

Structure-Aware Behavioral Alignment of Large Language Models for Robust Time Series Forecasting

Mingliang Hu

hum11993@163.com

Jiangsu Electronic Information Vocational College

Research Article

Keywords: Software-defined networking, Dynamic traffic engineering, Implicit graph dynamics, Spatiotemporal learning, Congestion control, Risk-aware routing

Posted Date: October 14th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11151674/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Structure-Aware Behavioral Alignment of Large Language Models for Robust Time Series Forecasting

Anonymous Author^{1,2*}, Anonymous Author^{2,3†} and
Anonymous Author^{1,2†}

^{1*}Department of [Anonymous], [Anonymous University], [Anonymous],
[Anonymous], 000000, [Anonymous], [Anonymous].

²Department of [Anonymous], [Anonymous University], [Anonymous],
[Anonymous], 000000, [Anonymous], [Anonymous].

³Department of [Anonymous], [Anonymous University], [Anonymous],
[Anonymous], 000000, [Anonymous], [Anonymous].

*Corresponding author(s). E-mail(s): anonymous@review.com;
Contributing authors: anonymous@review.com; anonymous@review.com;

[†]These authors contributed equally to this work.

Abstract

Large language models (LLMs) have recently emerged as a promising backbone for time-series forecasting, yet most existing LLM-based forecasters are still optimized with purely supervised objectives such as MSE or MAE. Such point-wise training is often misaligned with the sequence-level properties that matter in real deployments, including waveform fidelity, trend consistency, and robustness under distribution shift. Reinforcement learning (RL) is a natural candidate for sequence-level optimization, but directly transplanting RL recipes from natural language processing to time series is difficult because forecasting involves continuous outputs, long-horizon credit assignment, multi-objective rewards, and unstable policy updates. To address these issues, we propose **TSLLM-BO**, a two-stage framework that first learns a strong supervised forecaster and then performs structure-aware behavioral alignment through reinforcement learning. The framework combines three key ideas: 1) a time-text alignment architecture that links temporal patches with semantic prompts in a frozen LLM backbone; 2) a differentiable composite reward that jointly optimizes point accuracy, waveform consistency, and trend alignment; and 3) **Dynamic Policy Clipping (DPC)**, which adapts the PPO

trust region over training to stabilize long-horizon optimization. Extensive experiments on standard long-horizon forecasting benchmarks show that TSLLM-BO consistently outperforms strong LLM-based and deep forecasting baselines, especially under long prediction horizons, cross-dataset transfer, and noise/missingness perturbations. These results suggest that behavioral alignment, rather than supervised fitting alone, is a practical route for endowing LLM-based forecasters with stronger temporal structure awareness and deployment robustness.

Keywords: Software-defined networking; Dynamic traffic engineering; Implicit graph dynamics; Spatiotemporal learning; Congestion control; Risk-aware routing

1 Introduction

Time series forecasting is a core problem in modern data-centric systems and is particularly prominent in intelligent manufacturing, where production lines, industrial equipment, and quality-control systems continuously generate high-frequency, heterogeneous sensor streams. Accurate forecasting supports predictive maintenance, production scheduling, energy optimization, anomaly anticipation, and closed-loop process control; however, frequent operating-mode changes, non-stationarity, noise, and multiscale temporal dependencies make reliable prediction especially challenging in such settings [1]. Similar requirements arise in traffic control, grid dispatch, disease surveillance, and financial monitoring, motivating a broad family of deep forecasting architectures based on hierarchical interpolation, inverted-variable modeling, linear decomposition, patch-wise Transformers, two-dimensional temporal variation, and self-supervised representation learning [2–7]. More broadly, temporally evolving measurements also arise outside canonical forecasting benchmarks, including biological studies of aging and oxidative-stress responses and developmental transitions [8, 9]. Across these domains, the practical value of a forecasting model is determined not only by low point-wise error, but also by whether its predictions preserve temporal structure: trend turning points should occur at the right time, waveform morphology should remain realistic, and long-horizon forecasts should degrade gracefully under non-stationarity and noise.

Recently, large language models (LLMs) have attracted substantial interest in time-series analysis because their pretrained representations provide a powerful prior for abstraction, cross-domain transfer, and long-context reasoning [10, 11]. Representative studies include Time-LLM, which reprograms temporal inputs for language-model backbones [12], AutoTimes for autoregressive forecasting with LLMs [13], One-Fits-All for pretrained-language-model-based general time-series analysis [14], CALF for cross-modal fine-tuning [15], and zero-shot LLM forecasting [16]. Probabilistic ensemble forecasting provides a complementary non-LLM direction [17]. However, the dominant optimization paradigm in LLM-based forecasting remains supervised fine-tuning (SFT), typically driven by point-wise objectives such as MSE or MAE [18–20]. While effective for local numerical fitting, such objectives are only a partial surrogate for forecasting quality. They do not explicitly encourage shape preservation, temporal coherence, or

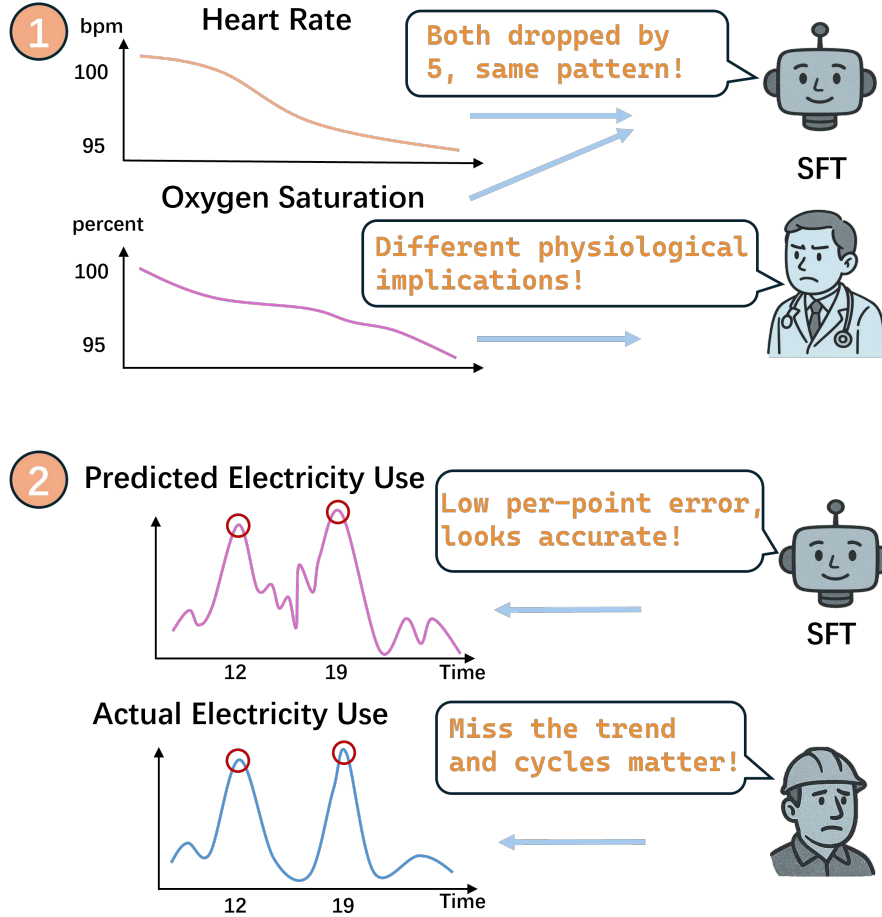


Fig. 1 Illustration of key limitations in SFT-only modeling. 1. The model treats heart rate (bpm) and oxygen saturation (%) as having similar numeric trends, without understanding their different physiological implications. 2. In electricity forecasting, SFT focuses only on per-point errors while ignoring the overall temporal structure—missing real-world patterns such as daily trends and cycles.

regime-aware behavior, and they often produce models that look accurate on average yet fail in structurally important scenarios, as illustrated in Fig. 1. Recent analyses further highlight limitations arising from modality alignment and from weaknesses of language models in low-level symbolic manipulation, reinforcing the need for forecasting-specific alignment mechanisms [21, 22].

A natural way to move beyond point-wise supervision is to optimize forecasting behavior at the sequence level. Reinforcement learning provides such a mechanism: instead of fitting only labels, it can align predictions with task-level objectives that better reflect downstream utility. In language modeling, reinforcement learning from human or AI feedback and related preference-optimization paradigms have established a general recipe for optimizing behaviors beyond next-token likelihood [23–27].

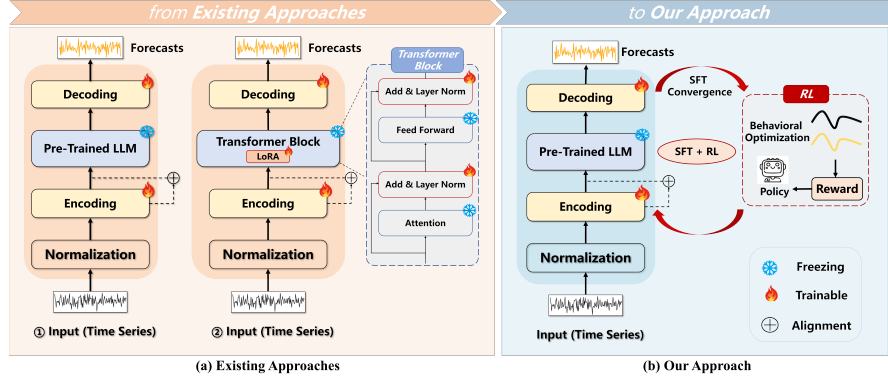


Fig. 2 Comparison between prevailing LLM-based forecasting pipelines and our framework. (a) Most existing methods optimize LLM forecasters only with supervised fine-tuning. (b) TSLLM-BO adopts a two-stage strategy: it first learns a strong supervised initializer and then performs reinforcement learning for structure-aware behavioral alignment, improving temporal consistency and robustness.

More recent reasoning-oriented work has explored reinforced mathematical reasoning, REINFORCE-style optimization, adaptive rollback, early-exit policy learning, and iterative reasoning in retrieval-augmented multimodal systems [28–33]. Yet applying these ideas to time-series forecasting is substantially harder than applying them to text generation. Forecasting involves continuous-valued outputs, long-horizon credit assignment, offline or semi-offline datasets, and multiple competing objectives such as point accuracy, smoothness, phase alignment, and trend consistency. Although PPO provides a practical clipped-policy optimization backbone [34], naively importing standard RL recipes from NLP often yields poorly calibrated rewards, unstable variance, or brittle policy updates when the horizon becomes long and the temporal distribution shifts.

To address this gap, we propose **TSLLM-BO** (Time Series Large Language Model & Behavioral Optimization), a structure-aware behavioral alignment framework for LLM-based forecasting. TSLLM-BO first learns a reliable supervised forecaster, then refines it with reinforcement learning using a composite temporal reward and an adaptive policy-control mechanism. Concretely, the method combines a time-text alignment architecture, a differentiable reward that jointly models point error, waveform similarity, and trend alignment, and a *Dynamic Policy Clipping* (DPC) strategy that adjusts the PPO trust region throughout training. The overall training philosophy is shown in Fig. 2: supervised learning provides a strong initialization, while RL performs sequence-level alignment that conventional regression losses cannot express.

Our main contributions are summarized as follows:

- **A behavioral-alignment perspective for LLM forecasting.** We formulate LLM-based time-series forecasting as a two-stage optimization problem in which supervised learning provides a reliable initializer and reinforcement learning subsequently aligns predictions with sequence-level temporal structure.
- **A structure-aware temporal reward.** We design a differentiable composite reward that jointly captures point accuracy, waveform consistency, and trend

alignment, allowing forecasting behavior to be optimized beyond local regression error.

- **Dynamic Policy Clipping for stable long-horizon optimization.** We introduce DPC to adapt the PPO clipping range during training, improving optimization stability under non-stationarity, long-horizon credit assignment, and multi-objective reward coupling.
- **Comprehensive empirical validation.** Experiments on standard long-horizon forecasting benchmarks, cross-dataset transfer settings, and perturbation-based robustness tests show that TSLLM-BO consistently improves over strong LLM-based and non-LLM baselines while exhibiting faster and more stable RL convergence.

2 Related Work

2.0.1 LLMs for Time-Series Forecasting

LLM-based forecasting methods adapt pretrained language backbones to temporal data through prompt engineering, token reprogramming, patch embeddings, or lightweight adapters. Time-LLM [12] reprograms time-series inputs for language models; Auto-Times [13] formulates LLM forecasting autoregressively; One-Fits-All [14] studies pretrained language models for general time-series analysis; CALF [15] introduces cross-modal fine-tuning; and LLMTIME demonstrates zero-shot forecasting capabilities [16]. These developments sit within a broader discussion of time-series foundation models and the actual utility of language models for temporal prediction [10, 11, 18]. Despite promising results, most existing LLM forecasters remain trained with static supervised objectives, and recent work has explicitly identified modality-alignment barriers in LLM-based time-series analysis [21]. As a result, existing methods do not directly optimize sequence-level properties such as waveform realism, temporal alignment, or robustness to regime shift. Our work builds on this line of research but replaces the “SFT-only” view with a behavioral-alignment perspective.

2.0.2 RL for LLM Alignment

Reinforcement learning has become a standard tool for aligning large language models with high-level objectives. RLHF and its variants optimize generation quality beyond next-token likelihood by introducing task-level rewards, preference signals, or KL-regularized policy updates [23–25]. Related alignment strategies include Constitutional AI, DPO-style preference optimization, reinforced reasoning, REINFORCE-style learning, and adaptive reasoning-control mechanisms [26–31]. Surveys of LLM-agent optimization and RL-enhanced iterative reasoning further illustrate the breadth of this design space [32, 33]. However, their assumptions do not transfer directly to forecasting. Time-series forecasting uses continuous outputs rather than discrete tokens, and the quality of a forecast depends on coupled sequence-level criteria rather than a single reward dimension. Our method therefore adopts PPO [34] as the optimization backbone while redesigning both the reward and trust-region control for temporal prediction.

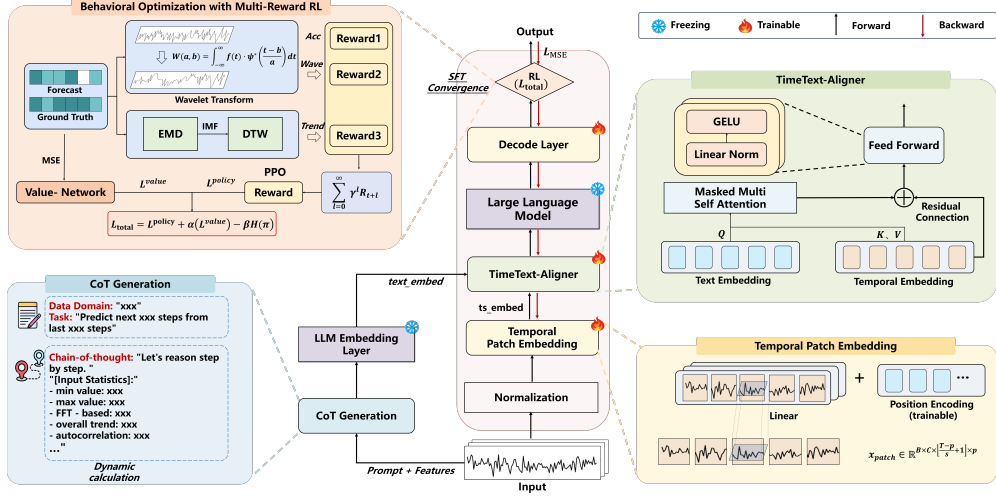


Fig. 3 Overview of the TSLLM-BO framework. The model is first trained by supervised fine-tuning and then refined by reinforcement learning. Temporal patches and textual prompts are aligned before entering a frozen LLM backbone. In the RL stage, a structure-aware reward jointly optimizes point accuracy, waveform similarity, and trend consistency, while Dynamic Policy Clipping stabilizes policy updates.

2.0.3 RL for Time-Series Forecasting

RL has a long history in sequential decision problems, from direct reinforcement learning for trading to deep value-based control [35, 36]. For forecasting specifically, RL has been used to dynamically combine predictive models [37]; recent work further couples language-guided forecasting with PPO [38] and studies time-series forecasting as reinforced reasoning [39]. RL-based optimization has also been applied to continuous waveform sensing in other signal-processing domains [40], illustrating the broader relevance of reward-driven adaptation for structured signals. Nevertheless, methods that combine LLMs with RL in forecasting remain comparatively scarce. Compared with these efforts, TSLLM-BO explicitly targets the central difficulties of LLM-based forecasting under RL: structure-aware reward design, stable long-horizon policy updates, and robustness under distribution shift.

3 Preliminaries

This section lays out the basic formalism that underpins our framework. We first recall the supervised forecasting setup and its standard objectives. We then cast forecasting as a sequential decision problem optimized via policy gradients. Finally, we present a task-agnostic multi-objective perspective that motivates our composite reward without committing to any particular instantiation.

3.1 Time-Series Forecasting as Supervised Learning

Given a multivariate time series $\mathbf{X}_{1:T}$, the goal is to predict the next T_{out} steps $\mathbf{Y}_{T+1:T+T_{\text{out}}}$. A forecasting model f_θ parameterized by θ produces:

$$\hat{\mathbf{Y}} = f_\theta(\mathbf{X}_{1:T}) \quad (1)$$

Conventional SFT minimizes point-wise errors such as mean squared error (MSE) or mean absolute error (MAE). A generic form is:

$$\mathcal{L}_{\text{SFT}} = \frac{1}{T_{\text{out}}} \sum_{t=1}^{T_{\text{out}}} \ell(\hat{y}_t, y_t) \quad (2)$$

where $\ell(\cdot)$ denotes a point-wise discrepancy function. While effective, this objective often overlooks structural or long-horizon properties of time series.

3.2 Reinforcement Learning for Sequence Prediction

We formulate forecasting as a sequential decision process. At step t , a state s_t encodes the historical context, the action a_t corresponds to the predicted value, and a reward R_t measures forecasting quality. The objective of reinforcement learning is to maximize the expected cumulative reward under policy π_θ :

$$J(\pi_\theta) = \mathbb{E}_{\pi_\theta} \left[\sum_{t=1}^{T_{\text{out}}} \gamma^{t-1} R_t \right] \quad (3)$$

where γ is a discount factor.

In policy-gradient methods, the optimization is based on updating parameters in the direction of

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\pi_\theta} [\nabla_\theta \log \pi_\theta(a_t | s_t) A_t] \quad (4)$$

where A_t is the advantage estimator. Various algorithms introduce different mechanisms to stabilize this update, such as trust-region constraints, clipping, or KL regularization, which limit large deviations between the updated policy and a reference policy.

3.3 A Multi-Objective View

Many forecasting applications inherently involve multiple quality criteria across horizons and structures. Let the per-step evaluation be vector-valued

$$\mathbf{r}_t = [r_t^{(1)}, \dots, r_t^{(K)}]^\top \in \mathbb{R}^K \quad (5)$$

where each component represents a task-agnostic objective (e.g., structural fidelity, stability, or risk-sensitive criteria without specifying a particular instantiation). A

common scalarization is a convex combination with weights $\mathbf{w} \in \Delta^{K-1}$:

$$R_t = \sum_{k=1}^K w_k r_t^{(k)} \quad \sum_{k=1}^K w_k = 1, w_k \geq 0. \quad (6)$$

Optimizing $J(\pi_\theta)$ with such R_t corresponds to seeking policies on or near the Pareto frontier of the underlying multi-objective problem, preventing degenerate solutions that over-optimize a single criterion at the expense of others. This abstract perspective will later be instantiated in our method by concrete design choices.

4 Methodology

The overall architecture is illustrated in Fig. 3. TSLLM-BO is designed around a simple principle: supervised learning should establish a strong numerical forecasting prior, while reinforcement learning should subsequently align the model with sequence-level temporal behavior. Accordingly, our method adopts a two-stage training pipeline. In the first stage, supervised fine-tuning learns the basic seasonal, local, and cross-variable dependencies from labeled data. In the second stage, PPO-based reinforcement learning refines the initialized model using a structure-aware reward and an adaptive policy-control mechanism.

The design contains three core ingredients. First, the model combines temporal patches with semantic prompt representations so that a frozen LLM backbone can process both numeric context and structured textual cues. Second, the reward function explicitly measures three complementary aspects of forecast quality: point accuracy, waveform fidelity, and trend alignment. Third, Dynamic Policy Clipping modulates the effective trust region during policy optimization, which is crucial for stable long-horizon RL in non-stationary forecasting tasks.

4.1 Model Structure

TSLLM-BO comprises five modules: temporal patch embedding, prompt construction and text embedding, a time-text aligner, a frozen LLM backbone, and a lightweight forecast decoder. In our implementation, the language backbone follows the LLaMA family [41]. The design deliberately separates representation learning from policy optimization: the front-end extracts temporally meaningful tokens, the frozen backbone supplies long-context semantic priors, and the RL stage later shapes the final forecasting behavior without destabilizing the full language model.

4.1.1 Temporal Patch Embedding

Given a multivariate time series input $\mathbf{X} \in \mathbb{R}^{B \times T \times N}$, where B is the batch size, T is the historical length, and N is the number of variables, we first permute it into shape $\mathbb{R}^{B \times N \times T}$ so that each variable is processed as an individual channel. Each channel is then segmented by a sliding window of length l and stride s :

$$\mathbf{Z} = \text{Unfold}(\mathbf{X}_{\text{perm}}, l, s) \in \mathbb{R}^{B \cdot N \times P \times l} \quad (7)$$

where $P = \lfloor \frac{T-l}{s} \rfloor + 1$ denotes the number of patches. Each patch is then projected to a latent space via a linear transformation:

$$\mathbf{E}_{\text{patch}} = \mathbf{Z} \cdot \mathbf{W}_e + \mathbf{b}_e \in \mathbb{R}^{B \cdot N \times P \times d} \quad (8)$$

4.1.2 Prompt Construction and Text Embedding

We summarize each input sequence with lightweight descriptive statistics, including extrema, median level, trend direction, and representative periodic lags, and convert them into natural-language prompts $\mathcal{T}_{\text{prompt}}$. The prompts are then tokenized and embedded by the frozen language model:

$$\mathbf{E}_{\text{text}} = \text{LLMEmbed}(\mathcal{T}_{\text{prompt}}) \in \mathbb{R}^{B \cdot N \times S \times d_{\text{LLM}}} \quad (9)$$

where S is the prompt token length and d_{LLM} denotes the hidden size of the LLM.

4.1.3 Time-Text Aligner

We use a multi-head cross-attention block to align temporal patch embeddings $\mathbf{E}_{\text{patch}}$ with textual prompt embeddings $\mathbf{E}_{\text{text}}$ so that numerical dynamics and semantic summaries interact before entering the LLM backbone:

$$\mathbf{E}_{\text{aligned}} = \text{MultiHeadAttn}(\mathbf{E}_{\text{patch}}, \mathbf{E}_{\text{text}}, \mathbf{E}_{\text{text}}) \quad (10)$$

This aligned representation is further enhanced by a feed-forward network (FFN) with residual connections:

$$\mathbf{E}_{\text{fused}} = \text{FFN}(\mathbf{E}_{\text{patch}} + \mathbf{E}_{\text{aligned}}) \quad (11)$$

4.1.4 Frozen LLM Backbone

The fused patch representations are concatenated with the prompt embeddings and passed through a frozen LLM backbone, which supplies long-context modeling capacity while keeping the RL fine-tuning stage computationally manageable:

$$\mathbf{H} = \text{LLM}([\mathbf{E}_{\text{text}}; \mathbf{E}_{\text{fused}}]) \quad (12)$$

4.1.5 Forecast Decoder

The LLM outputs corresponding to the patch tokens are reshaped and decoded by a parameter-efficient head to produce final predictions:

$$\hat{\mathbf{Y}} \in \mathbb{R}^{B \times T_{\text{out}} \times N} \quad (13)$$

This modular design allows TSLLM-BO to preserve local temporal detail, inject high-level semantic context, and expose a stable forecasting interface for the subsequent reinforcement-learning stage.

4.2 Two-Stage SFT→RL Training

4.2.1 Supervised Fine-Tuning

In the first stage we cast multivariate time-series forecasting as a standard supervised regression problem. Given an input tensor $\mathbf{X} \in \mathbb{R}^{B \times T \times N}$, the model produces predictions $\hat{\mathbf{Y}} \in \mathbb{R}^{B \times T_{\text{out}} \times N}$ that are compared with the ground truth $\mathbf{Y}$. To let the network learn short-term dynamics, seasonal components, and basic noise patterns, we minimise the MSE:

$$\mathcal{L}_{\text{SFT}} = \frac{1}{B T_{\text{out}} N} \sum_{i=1}^B \sum_{t=1}^{T_{\text{out}}} \sum_{n=1}^N (\hat{y}_{itn} - y_{itn})^2. \quad (14)$$

Minimising $\mathcal{L}_{\text{SFT}}$ drives the parameters to a stable solution θ_0 , providing a reliable forecasting baseline and well-conditioned gradients for the subsequent reinforcement-learning stage.

4.2.2 Reinforcement Learning Stage

After supervised initialization, we reinterpret forecasting as a sequential decision process in which each predicted step is an action and the entire trajectory is evaluated by a structure-aware reward. We then refine the model with PPO, using reward shaping and adaptive policy control to improve sequence-level behavior beyond what point-wise regression can express.

We design a combined instantaneous reward R_t at each time step t as a weighted sum of multiple sub-objectives:

$$R_t = w_1 R_t^{\text{acc}} + w_2 R_t^{\text{wave}} + w_3 R_t^{\text{trend}}, \quad (15)$$

where $R_t^{\text{acc}} = -\|\hat{\mathbf{y}}_t - \mathbf{y}_t\|_2$ quantifies point-wise prediction error, and

$$R_t^{\text{wave}} = -\frac{\|\nabla \hat{\mathbf{y}}_t - \nabla \mathbf{y}_t\|_2}{\|\nabla \mathbf{y}_t\|_2 + \epsilon} \quad (16)$$

captures waveform similarity by comparing first-order differences between the predicted and ground-truth sequences, with $\nabla \hat{\mathbf{y}}_t$ denoting the discrete temporal gradient and ϵ a small constant for numerical stability.

To evaluate long-range trend consistency, R_t^{trend} is defined based on Dynamic Time Warping (DTW), a classical alignment measure for sequences whose properties and variants have been studied extensively [42, 43], between the predicted and true sequences:

$$R_t^{\text{trend}} = -D(T, T), \quad (17)$$

$$D(i, j) = |\hat{y}_i - y_j| + \min \left\{ \begin{aligned} &D(i-1, j), \\ &D(i, j-1), \\ &D(i-1, j-1) \end{aligned} \right\} \quad (18)$$

with initialization $D(0, 0) = 0$. The coefficients $w_1, w_2, w_3 > 0$ are hyperparameters that balance the trade-offs among accuracy, waveform fidelity, and trend alignment.

The cumulative reward G_t from time step t onwards is

$$G_t = \sum_{l=0}^{\infty} \gamma^l R_{t+l}, \quad (19)$$

where γ is the discount factor. We use generalized advantage estimation (GAE) to compute

$$A_t = \sum_{l=0}^{\infty} (\gamma\lambda)^l (R_{t+l} + \gamma V_{\phi}(\mathbf{s}_{t+l+1}) - V_{\phi}(\mathbf{s}_t)), \quad (20)$$

where $V_{\phi}(\mathbf{s})$ is the value network and λ controls the bias-variance trade-off.

We optimize the policy network π_{θ} by maximizing

$$J(\pi) = \mathbb{E}_{\mathbf{a} \sim \pi} \left[\sum_{t=1}^T \gamma^{t-1} R_t \right]. \quad (21)$$

The policy loss with Dynamic Policy Clipping (DPC) is

$$\mathcal{L}_{\text{policy}} = -\mathbb{E}_t [\min(r_t A_t, \text{clip}(r_t, 1 - \varepsilon_t, 1 + \varepsilon_t) A_t)], \quad (22)$$

where $r_t = \frac{\pi_{\theta}(\mathbf{a}_t|\mathbf{s}_t)}{\pi_{\text{ref}}(\mathbf{a}_t|\mathbf{s}_t)}$ and the dynamic clipping factor is

$$\varepsilon_t = \text{clip}(\varepsilon_0 e^{-\lambda t} \cdot (1 + \kappa |r_t - 1|), \varepsilon_{\min}, \varepsilon_{\max}), \quad (23)$$

with $\varepsilon_0 e^{-\lambda t}$ an exponentially decaying baseline and $1 + \kappa |r_t - 1|$ a deviation-aware adjustment ($\kappa > 0$).

The value loss is

$$\mathcal{L}_{\text{value}} = \frac{1}{2} \mathbb{E}_t [(G_t - V_{\phi}(\mathbf{s}_t))^2], \quad (24)$$

and the policy entropy is

$$H(\pi_{\theta}) = -\mathbb{E}_t [\log \pi_{\theta}(\mathbf{a}_t|\mathbf{s}_t)]. \quad (25)$$

The total objective combines all terms:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{policy}} + \alpha \mathcal{L}_{\text{value}} - \beta H(\pi_{\theta}), \quad (26)$$

where α and β control relative importance.

4.2.3 Properties of Dynamic Policy Clipping

To characterize the effect of Dynamic Policy Clipping (DPC), we analyze its role in stabilizing training and accelerating convergence. Recall that the PPO clipped

surrogate objective is

$$L_t^{\text{clip}}(\theta) = \min \left(r_t A_t, \text{clip}(r_t, 1 - \varepsilon_t, 1 + \varepsilon_t) A_t \right), \quad (27)$$

where $r_t = \pi_\theta(a_t|s_t)/\pi_{\theta_{\text{ref}}}(a_t|s_t)$ and A_t is the estimated advantage. The dynamic clipping range ε_t is defined in (23), which combines an exponentially decaying baseline with a deviation-aware adjustment factor.

This adaptive rule guarantees that the update step remains bounded. The gradient of the policy term can be written as

$$\nabla_\theta L_t^{\text{clip}} \approx \nabla_\theta \log \pi_\theta(a_t|s_t) \hat{A}_t w_t, \quad (28)$$

with $w_t \in [1 - \varepsilon_t, 1 + \varepsilon_t]$. Because ε_t is restricted to $[\varepsilon_{\min}, \varepsilon_{\max}]$, the update magnitude satisfies

$$|\nabla_\theta L_t^{\text{clip}}| \leq (1 + \varepsilon_{\max}) |\hat{A}_t| \cdot |\nabla_\theta \log \pi_\theta(a_t|s_t)|, \quad (29)$$

ensuring bounded gradients and thereby mitigating the risk of explosion.

Furthermore, the exponential decay progressively reduces the effective trust region as training advances. Larger values of ε_t in the early phase promote exploration and rapid policy improvement, while smaller values in the later phase suppress excessive updates and reduce overfitting to noise. The deviation-aware factor in (23) further enlarges the clipping bound only when the current policy diverges significantly from the reference, while otherwise enforcing a tighter constraint. This design effectively establishes an adaptive trust region that combines faster exploration at the beginning of training with stable refinement in the later stage, thereby improving both convergence speed and final stability.

4.2.4 Policy–Reward Structural Coupling and Theoretical Insights into DPC

We formalize how the reward structure in (15) interacts with the DPC rule (23), explaining why the proposed SFT→RL pipeline remains stable under multi-objective optimization.

Proposition 1 (Scale-compatibility of rewards and clipping). Let the reward components be linearly rescaled as $R_t^{(k)'} = a_k R_t^{(k)}$ with $a_k > 0$, and define $R_t' = \sum_k w_k' R_t^{(k)'}$ with $w_k' = w_k/a_k$ (i.e., preserving the overall mixture). If the DPC baseline ε_0 is scaled by the same positive factor c as the induced change in the empirical policy ratio dispersion (i.e., choose $\varepsilon_0' = c\varepsilon_0$ while keeping κ, λ unchanged), then the gradient bound (29) is preserved up to a constant factor independent of $\{a_k\}$. Linear rescaling that is neutralized in the mixture leaves the signal-to-noise ratio of advantages essentially unchanged; the variance of r_t changes only by a constant factor captured by c . Since $w_t \in [1 - \varepsilon_t, 1 + \varepsilon_t]$, scaling ε_0 by c preserves the effective clipping window, keeping the RHS of (29) within the same order.

Proposition 2 (Adaptive trust-region over long horizons). With ε_t defined by (23) and $\lambda > 0$, the effective trust region width is non-increasing in t when $|r_t - 1|$ is bounded, and temporarily widens only when $|r_t - 1|$ grows. Consequently, the

variance of policy updates is large in early iterations (exploration) and diminishes as training proceeds (refinement), yielding stable convergence under bounded advantages. Monotonic decay from $e^{-\lambda t}$ contracts the window; the deviation-aware term $1 + \kappa|r_t - 1|$ expands it proportionally to mismatch. Under bounded A_t and log-gradients, the surrogate step stays within a clipped ball whose radius is controlled by ε_t .

Proposition 3 (Sensitivity bound w.r.t. reward weights). Consider the reward mixture weights $w = (w_1, w_2, w_3)$ in (15). There exists a feasible region of $(\varepsilon_0, \kappa, \lambda)$ such that for any small perturbation δw with $\|\delta w\|_2 \leq \rho$, the expected policy parameter update satisfies a Lipschitz-type bound $\|\Delta\theta\| \leq C(\varepsilon_{\max})\rho$, where $C(\cdot)$ is non-decreasing and independent of the absolute scale of R_t . The policy gradient depends on A_t , which is linear in R_t ; the DPC term clips r_t so that the amplification of δw is controlled by $\varepsilon_{\max}$. Standard smoothness assumptions on $\log \pi_\theta$ yield the Lipschitz dependence.

Practical tuning guidelines. Emphasizing point accuracy ($w_1 \uparrow$) suggests reducing ε_0 or increasing λ to prevent over-aggressive updates. Emphasizing long-range trend ($w_3 \uparrow$) suggests using smaller λ (slower decay) and a modestly larger κ to accommodate transient policy shifts induced by DTW-based rewards. When waveform fidelity (w_2) is high, keep $\varepsilon_{\min}$ moderately small to avoid oversmoothing sharp regime changes while retaining stability.

Remark (Pareto-consistent alignment). Under Propositions 1–3, DPC acts as an adaptive trust-region controller that harmonizes multi-objective rewards: it prevents single-objective dominance (“reward hijacking”), while enabling consistent progress toward Pareto-improving policies across accuracy, waveform similarity, and trend alignment.

5 Experiments

Table 1 Summary of datasets used for evaluation.

Dataset	Channels	Freq.	Points	Application domain
ETTm1/ETTm2	7	15 min	57,507	Energy infrastructure observation
ETTh1/ETTh2	7	1 h	14,307	Energy infrastructure observation
Electricity	321	1 h	26,211	Electricity consumption
Weather	21	10 min	52,603	Meteorological monitoring
Exchange	8	1 day	7,207	Foreign exchange market
Illness	7	1 month	720	Disease incidence statistics

Table 2 Comprehensive comparison of inference efficiency and RL training overhead between LangTime and TSLLM-BO.

Metric	LangTime	TSLLM-BO	Relative Diff.
<i>Inference efficiency (RL excluded)</i>			
LLM backbone size (B)	7.0	7.0	0%
Extra parameters (M)	20.3	17.0	-19.4%
FLOPs / sample (G)	3.40	2.30	-32.4%
Peak GPU memory (GB)	19.2	12.5	-34.9%
Average power (W)	297	218	-26.6%
Latency (ms)	82.0	60.5	-26.2%
<i>RL training overhead (on top of SFT; per epoch)</i>			
Extra FLOPs / epoch (PFLOPs)	1.15	0.78	-32.2%
Wall-clock / epoch (h)	12.0	8.5	-29.2%
Total training time (h)	240	172	-28.3%
Peak GPU memory (GB)	42.0	27.5	-34.5%
Peak power (W)	431	367	-14.8%
RL loss variance σ	0.42	0.29	-31.0%

5.1 Main Result

5.1.1 Long-Horizon Forecasting and Zero-Shot Transfer

We evaluate TSLLM-BO from two complementary perspectives: standard long-horizon forecasting on seen datasets and zero-shot transfer across related datasets. The first setting measures in-domain forecasting quality under multiple horizons, while the second probes whether the learned forecasting behavior transfers to unseen temporal distributions without additional fine-tuning.

As shown in Table 3, TSLLM-BO consistently ranks first or second across the vast majority of dataset–horizon pairs in both MSE and MAE. The gains are especially visible on difficult settings with long prediction horizons and noisy dynamics, such as *Traffic*, *Illness*, and the longer ETT horizons. These results indicate that the RL stage is not merely overfitting local training signals; rather, it improves how the model behaves over an entire forecast trajectory. Figure 4 provides a qualitative view: compared with strong baselines, TSLLM-BO better preserves turning points, local smoothness, and long-range trend evolution.

Table 4 reports zero-shot transfer results. In this setting, a model is trained on a source dataset and directly evaluated on a target dataset without target-domain adaptation. TSLLM-BO achieves the best performance on all four transfer tasks, outperforming the strongest baseline by a clear margin on average. This result is important because it suggests that the proposed behavioral alignment does not only improve in-distribution accuracy; it also yields forecasting policies that generalize more robustly under cross-dataset distribution shift.

Table 3 Forecasting performance comparisons. The input sequence length is 96 for the Illness dataset and 512 for others. Lower values indicate better performance. Best results are **bold**, second best are underlined.

	Methods	TSLLM-BO (Ours)		LangTime (2025)		TimeLLM (2024)		CALF (2024)		GPT4TS (2023)		iTransformer (2023)		DLinear (2023)		PatchTST (2023)		TimesNet (2022)	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
51	ETTh1	96	0.365 0.381	0.372	<u>0.392</u>	0.376	0.402	<u>0.370</u>	0.393	0.376	0.397	0.386	0.405	0.375	0.399	0.404	0.413	0.384	0.402
		192	0.405 0.396	0.413	<u>0.410</u>	<u>0.407</u>	0.421	0.429	0.426	0.416	0.418	0.441	0.436	0.408	0.416	0.454	0.430	0.436	0.429
		336	<u>0.432</u> 0.429	0.443	0.448	0.430	<u>0.433</u>	0.451	0.440	0.442	0.433	0.487	0.458	0.439	0.443	0.497	0.462	0.491	0.469
		720	0.438 0.447	<u>0.450</u>	<u>0.459</u>	0.457	0.468	0.476	0.466	0.477	0.476	0.503	0.491	0.472	0.490	0.496	0.481	0.521	0.500
		avg	0.410 0.413	0.420	<u>0.427</u>	<u>0.418</u>	0.432	0.432	0.431	0.428	0.431	0.454	0.448	0.423	0.437	0.463	0.447	0.458	0.450
	ETTh2	96	0.275 0.325	<u>0.283</u>	<u>0.331</u>	0.286	0.346	0.284	0.336	0.285	0.342	0.297	0.349	0.289	0.353	0.312	0.358	0.340	0.374
		192	0.339 0.356	<u>0.347</u>	<u>0.373</u>	0.361	0.391	0.353	0.378	0.354	0.389	0.380	0.400	0.383	0.418	0.397	0.408	0.402	0.414
		336	<u>0.371</u> 0.380	0.377	0.395	0.390	0.414	0.361	<u>0.394</u>	0.373	0.407	0.428	0.432	0.448	0.465	0.435	0.440	0.452	0.452
		720	0.384 0.422	<u>0.395</u>	<u>0.426</u>	0.405	0.428	0.406	0.428	0.406	0.441	0.427	0.445	0.605	0.551	0.436	0.449	0.462	0.468
		avg	0.342 0.371	<u>0.351</u>	<u>0.383</u>	0.361	0.396	0.351	0.384	0.355	0.395	0.383	0.407	0.431	0.447	0.395	0.414	0.414	0.427
	ETTm1	96	0.291 0.332	0.293	<u>0.341</u>	0.299	0.350	0.323	0.350	<u>0.292</u>	0.346	0.334	0.368	0.299	0.343	0.334	0.373	0.338	0.375
		192	0.334 0.356	0.341	0.369	0.346	<u>0.367</u>	0.375	0.376	<u>0.339</u>	0.372	0.377	0.391	0.335	0.365	0.367	0.386	0.374	0.387
		336	0.359 0.379	<u>0.371</u>	<u>0.380</u>	0.372	0.398	0.411	0.401	0.366	0.394	0.426	0.420	0.369	0.386	0.392	0.407	0.410	0.411
		720	0.433 0.418	<u>0.432</u>	<u>0.419</u>	0.441	0.431	0.476	0.438	0.424	0.421	0.491	0.459	0.425	0.421	0.464	0.442	0.478	0.450
		avg	0.354 0.372	<u>0.359</u>	<u>0.377</u>	0.364	0.386	0.396	0.391	0.355	0.383	0.407	0.410	0.357	0.379	0.389	0.402	0.400	0.406
	ETTh2	96	0.160 0.254	<u>0.162</u>	0.248	0.173	0.269	0.177	0.255	0.173	0.262	0.180	0.264	0.167	0.260	0.177	0.260	0.187	0.267
		192	<u>0.226</u> 0.288	0.235	0.304	<u>0.233</u>	0.300	0.245	0.300	0.229	0.301	0.250	0.309	0.224	0.303	0.246	0.305	0.249	0.309
		336	0.277 0.317	0.280	<u>0.329</u>	0.289	0.335	0.309	0.341	0.286	0.341	0.311	0.348	0.281	0.343	0.305	0.343	0.321	0.351
		720	0.353 0.378	<u>0.366</u>	<u>0.382</u>	<u>0.365</u>	0.398	0.402	0.395	0.378	0.401	0.412	0.407	0.397	0.421	0.410	0.405	0.408	0.403
		avg	0.254 0.309	<u>0.261</u>	<u>0.316</u>	0.265	0.326	0.283	0.323	0.267	0.326	0.288	0.332	0.267	0.332	0.285	0.328	0.291	0.333
	Weather	96	0.150 0.194	0.160	0.217	<u>0.155</u>	<u>0.199</u>	0.168	0.207	0.162	0.212	0.174	0.214	0.176	0.237	0.177	0.218	0.172	0.220
		192	0.197 0.237	0.206	0.255	0.223	0.261	0.216	0.251	<u>0.204</u>	<u>0.248</u>	0.221	0.254	0.220	0.282	0.219	0.259	0.219	0.261
		336	0.251 0.276	0.263	0.294	0.255	<u>0.279</u>	0.271	0.292	<u>0.254</u>	<u>0.286</u>	0.278	0.296	0.265	0.319	0.280	0.297	0.280	0.306
		720	0.331 0.319	0.343	<u>0.337</u>	0.345	0.342	0.350	0.345	<u>0.326</u>	0.338	0.358	0.347	0.323	0.362	0.365	0.347	0.365	0.359
		avg	0.232 0.257	0.243	0.276	0.245	<u>0.270</u>	0.251	0.274	<u>0.237</u>	0.271	0.258	0.278	0.246	0.300	0.260	0.280	0.259	0.287
	Electricity	96	0.130 0.220	0.141	<u>0.232</u>	<u>0.137</u>	0.233	0.147	0.240	0.139	0.238	0.148	0.240	0.140	0.237	0.186	0.269	0.168	0.272
		192	0.151 0.242	0.159	0.256	<u>0.152</u>	<u>0.247</u>	0.163	0.254	0.153	0.251	0.162	0.253	0.153	0.249	0.190	0.273	0.184	0.289
		336	0.162 0.244	<u>0.169</u>	<u>0.258</u>	0.171	0.267	0.178	0.270	0.169	0.266	0.178	0.269	0.169	0.267	0.206	0.290	0.198	0.300
		720	0.197 0.253	0.204	<u>0.267</u>	<u>0.200</u>	0.298	0.215	0.300	0.206	0.297	0.225	0.317	0.203	0.301	0.247	0.322	0.220	0.320
		avg	0.160 0.240	0.168	<u>0.253</u>	<u>0.165</u>	0.261	0.176	0.266	0.167	0.263	0.178	0.270	0.166	0.264	0.207	0.289	0.193	0.295
	Traffic	96	0.361 0.249	0.371	0.264	0.392	0.267	0.416	0.274	0.388	0.282	0.395	0.268	0.410	0.282	<u>0.369</u>	0.253	0.593	0.321
		192	0.376 <u>0.261</u>	0.385	0.275	0.409	0.271	0.430	0.276	0.407	0.290	0.417	0.276	0.423	0.287	<u>0.379</u>	0.259	0.617	0.336
		336	0.389 <u>0.277</u>	0.397	0.299	0.434	0.296	0.451	0.286	0.412	0.294	0.433	0.283	0.436	0.296	<u>0.392</u>	0.273	0.629	0.336
		720	0.432 0.286	0.443	0.304	0.451	0.301	0.478	0.301	0.450	0.312	0.467	0.302	0.466	0.315	<u>0.441</u>	0.289	0.640	0.350
		avg	0.389 0.268	0.399	0.286	0.422	0.283	0.444	0.284	0.414	0.295	0.428	0.282	0.434	0.295	<u>0.391</u>	<u>0.269</u>	0.620	0.336
	Illness	24	1.451 0.783	1.458	0.803	1.792	0.807	1.460	0.788	2.063	0.881	2.347	1.731	2.215	1.081	2.335	0.989	2.317	0.934
		36	1.566 0.801	<u>1.573</u>	<u>0.817</u>	1.833	0.833	1.573	0.837	1.868	0.892	2.468	0.998	1.963	0.963	2.561	1.035	1.972	0.920
		48	1.664 0.857	<u>1.677</u>	<u>0.876</u>	2.269	1.012	1.784	0.890	1.790	0.884	2.489	1.016	2.130	1.024	2.465	1.022	2.238	0.940
		60	1.641 <u>0.860</u>	<u>1.651</u>	<u>0.879</u>	2.177	0.925	1.982	0.962	1.979	0.957	2.471	1.065	2.368	1.096	2.189	0.997	2.027	0.928
		avg	1.581 0.825	<u>1.590</u>	<u>0.844</u>	2.018	0.894	1.700	0.869	1.925	0.904	2.444	1.203	2.169	1.041	2.388	1.011	2.139	0.931

Table 4 Zero-shot learning results on ETT datasets. Each result is the average of predictions for {96, 192, 336, 720} horizons. Best results are **bold**, second best are underlined.

Methods	TSLLM-BO (Ours)		LangTime (2025)		TIME-LLM (2024)		GPT4TS (2023)		LLMTime (2023)		DLinear (2023)		PatchTST (2023)		TimesNet (2023)	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
$ETTh1 \rightarrow ETTh2$	0.357	0.385	0.364	0.399	<u>0.363</u>	<u>0.396</u>	0.406	0.422	0.992	0.708	0.493	0.488	0.380	0.405	0.421	0.431
$ETTh1 \rightarrow ETTm2$	0.276	0.348	<u>0.283</u>	<u>0.352</u>	0.292	0.357	0.325	0.363	1.867	0.869	0.415	0.452	0.314	0.360	0.327	0.361
$ETTh2 \rightarrow ETTh1$	0.477	0.483	<u>0.487</u>	<u>0.495</u>	0.499	0.496	0.757	0.578	1.961	0.981	0.703	0.574	0.565	0.513	0.865	0.621
$ETTh2 \rightarrow ETTm2$	0.281	0.346	<u>0.287</u>	<u>0.356</u>	0.303	0.358	0.335	0.370	1.867	0.869	0.328	0.368	0.325	0.365	0.342	0.376

Table 5 Robustness Comparison under Noise and Missingness Perturbations across Representative Datasets. Forecasting horizons are 512 steps for ETTh1, ETTh2, and Traffic, and 96 steps for Illness. Best results are **bold**, second best are underlined.

Dataset / Perturbation	TSLLM-BO (Ours)		LangTime (2025)		Time-LLM (2024)		GPT4TS (2023)		CALF (2025)		PatchTST (2023)	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
<i>ETTh1 (96-step Forecasting)</i>												
0% Noise / Missing	0.365	0.381	0.372	<u>0.392</u>	0.376	0.402	0.376	0.397	<u>0.370</u>	0.393	0.404	0.413
10% Noise	0.376	0.386	0.386	<u>0.398</u>	0.389	0.409	0.392	0.404	<u>0.383</u>	0.399	0.419	0.422
20% Noise	0.392	0.392	<u>0.403</u>	<u>0.409</u>	0.407	0.417	0.412	0.411	<u>0.404</u>	0.411	0.436	0.429
10% Missing	0.378	0.387	0.389	0.399	0.393	0.409	0.395	0.403	<u>0.387</u>	<u>0.398</u>	0.421	0.421
20% Missing	0.398	0.395	0.411	<u>0.408</u>	0.414	0.418	0.419	0.410	<u>0.404</u>	0.409	0.442	0.430
<i>ETTh2 (96-step Forecasting)</i>												
0% Noise / Missing	0.275	0.325	0.283	<u>0.331</u>	0.286	0.346	0.285	0.342	0.284	0.336	0.312	0.358
10% Noise	0.284	0.331	<u>0.292</u>	<u>0.338</u>	0.298	0.352	0.295	0.349	0.296	0.343	0.321	0.363
20% Noise	0.298	0.339	<u>0.307</u>	<u>0.346</u>	0.315	0.358	0.314	0.357	0.311	0.354	0.336	0.372
10% Missing	0.287	0.329	<u>0.296</u>	<u>0.336</u>	0.301	0.351	0.299	0.347	0.298	0.342	0.326	0.361
20% Missing	0.303	0.341	<u>0.313</u>	<u>0.348</u>	0.320	0.356	0.318	0.354	0.315	0.350	0.339	0.369
<i>Traffic (96-step Forecasting)</i>												
0% Noise / Missing	0.361	0.249	<u>0.371</u>	<u>0.264</u>	0.392	0.267	0.416	0.274	0.388	0.282	0.369	0.253
10% Noise	0.372	0.254	<u>0.398</u>	<u>0.279</u>	0.416	0.283	0.409	0.289	0.432	0.281	0.384	0.260
20% Noise	0.388	0.263	<u>0.432</u>	<u>0.293</u>	0.448	0.298	0.442	0.303	0.461	0.293	0.417	0.275
10% Missing	0.371	0.253	<u>0.423</u>	<u>0.281</u>	0.440	0.289	0.434	0.286	0.452	0.286	0.401	0.265
20% Missing	0.385	0.259	<u>0.452</u>	<u>0.296</u>	0.469	0.303	0.461	0.302	0.476	0.295	0.428	0.275
<i>Illness (24-step Forecasting)</i>												
0% Noise / Missing	1.451	0.783	<u>1.458</u>	0.803	1.792	0.807	1.460	<u>0.788</u>	2.063	0.881	2.335	0.989
10% Noise	1.478	0.788	<u>1.485</u>	0.807	1.826	0.813	1.491	<u>0.806</u>	2.092	0.886	2.362	0.993
20% Noise	1.525	0.794	<u>1.541</u>	<u>0.814</u>	1.872	0.820	1.548	0.816	2.141	0.893	2.405	1.001
10% Missing	1.466	0.786	<u>1.474</u>	0.806	1.810	0.811	1.479	<u>0.797</u>	2.079	0.884	2.353	0.991
20% Missing	1.512	0.793	<u>1.524</u>	0.812	1.857	0.818	1.528	<u>0.808</u>	2.128	0.890	2.398	0.998

Table 6 Ablation study on key components of TSLLM-BO. Metrics: MSE / MAE (lower is better).

R_t^{acc}	R_t^{wave}	R_t^{trend}	Traffic		Illness	
			MSE	MAE	MSE	MAE
✓	✓	✓	0.389	0.268	1.581	0.825
	✓	✓	0.406	0.281	1.843	0.861
✓		✓	0.399	0.276	1.797	0.855
✓	✓		0.396	0.277	1.703	0.841
			0.416	0.284	2.015	0.888

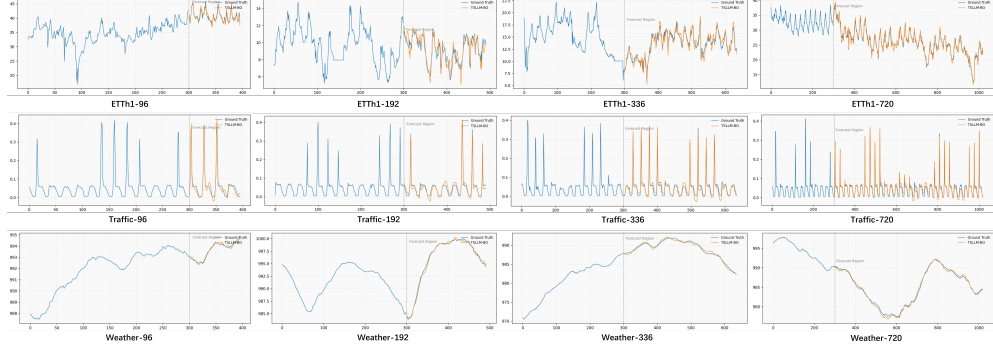


Fig. 4 Ground-truth and forecasted time-series curves on ETTh1, Traffic, and Weather datasets under different forecast horizons (96, 192, 336, and 720). The dashed line separates the historical input and forecast regions.

5.2 Experimental Settings

5.2.1 Benchmarks

We evaluate TSLLM-BO on standard long-horizon multivariate forecasting benchmarks spanning energy infrastructure, meteorology, electricity consumption, traffic, epidemiology, and exchange-rate data [2]. These datasets differ substantially in sampling frequency, dimensionality, seasonality, and noise level, which makes them suitable for testing not only average forecasting accuracy but also robustness across heterogeneous temporal regimes. Dataset statistics are summarized in Table 1.

5.2.2 Baselines

We compare TSLLM-BO against strong LLM-based forecasters, including LangTime [38], TimeLLM [12], GPT4TS [14], CALF [15], and LLMTIME [16], as well as representative neural forecasting models such as iTransformer [3], DLinear [4], PatchTST [5], and TimesNet [6]. To ensure a fair architectural comparison among LLM-based methods, we use the same LLaMA-7B backbone [41] whenever applicable.

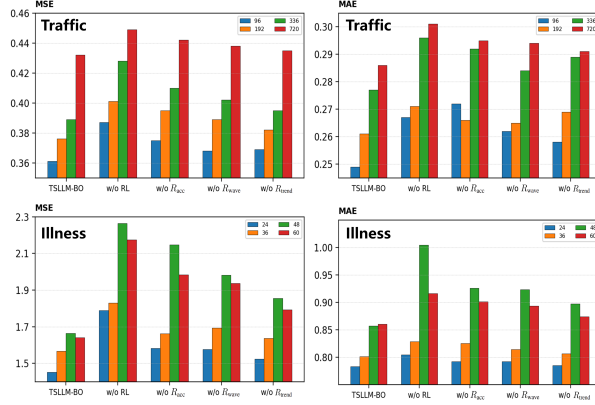


Fig. 5 Impact of Core Component Ablations on TSLLM-BO: MSE and MAE Results on Traffic and Illness Datasets.

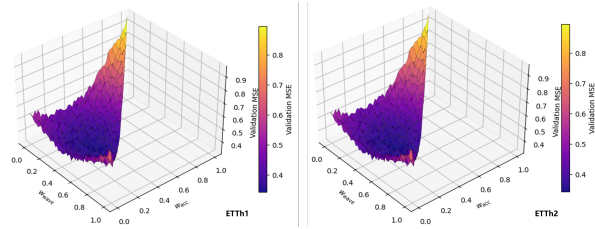


Fig. 6 Reward-weight sensitivity surfaces for the 96-step horizon. Darker colors indicate lower MSE, with minima at balanced weights.

Because the official implementation of LangTime is unavailable, we carefully reproduce it from the paper description and report a separate efficiency comparison in Table 2.

5.2.3 Evaluation Protocol

We report mean squared error (MSE) [19] and mean absolute error (MAE) [20], following the standard long-horizon forecasting protocol used in prior work. All methods are evaluated under the same data splits and preprocessing pipeline provided by the Time-Series-Library¹. Experiments are conducted on an NVIDIA RTX6000 Ada GPU and repeated with at least three random seeds. In addition to standard forecasting accuracy, we further examine cross-dataset transfer, perturbation robustness, ablation behavior, and RL convergence stability.

5.2.4 Ablation Study

To isolate the contribution of each design choice, we conduct component ablations on the training pipeline and reward function (Table 6 and Fig. 5). The first observation is that removing the RL stage consistently hurts performance, confirming that TSLLM-BO gains are not explained by the LLM backbone alone. The degradation is particularly

¹<https://github.com/thuml/Time-Series-Library>

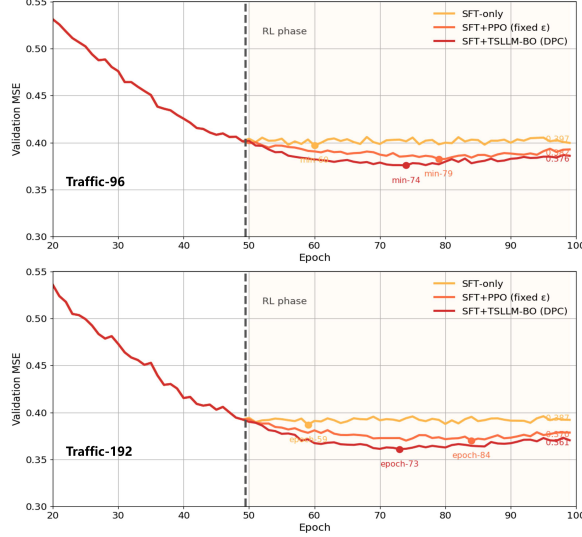


Fig. 7 Validation MSE trajectories during the RL stage on the Traffic dataset. TSLLM-BO with Dynamic Policy Clipping converges faster and reaches a lower error floor than fixed-clipping or SFT-only variants.

clear on noisy or harder datasets such as *Traffic* and *Illness*, where the SFT-only variant struggles to maintain stable long-horizon behavior.

The second observation is that the three reward terms are complementary. Removing either the point-accuracy term or the waveform term produces the largest degradation, indicating that local numerical fidelity and shape preservation are both necessary. Eliminating the trend term also causes a measurable drop, especially on longer horizons where phase consistency matters more. Overall, the full model achieves the best balance across MSE, MAE, and trajectory quality metrics, validating the design of the composite reward.

We further study reward-weight sensitivity by sweeping the coefficient space on *ETTh1* and *ETTh2* (Fig. 6). The resulting validation-error landscapes show broad low-error regions near balanced weight assignments rather than a brittle single-point optimum. This is encouraging from a deployment perspective: the reward is expressive enough to guide behavior, yet sufficiently well conditioned that moderate coefficient perturbations do not collapse performance.

5.2.5 Convergence Efficiency Analysis

Figure 7 compares validation-MSE trajectories during the RL stage. Once supervised training saturates, the SFT-only model quickly plateaus, while PPO with a fixed clipping range continues to improve only slowly. TSLLM-BO reaches a better minimum earlier, showing that DPC improves both optimization speed and final solution quality rather than merely smoothing the curve.

We also inspect KL divergence and the learned clipping range ϵ_t in Fig. 8. Under DPC, the policy shift remains tightly controlled, whereas fixed or unclipped baselines

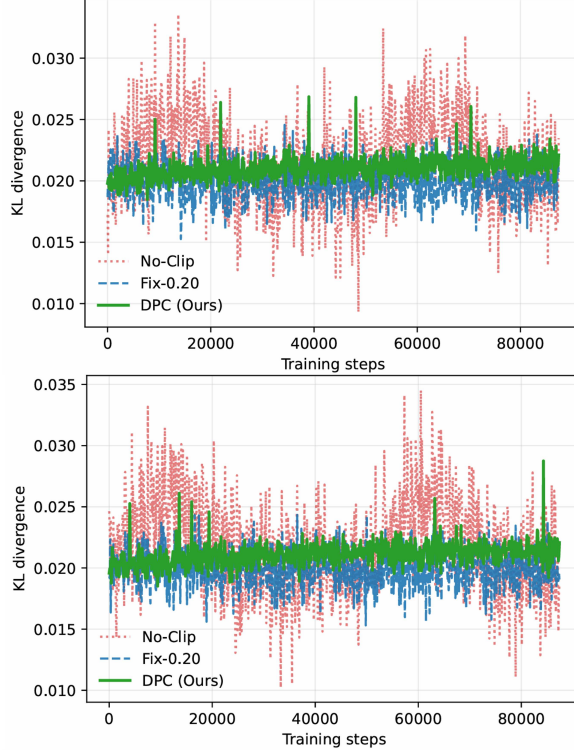


Fig. 8 Training-stability analysis of Dynamic Policy Clipping. Left: KL-divergence evolution during RL fine-tuning. Right: adaptive clipping range ϵ_t over training. DPC keeps policy updates stable while shrinking the trust region smoothly.

exhibit larger oscillations and occasional instability. The adaptive clipping schedule therefore behaves as intended: it allows relatively aggressive updates early in training and progressively tightens the trust region as the policy approaches a stable solution. This behavior is especially valuable in long-horizon forecasting, where poorly controlled policy drift can rapidly destabilize sequence-level rewards.

6 Discussion

The empirical results support a central claim of this paper: for LLM-based time-series forecasting, optimizing point-wise regression loss alone is not sufficient. TSLLM-BO improves performance because it adds a second stage that explicitly aligns forecast trajectories with sequence-level temporal structure. This is most visible in three scenarios: long-horizon prediction, cross-dataset transfer, and robustness under noise or missingness. In all three settings, the gap between pure SFT and structure-aware RL becomes larger, suggesting that behavioral alignment is especially useful when forecasting uncertainty accumulates over time.

Another important observation is that the gain does not come from a larger or more flexible language backbone. We keep the LLM frozen and instead improve how

the forecasting head is trained and aligned. This makes the method attractive for practical deployment, where full-parameter RL tuning of a large backbone would be prohibitively expensive. The comparison with LangTime further indicates that stable RL for forecasting depends not only on having a reward, but on how that reward is structured and how policy updates are controlled.

At the same time, TSLLM-BO has limitations. First, the reward coefficients remain manually specified, even though our sensitivity analysis shows that the method is reasonably robust in a broad region. Second, the current formulation focuses on point forecasting; extending the same behavioral-alignment principle to probabilistic forecasting, calibrated uncertainty, or decision-aware forecasting is a promising next step. Third, while the frozen-backbone design is computationally attractive, future work could explore parameter-efficient adaptation strategies that selectively update deeper temporal reasoning layers.

Overall, we view TSLLM-BO not only as a forecasting model, but also as a methodological bridge between LLM alignment and temporal prediction. It suggests that importing alignment ideas from language modeling into time-series analysis is fruitful, provided that the reward design and optimization mechanisms are redesigned for temporal structure rather than copied directly from text.

7 Conclusion

We presented TSLLM-BO, a structure-aware behavioral alignment framework for LLM-based time-series forecasting. The method combines a supervised initialization stage with reinforcement-learning refinement, a composite temporal reward, and Dynamic Policy Clipping for stable long-horizon optimization. Across standard forecasting benchmarks, zero-shot transfer tasks, and perturbation-based robustness evaluations, TSLLM-BO consistently improves over strong LLM-based and neural baselines. These findings indicate that the next step for LLM forecasting is not merely scaling backbone size, but aligning model behavior with sequence-level temporal objectives. We hope this work encourages further research on RL-based alignment, uncertainty-aware temporal rewards, and decision-oriented forecasting with foundation models.

References

- [1] Yu, Q., Yang, G., Wang, X., Shi, Y., Feng, Y., Liu, A.: A review of time series forecasting and spatio-temporal series forecasting in deep learning. *The Journal of Supercomputing* **81**(10), 1–48 (2025)
- [2] Challu, C., Olivares, K.G., Oreshkin, B.N., Ramirez, F.G., Canseco, M.M., Dubrawski, A.: NHITS: Neural hierarchical interpolation for time series forecasting. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, pp. 6989–6997 (2023)
- [3] Liu, Y., Hu, T., Zhang, H., Wu, H., Wang, S., Ma, L., Long, M.: iTransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625* (2023)

- [4] Zeng, A., Chen, M., Zhang, L., Xu, Q.: Are transformers effective for time series forecasting? In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, pp. 11121–11128 (2023)
- [5] Nie, Y., Nguyen, N.H., Sinthong, P., Kalagnanam, J.: A time series is worth 64 words: Long-term forecasting with transformers. arXiv preprint arXiv:2211.14730 (2023)
- [6] Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., Long, M.: TimesNet: Temporal 2d-variation modeling for general time series analysis. arXiv preprint arXiv:2210.02186 (2022)
- [7] Zhang, W., Yang, L., Geng, S., Hong, S.: Self-supervised time series representation learning via cross reconstruction transformer. IEEE Transactions on Neural Networks and Learning Systems (2023)
- [8] Huang, K., Chen, W., Zhu, F., Li, P.W.-L., Kapahi, P., Bai, H.: RiboTag translatomic profiling of *Drosophila* oenocytes under aging and induced oxidative stress. BMC Genomics **20**(1), 50 (2019)
- [9] Zhang, J., Zhang, W., Wei, L., Zhang, L., Liu, J., Huang, S., Li, S., Yang, W., Li, K.: E93 promotes transcription of RHG genes to initiate apoptosis during *Drosophila* salivary gland metamorphosis. Insect Science **30**(3), 588–598 (2023)
- [10] Jin, M., Zhang, Y., Chen, W., Zhang, K., Liang, Y., Yang, B., Wang, J., Pan, S., Wen, Q.: Position: What can large language models tell us about time series analysis. In: Proceedings of the 41st International Conference on Machine Learning (ICML) (2024)
- [11] Ye, J., Zhang, W., Yi, K., Yu, Y., Li, Z., Li, J., Tsung, F.: A survey of time series foundation models: Generalizing time series representation with large language model. arXiv preprint arXiv:2405.02358 (2024)
- [12] Jin, M., Wang, S., Ma, L., Chu, Z., Zhang, J.Y., Shi, X., Chen, P.-Y., Liang, Y., Li, Y.-F., Pan, S.: Time-LLM: Time series forecasting by reprogramming large language models. arXiv preprint arXiv:2310.01728 (2023)
- [13] Liu, Y., Qin, G., Huang, X., Wang, J., Long, M.: AutoTimes: Autoregressive time series forecasters via large language models. Advances in Neural Information Processing Systems **37**, 122154–122184 (2024)
- [14] Zhou, T., Niu, P., Sun, L., Jin, R.: One fits all: Power general time series analysis by pretrained LM. Advances in Neural Information Processing Systems **36**, 43322–43355 (2023)
- [15] Liu, P., Guo, H., Dai, T., Li, N., Bao, J., Ren, X., Jiang, Y., Xia, S.-T.: CALF:

- Aligning LLMs for time series forecasting via cross-modal fine-tuning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 39, pp. 18915–18923 (2025)
- [16] Gruver, N., Finzi, M., Qiu, S., Wilson, A.G.: Large language models are zero-shot time series forecasters. *Advances in Neural Information Processing Systems* **36**, 19622–19635 (2023)
 - [17] Zhou, Y., Chu, Z., Ruan, Y., Jin, G., Huang, Y., Li, S.: PTSE: A multi-model ensemble method for probabilistic time series forecasting. *arXiv preprint arXiv:2305.11304* (2023)
 - [18] Tan, M., Merrill, M., Gupta, V., Althoff, T., Hartvigsen, T.: Are language models actually useful for time series forecasting? *Advances in Neural Information Processing Systems* **37**, 60162–60191 (2024)
 - [19] Köksoy, O.: Multiresponse robust design: Mean square error (MSE) criterion. *Applied Mathematics and Computation* **175**(2), 1716–1729 (2006)
 - [20] Hodson, T.O.: Root mean square error (RMSE) or mean absolute error (MAE): When to use them or not. *Geoscientific Model Development Discussions* **2022**, 1–10 (2022)
 - [21] Zheng, L.N., Dong, C., Zhang, W.E., Yue, L., Xu, M., Maennel, O., Chen, W.: Understanding why large language models can be ineffective in time series analysis: The impact of modality alignment. In: Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Vol. 2, pp. 4026–4037 (2025)
 - [22] Zhang, Y., He, Z., Yen, G.G.: Uncovering large language model weaknesses in character and word understanding and manipulating. *IEEE Transactions on Neural Networks and Learning Systems* (2025)
 - [23] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., *et al.*: Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems* **35**, 27730–27744 (2022)
 - [24] Chaudhari, S., Aggarwal, P., Murahari, V., Rajpurohit, T., Kalyan, A., Narasimhan, K., Deshpande, A., Silva, B.: RLHF deciphered: A critical analysis of reinforcement learning from human feedback for LLMs. *ACM Computing Surveys* (2024)
 - [25] Lee, H., Phatale, S., Mansoor, H., Lu, K.R., Mesnard, T., Ferret, J., Bishop, C., Hall, E., Carbune, V., Rastogi, A.: RLAIF: Scaling reinforcement learning from human feedback with AI feedback. *Online preprint* (2023)

- [26] Bai, Y., Kadavath, S., Kundu, S., Askell, A., Kernion, J., Jones, A., Chen, A., Goldie, A., Mirhoseini, A., McKinnon, C., et al.: Constitutional AI: Harmlessness from AI feedback. arXiv preprint arXiv:2212.08073 (2022)
- [27] Pal, A., Karkhanis, D., Dooley, S., Roberts, M., Naidu, S., White, C.: Smaug: Fixing failure modes of preference optimisation with DPO-positive. arXiv preprint arXiv:2402.13228 (2024)
- [28] Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y.K., Wu, Y., et al.: DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300 (2024)
- [29] Ahmadian, A., Cremer, C., Gallé, M., Fadaee, M., Kreutzer, J., Pietquin, O., Üstün, A., Hooker, S.: Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs. arXiv preprint arXiv:2402.14740 (2024)
- [30] Chen, S., Li, B.: Toward adaptive reasoning in large language models with thought rollback. arXiv preprint arXiv:2412.19707 (2024)
- [31] Dai, M., Yang, C., Si, Q.: S-GRPO: Early exit via reinforcement learning in reasoning models. arXiv preprint arXiv:2505.07686 (2025)
- [32] Wang, Q., Ding, R., Zeng, Y., Chen, Z., Chen, L., Wang, S., Xie, P., Huang, F., Zhao, F.: VRAG-RL: Empower vision-perception-based RAG for visually rich information understanding via iterative reasoning with reinforcement learning. arXiv preprint arXiv:2505.22019 (2025)
- [33] Du, S., Zhao, J., Shi, J., Xie, Z., Jiang, X., Bai, Y., He, L.: A survey on the optimization of large language model-based agents. arXiv preprint arXiv:2503.12434 (2025)
- [34] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)
- [35] Moody, J., Saffell, M.: Learning to trade via direct reinforcement. IEEE Transactions on Neural Networks **12**(4), 875–889 (2001)
- [36] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529–533 (2015)
- [37] Fu, Y., Wu, D., Boulet, B.: Reinforcement learning based dynamic model combination for time series forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 6639–6647 (2022)
- [38] Niu, W., Xie, Z., Sun, Y., He, W., Xu, M., Hao, C.: LangTime: A language-guided

- unified model for time series forecasting with proximal policy optimization. arXiv preprint arXiv:2503.08271 (2025)
- [39] Luo, Y., Zhou, Y., Cheng, M., Wang, J., Wang, D., Pan, T., Zhang, J.: Time series forecasting as reasoning: A slow-thinking approach with reinforced LLMs. arXiv preprint arXiv:2506.10630 (2025)
- [40] Zhu, P., Si, X., Han, J., Liang, J.: Nonlinear waveform sensing for cognitive radar based on reinforcement learning. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* (2025)
- [41] Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al.: LLaMA 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 (2023)
- [42] Yadav, M., Alam, M.A.: Dynamic time warping (DTW) algorithm in speech: A review. *International Journal of Research in Electronics and Computer Engineering* **6**(1), 524–528 (2018)
- [43] Rajput, K., Nguyen, D.B., Chen, G.: Evaluating DTW measures via a synthesis framework for time-series data. arXiv preprint arXiv:2402.08943 (2024)