller η leads to lower MSE and MAE, indicating that conservative weight updates are preferable for stable online ensembling. By contrast, performance varies more moderately with σ, suggesting that the regularization term has a limited influence on forecasting accuracy within a broad range of values. Notably, increasing σ generally does not degrade performance and often yields slight improvements. This observation indicates that the regularization introduced to ensure strong convexity for the regret analysis can be incorporated without compromising empirical forecasting performance.