CLAIR: CONSISTENT LLM ADAPTATION VIA INDEX-FREE REPROGRAMMING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) have been adapted to time-series forecasting through patch-based reprogramming and task-specific conditioning. However, aligning contextualized patch representations requires choosing which patches to compare, while representation agreement alone does not ensure forecast agreement. We propose CLAIR, an asymmetric teacher-guided framework for Consistent LLM Adaptation via Index-free Reprogramming that couples patch-level transport alignment (PTA) with forecast-level consistency (FC). A clean-view exponential moving average (EMA) teacher provides a shared reference for a student processing a mildly perturbed view of the same input. PTA uses entropically regularized optimal transport to establish soft correspondences between post-backbone student and teacher patch representations, while FC encourages agreement between their predictions. Together with structure-aware input conditioning, these objectives complement the supervised forecasting loss without reordering the patch states used for prediction. Experiments across long-term, short-term, and few-shot forecasting demonstrate competitive performance against representative LLM-based and non-LLM forecasters, with mean relative reductions in horizon-averaged MSE of 3.1% and 3.8% across six benchmarks under the 10% and 5% few-shot settings, respectively, relative to the best evaluated LLM baseline on each dataset. Controlled comparisons on ETTh1 and ETTm1 show that PTA and index-wise alignment perform comparably without FC. When both are combined with FC, PTA reduces horizon-averaged MSE by 2.1% and 2.6%, respectively, relative to index-wise alignment.

1 INTRODUCTION

Time-series forecasting supports decision-making in domains such as energy, traffic, finance, climate, healthcare, and industrial monitoring. Real-world series often exhibit non-stationary dynamics, noisy observations, and dependencies spanning multiple temporal scales, making it difficult to learn representations that generalize to future observations. In parallel, recent work has explored adapting large language models (LLMs) as forecasting backbones, motivated by the possibility of reusing contextual representations acquired through large-scale pretraining (Zhou et al., 2023; Jin et al., 2024). However, numerical time series differ substantially from natural language in their structure and temporal organization, making effective adaptation nontrivial. Existing approaches address this gap through patch-based input adaptation (Zhou et al., 2023; Jin et al., 2024), semantic or structural conditioning, and explicit temporal modeling (Pan et al., 2024; Hu et al., 2025; Liu et al., 2026). Cross-modal supervision also extends to intermediate representations and final predictions through distinct feature- and output-level objectives (Liu et al., 2025b; Wang et al., 2025).

Within this line of work, a further design question concerns how patch-level representation correspondence should be coupled with forecast-level consistency. Self-attention incorporates information from other sequence positions into each patch representation (Vaswani et al., 2017). Our token-replacement diagnostic makes this contextualization explicit. Replacing a single patch embedding produces a localized change before the backbone, but affects representations at other temporal positions afterward (Figure 1 (a),(b)). Although post-backbone patch states remain indexed by their input positions, they encode context beyond the corresponding local segments. Moreover, when aligning patch representations defined over the same temporal grid, index-wise alignment provides a natural correspondence based on position, whereas representation-dependent matching

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.

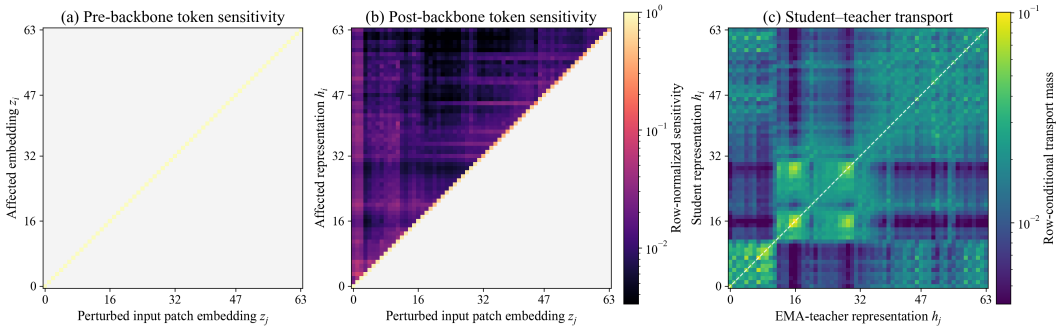


Figure 1: Contextualization of patch representations. (a, b) Sensitivity of patch representation at position i (y-axis) to replacing the input patch at position j (x-axis). (a) Changes are localized before the backbone, yielding a diagonal pattern. (b) After self-attention, changes extend across temporal positions, reflecting contextualization beyond individual segments. (c) Learned student–teacher patch correspondence in CLAIR for the same sequence, including associations beyond same-index matching (dashed diagonal).

allows cross-position associations based on contextualized features. However, the predictive value of such non-local matches cannot be assumed. Representational similarity and functional similarity need not coincide (Braun et al., 2025). Conversely, forecast-level agreement does not explicitly specify which individual patch representations should correspond.

To address this question, we propose CLAIR, an asymmetric teacher-guided framework for Consistent LLM Adaptation via Index-free Reprogramming. Here, index-free refers to the representation-level correspondence used for alignment, rather than to the removal of temporal indices or patch order. Following weight-averaged teacher–student learning (Tarvainen & Valpola, 2017), a clean-view exponential moving average (EMA) teacher processes the original normalized input and provides a common stop-gradient reference for a student processing a mildly perturbed view of the same series. Patch-level transport alignment (PTA) uses entropically regularized optimal transport to establish soft correspondences between normalized post-backbone student and teacher patch representations, while forecast-level consistency (FC) encourages agreement between their normalized predictions. PTA affects only the correspondence used for representation alignment. The patch states supplied to the forecasting head remain in their original temporal order. Both objectives complement the supervised forecasting loss. Learnable soft prompts and a multi-scale structural anchor provide additional task- and sequence-level conditioning. The EMA teacher and transport computation are used only during training, and inference requires only the student model.

Our main contributions are summarized as follows:

- We introduce CLAIR, an asymmetric teacher–student framework that couples PTA with FC under a shared clean-view EMA reference, while preserving the ordered patch representations used for forecasting.
- Controlled comparisons on ETTh1 and ETTm1 show that PTA and index-wise alignment perform comparably without FC, whereas PTA achieves lower forecasting errors when both are combined with FC. These results support jointly designing patch correspondence and forecast-level consistency.
- Evaluations across long-term, short-term, and few-shot forecasting demonstrate improvements over representative LLM-based baselines in most evaluated settings. The teacher and transport computation are used only during training, so inference requires only the student model.

2 Related Work

2.1 Teacher–Student and Non-contrastive Learning

Teacher–student learning encompasses both prediction consistency and representation-level supervision. Weight-averaged teacher predictions provide consistency targets under perturbations (Tarvainen & Valpola, 2017). At the representation level, non-contrastive approaches learn across views without explicit negative pairs, using asymmetric branches with momentum targets or stop-gradient and a prediction head (Grill et al., 2020; Chen & He, 2021). These approaches differ in both how reference targets are updated and whether the learning signal acts on representations or final predictions. In time-series learning, teacher supervision has been incorporated into hierarchical masked modeling for forecasting (Zhao et al., 2024), reconstruction-based representation learning (Skaf et al., 2025), and change-point modeling (Fan et al., 2025). Beyond target construction, distillation also requires specifying the correspondence between the quantities being matched. Language-model distillation illustrates different choices: predefined layer and token-position mappings support hidden-state matching (Jiao et al., 2020), whereas optimal transport enables token- and sequence-level alignment of logit distributions across different tokenizers (Cui et al., 2025).

2.2 LLM-based Time-Series Forecasting and Adaptation

LLM-based time-series forecasting adapts pretrained language models through numerical input interfaces and task-specific parameter updates. OFA projects time-series patches into the embedding space and selectively updates pretrained parameters (Zhou et al., 2023), while LLM4TS also employs patch-based adaptation (Chang et al., 2025). Time-LLM reprograms patches through text prototypes and incorporates natural-language prefix prompts (Jin et al., 2024). S^2IP-LLM derives semantic anchors from pretrained word embeddings for alignment and prompting (Pan et al., 2024). Structural and temporal modeling further enrich these interfaces. FSCA uses graph-based context alignment to model structural and logical relations between time-series tokens and language prompts, with modules that can be integrated into multiple LLM layers (Hu et al., 2025). SE-LLM enhances temporal–semantic representations and introduces an attention-integrated Time-Adapter for temporal dependency modeling (Liu et al., 2026). Cross-modal approaches also differ in the representations and objectives used for supervision. CALF regularizes intermediate features and outputs across temporal and language-space branches (Liu et al., 2025b). TimeCMA aligns time-series representations with LLM-derived prompt embeddings (Liu et al., 2025a), while TS-HTFA combines intermediate-feature alignment with output-level optimal transport (Wang et al., 2025). Whereas these alignment-oriented approaches primarily differ in the representations and supervision objectives, CLAIR treats patch correspondence itself as an explicit design choice. Specifically, it establishes flexible patch-level transport between clean and perturbed views of the same series, while examining how flexible versus index-wise alignment interacts with FC.

3 Method

CLAIR couples PTA with FC under a shared clean-view EMA teacher, guiding a student whose extracted patch and anchor windows are mildly perturbed. Soft prompts and a multi-scale structural anchor provide task-level and sequence-specific input conditioning, respectively. Figure 2 presents an overview of the architecture.

3.1 Problem Setting

Given multivariate input sequences X, the goal is to generate forecasts $\hat{Y}$ for the ground-truth future sequences Y, where

$$X \in \mathbb{R}^{B\times L\times M}, \qquad Y, \hat{Y} \in \mathbb{R}^{B\times H\times M}. \tag{1}$$

Here, B, L, M, and H denote the batch size, input length, number of variables, and forecasting horizon, respectively. Following the channel-independent setting, we reshape X into $U = BM$ univariate sequences $x^{(u)} \in \mathbb{R}^L$ and normalize each using its input-window statistics. Predictions are inverse-normalized using the same statistics and reshaped into the multivariate output format.

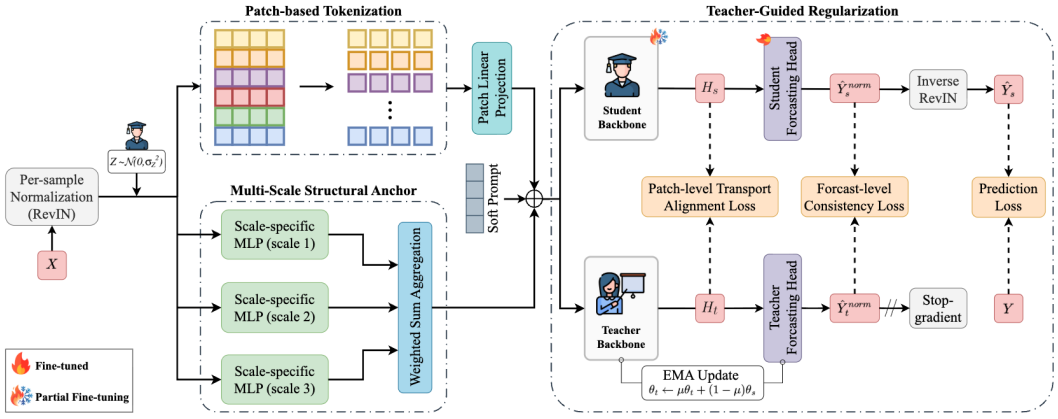


Figure 2: Overall architecture of CLAIR. Time-series patches are reprogrammed into the backbone's embedding space with soft prompts and a multi-scale structural anchor. A perturbed student is guided by a clean-view EMA teacher via representation-level and forecast-level regularization, while only the student is used at inference.

3.2 CLAIR Architecture

Backbone and Patch Reprogramming. We use GPT-2 (Radford et al., 2019) as the pretrained backbone. Its self-attention and feed-forward parameters are frozen, while its positional embeddings and layer-normalization parameters are updated together with the patch projection, input-conditioning modules, and forecasting head. Each normalized sequence $\bar{x}^{(u)}$ is divided into N overlapping patches of length P and stride S. For each patch $p_i^{(u)} \in \mathbb{R}^P$, a shared linear layer projects it into the GPT-2 embedding space:

$$z_i^{(u)} = W_p p_i^{(u)} + b_p, \qquad W_p \in \mathbb{R}^{D\times P}, \quad b_p, z_i^{(u)} \in \mathbb{R}^D, \tag{2}$$

where D is the GPT-2 hidden dimension. The projected patch tokens are stacked as

$$Z_{\text{patch}}^{(u)} = [z_1^{(u)}, \ldots, z_N^{(u)}]^\top \in \mathbb{R}^{N\times D}. \tag{3}$$

Input Conditioning. Learnable soft prompts $P_{\text{soft}} \in \mathbb{R}^{n_p\times D}$ provide an input-independent task prefix shared across sequences. In contrast, the structural anchor $a_{\text{struct}}^{(u)} \in \mathbb{R}^D$ summarizes the current input across K temporal scales into a single context token. At scale k, a two-layer MLP $f_k(\cdot)$ encodes sliding windows $w_i^{(u,k)}$, whose features are aggregated as

$$\begin{aligned} s^{(u,k)} &= \frac{\alpha_k}{T_k}\sum_{i=1}^{T_k} f_k\left(w_i^{(u,k)}\right), \\ a_{\text{struct}}^{(u)} &= \sum_{k=1}^{K} \pi_k s^{(u,k)}, \qquad \pi_k = \frac{\exp(\beta_k)}{\sum_{j=1}^{K}\exp(\beta_j)}. \end{aligned} \tag{4}$$

Here, T_k is the window count, α_k rescales each scale-level summary, and π_k provides normalized mixing across scales. The gains α_k and logits β_k are learned and shared across sequences.

The prompts and anchor are prepended to the patch embeddings:

$$Z_{\text{in}}^{(u)} = \left[P_{\text{soft}} \parallel a_{\text{struct}}^{(u)} \parallel Z_{\text{patch}}^{(u)}\right] \in \mathbb{R}^{(n_p+1+N)\times D}, \tag{5}$$

where $\parallel$ denotes concatenation along the token dimension. These prefixes condition patch representations through GPT-2 attention. Only the patch hidden states $H_{\text{patch}} \in \mathbb{R}^{U\times N\times D}$ are retained for PTA and the forecasting head g, which produces normalized forecasts:

$$\hat{Y}^{\text{norm}} = g(\text{flatten}(H_{\text{patch}})), \qquad \hat{Y}^{\text{norm}} \in \mathbb{R}^{U\times H}. \tag{6}$$

Flattening combines the patch and feature dimensions within each sequence in their original order, which is unchanged by PTA.

3.3 TEACHER-GUIDED REGULARIZATION

Asymmetric EMA Teacher. The student F_{θ_s} and teacher F_{θ_t} share the same architecture. The teacher is initialized from the student and updated after each student optimization step as

$$\theta_t \leftarrow \mu\theta_t + (1-\mu)\theta_s, \tag{7}$$

where μ is the EMA momentum. The update covers all trainable student parameters, including the forecasting head and the unfrozen GPT-2 parameters. The teacher receives no gradients, and its outputs are treated as stop-gradient targets.

For each normalized sequence, let $P^{(u)} = [p_1^{(u)}, \dots, p_N^{(u)}]$ denote the patch windows and $W^{(u,k)} = [w_1^{(u,k)}, \dots, w_{T_k}^{(u,k)}]$ the windows at anchor scale k. The two branches process asymmetric versions of these extracted windows:

$$\begin{aligned} P_s^{(u)} &= \mathcal{A}_p\left(P^{(u)}\right), \qquad P_t^{(u)} = P^{(u)}, \\ W_s^{(u,k)} &= \mathcal{A}_k\left(W^{(u,k)}\right), \quad W_t^{(u,k)} = W^{(u,k)}, \qquad k = 1, \dots, K. \end{aligned} \tag{8}$$

Here, $\mathcal{A}_p$ and $\mathcal{A}_k$ denote stochastic perturbation operators applied independently to the extracted patch windows and the windows at anchor scale k, respectively. Specifically, these operators apply Gaussian noise and elementwise dropout only to the student branch, before patch projection and anchor encoding. The corresponding perturbed and unperturbed windows are then processed by the student and teacher, producing $H_s, \hat{Y}_s^{\text{norm}}$ and $H_t, \hat{Y}_t^{\text{norm}}$, respectively. Here, $H_s, H_t \in \mathbb{R}^{U\times N\times D}$ and $\hat{Y}_s^{\text{norm}}, \hat{Y}_t^{\text{norm}} \in \mathbb{R}^{U\times H}$. The clean view refers to the unperturbed extracted windows, rather than to an assumption that the original observations are noise-free.

Patch-level Transport Alignment. Although the two branches retain the same patch order, their contextualized post-backbone representations need not be compared only through fixed same-index correspondence. PTA therefore permits flexible representation-level correspondence.

We first normalize the patch representations along the embedding dimension:

$$\tilde{H}_s = \text{normalize}(H_s), \qquad \tilde{H}_t = \text{normalize}(H_t). \tag{9}$$

For each sequence u, the pairwise cost matrix is

$$C_{ij}^{(u)} = \left\| \tilde{h}_{s,i}^{(u)} - \text{sg}\left(\tilde{h}_{t,j}^{(u)}\right) \right\|_2^2, \tag{10}$$

where $\text{sg}(\cdot)$ denotes stop-gradient.

With uniform marginals $r = c = \frac{1}{N}\mathbf{1}_N$, the transport plan is obtained by

$$\Pi^{\star(u)} = \arg\min_{\Pi\in\mathcal{U}(r,c)} \left\langle \Pi, C^{(u)} \right\rangle + \varepsilon_{\text{OT}} \sum_{i,j} \Pi_{ij} \left(\log \Pi_{ij} - 1\right), \tag{11}$$

where $\mathcal{U}(r,c)$ is the set of transport plans with marginals r and c, and ε_{OT} is the entropic regularization coefficient. The transport plan is computed using Sinkhorn iterations (Cuturi, 2013).

The PTA loss is

$$\mathcal{L}_{\text{PTA}} = \frac{1}{U} \sum_{u=1}^{U} \left\langle \Pi^{\star(u)}, C^{(u)} \right\rangle. \tag{12}$$

Importantly, the transport plan is used only to define correspondence in the representation-level loss; it does not reorder or aggregate the hidden states passed to the forecasting head.

Forecast-level Consistency. Because PTA permits flexible correspondence at the representation level, a low transport cost alone does not require the forecasts produced from the ordered hidden states to agree. FC therefore provides a complementary forecast-level constraint on the student and teacher predictions. Applying the corresponding forecasting heads in equation 6 to H_s and H_t produces $\hat{Y}_s^{\text{norm}}$ and $\hat{Y}_t^{\text{norm}}$, respectively:

$$\mathcal{L}_{\text{FC}} = \frac{1}{UH} \left\| \hat{Y}_s^{\text{norm}} - \text{sg}\left(\hat{Y}_t^{\text{norm}}\right) \right\|_F^2. \tag{13}$$

3.4 Optimization and Inference

The inverse-normalized student forecast $\hat{Y}_s \in \mathbb{R}^{B\times H\times M}$ is optimized using a task-specific supervised forecasting loss:

$$\mathcal{L}_{\text{pred}} = \ell_{\text{pred}}\left(\hat{Y}_s, Y\right), \tag{14}$$

where ℓ_{pred} denotes the supervised forecasting criterion used for the corresponding forecasting setting.

The final objective is

$$\mathcal{L} = \mathcal{L}_{\text{pred}} + \gamma_e\left(\lambda_{\text{PTA}}\mathcal{L}_{\text{PTA}} + \lambda_{\text{FC}}\mathcal{L}_{\text{FC}}\right), \tag{15}$$

where λ_{PTA} and λ_{FC} control the two auxiliary objectives, and

$$\gamma_e = \min\left(1, \frac{e+1}{E_{\text{warm}}}\right) \tag{16}$$

gradually introduces them during the first E_{warm} epochs. At inference, only the student model is retained. Student-side perturbation, the EMA teacher, Sinkhorn transport, and the auxiliary consistency losses are removed.

4 Experiments

We evaluate CLAIR across long-term, short-term, and few-shot forecasting settings. Controlled comparisons examine PTA and index-wise alignment with and without FC. Component ablations and transport diagnostics further characterize the contributions of individual components and the learned correspondence patterns. Additional computational profiling in the appendix analyzes training and inference costs.

4.1 Experimental Setup

Datasets. For long-term and few-shot forecasting, we evaluate CLAIR on six widely used multivariate benchmarks: Weather (Wu et al., 2021), Electricity (Wu et al., 2021), and four ETT datasets, namely ETTh1, ETTh2, ETTm1, and ETTm2 (Zhou et al., 2021). For short-term forecasting, we use the M4 benchmark (Makridakis et al., 2018). These datasets cover heterogeneous temporal resolutions, sampling rates, and domain characteristics.

Baselines. We compare CLAIR with representative LLM-based and non-LLM forecasting methods. LLM-based baselines include Time-LLM (Jin et al., 2024), One Fits All (OFA) (Zhou et al., 2023), S^2IP-LLM (Pan et al., 2024), CALF (Liu et al., 2025b), FSCA (Hu et al., 2025), and SE-LLM (Liu et al., 2026). These methods cover patch-based adaptation, semantic and structural conditioning, temporal modeling, and cross-modal alignment. Non-LLM baselines include DLinear (Zeng et al., 2023), iTransformer (Liu et al., 2024), and PatchTST (Nie et al., 2023).

Experimental Protocol. Following standard benchmarks, we fix the lookback window to $L = 512$ and evaluate performance across four future horizons $H \in \{96, 192, 336, 720\}$. Forecasting accuracy on multivariate series is evaluated using Mean Squared Error (MSE) and Mean Absolute Error (MAE). For short-term forecasting on the M4 benchmark, we strictly follow the official frequency-specific protocol across diverse sampling granularities (Yearly, Quarterly, Monthly, Weekly, Daily, and Hourly), reporting Symmetric Mean Absolute Percentage Error (SMAPE), Mean Absolute Scaled Error (MASE), and Overall Weighted Average (OWA).

4.2 Forecasting Performance

Long-term Forecasting. Table 1 summarizes multivariate long-term forecasting results averaged across prediction horizons. CLAIR achieves the lowest MSE on Weather and all ETT benchmarks, with similarly strong MAE results. This pattern supports the effectiveness of the framework across different temporal resolutions, while FSCA's stronger performance on Electricity indicates that the relative advantage remains dataset-dependent.

Table 1: Long-term forecasting results averaged over prediction horizons H.

	CLAIR		Time-LLM		S^{2}IP-LLM		OFA		CALF		FSCA		SE-LLM		DLinear		iTransformer		PatchTST	
Dataset	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Weather	**0.2261**	**0.2642**	0.2346	0.2723	0.2283	0.2656	0.2285	0.2667	0.2363	0.2719	0.2281	0.2678	0.2314	0.2676	0.2416	0.2941	0.2443	0.2775	0.2338	0.2748
ETTh1	**0.4094**	**0.4287**	0.4409	0.4516	0.4154	0.4346	0.4265	0.4356	0.4281	0.4433	0.4180	0.4332	0.4258	0.4379	0.4184	0.4386	0.4485	0.4612	0.4600	0.4643
ETTh2	**0.3564**	**0.3971**	0.3686	0.4056	0.3624	0.4052	0.3742	0.4065	0.3666	0.4020	0.3597	0.3993	0.3823	0.4113	0.4994	0.4803	0.3846	0.4154	0.4036	0.4305
ETTm1	**0.3493**	0.3820	0.3790	0.3966	0.3509	0.3835	0.3596	0.3899	0.3580	0.3869	0.3674	0.3948	0.3684	0.3948	0.3587	**0.3818**	0.3689	0.3989	0.3770	0.4048
ETTm2	**0.2562**	**0.3200**	0.2649	0.3273	0.2601	0.3236	0.2661	0.3288	0.2628	0.3203	0.2581	0.3212	0.2765	0.3280	0.2767	0.3422	0.2718	0.3312	0.2797	0.3353
Electricity	0.1655	0.2600	0.1719	0.2720	0.1687	0.2660	0.1664	0.2630	0.1664	0.2611	**0.1635**	**0.2574**	0.1688	0.2627	0.1670	0.2675	0.1704	0.2668	0.1663	0.2664

Table 2: Short-term forecasting results on the M4 dataset averaged over prediction horizons H.

		CLAIR	Time-LLM	S^{2}IP-LLM	OFA	CALF	FSCA	SE-LLM	DLinear	iTransformer	PatchTST
Average	SMAPE	12.008	12.522	12.314	12.256	12.193	**12.003**	12.145	12.507	13.359	12.873
	MASE	**1.628**	1.704	1.684	1.714	1.632	1.637	1.656	1.684	1.786	1.757
	OWA	**0.868**	0.907	0.894	0.900	0.876	0.871	0.881	0.901	0.959	0.934

Table 3: Few-shot forecasting results with 10% training data averaged over prediction horizons H.

	CLAIR		Time-LLM		S^{2}IP-LLM		OFA		CALF		FSCA		SE-LLM		DLinear		iTransformer		PatchTST	
Dataset	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Weather	**0.2372**	**0.2766**	0.2491	0.2874	0.2404	0.2787	0.2430	0.2793	0.2573	0.2937	0.2425	0.2815	0.2512	0.2844	0.2419	0.2894	0.2893	0.3163	0.2529	0.2893
ETTh1	0.5589	0.5064	0.6879	0.5585	0.7431	0.5759	0.5912	0.5263	0.7168	0.5860	0.6337	0.5389	0.6420	0.5427	**0.5308**	**0.5024**	0.7026	0.5787	0.7259	0.5966
ETTh2	**0.3846**	**0.4226**	0.4041	0.4333	0.4647	0.4660	0.4449	0.4578	0.4207	0.4422	0.4112	0.4345	0.3951	0.4272	0.4137	0.4459	0.4262	0.4473	0.4555	0.4659
ETTm1	0.4482	0.4335	0.4743	0.4534	0.4834	0.4600	0.4599	0.4391	0.4984	0.4721	0.4924	0.4540	0.6259	0.5152	**0.4131**	**0.4275**	0.5705	0.5101	0.6023	0.5033
ETTm2	**0.2611**	**0.3217**	0.3017	0.3515	0.3077	0.3531	0.2899	0.3358	0.2944	0.3427	0.2862	0.3341	0.2892	0.3419	0.2965	0.3507	0.3147	0.3600	0.3299	0.3646
Electricity	0.1819	0.2778	0.1860	0.2845	0.1788	0.2743	**0.1782**	**0.2723**	0.1932	0.2909	0.1820	0.2762	0.1892	0.2829	0.2201	0.3287	0.1946	0.2913	0.1848	0.2841

Table 4: Few-shot forecasting results with 5% training data averaged over prediction horizons H.

	CLAIR		Time-LLM		S^{2}IP-LLM		OFA		CALF		FSCA		SE-LLM		DLinear		iTransformer		PatchTST	
Dataset	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Weather	**0.2642**	**0.3046**	0.2778	0.3096	0.2918	0.3244	0.2720	0.3089	0.2758	0.3113	0.2670	0.3055	0.2745	0.3087	0.2724	0.3245	0.2917	0.3272	0.2846	0.3180
ETTh1	0.6719	0.5509	0.8125	0.5919	1.0053	0.6719	0.8402	0.6190	0.8672	0.6376	0.7419	0.5887	0.7752	0.6030	**0.6269**	**0.5478**	0.8137	0.6157	0.8857	0.6572
ETTh2	**0.3729**	**0.4235**	0.5466	0.5100	0.6167	0.5454	0.4697	0.4844	0.4411	0.4591	0.3754	0.4248	0.4600	0.4807	0.4872	0.5090	0.4313	0.4516	0.4052	0.4274
ETTm1	0.4604	**0.4451**	0.5077	0.4729	0.5201	0.4860	0.4939	0.4625	0.5242	0.4850	0.4970	0.4630	0.6061	0.5066	**0.4464**	0.4463	0.5634	0.5047	0.5703	0.5047
ETTm2	**0.2933**	**0.3405**	0.3228	0.3618	0.3199	0.3624	0.3174	0.3517	0.3197	0.3598	0.3064	0.3473	0.3083	0.3514	0.3263	0.3762	0.3085	0.3573	0.3533	0.3787
Electricity	**0.1806**	**0.2757**	0.1890	0.2888	0.1831	0.2801	0.1850	0.2829	0.2281	0.3231	0.1821	0.2787	0.1967	0.2911	0.2958	0.3935	0.2102	0.3090	0.1903	0.2901

Short-term Forecasting. Table 2 reports results on the M4 benchmark. CLAIR combines the lowest MASE with near-best SMAPE, yielding the lowest OWA among the compared methods. This combination indicates a favorable balance between scaled absolute and percentage errors under the frequency-specific protocol, supporting the applicability of the framework beyond multivariate long-term forecasting.

Few-shot Forecasting. Tables 3 and 4 report horizon-averaged forecasting results using 10% and 5% of the training data, respectively. CLAIR outperforms the evaluated LLM-based baselines on most datasets in the 10% setting and across all datasets in the 5% setting. This pattern supports the effectiveness of the proposed adaptation framework when task-specific supervision is limited, showing that its competitive performance persists under more restrictive data budgets.

4.3 Patch Correspondence and Forecast Consistency

We compare three representation-guidance settings, namely no guidance, index-wise alignment, and PTA, both with and without FC, while keeping the student architecture, input perturbations, and supervised forecasting objective fixed. Index-wise alignment uses the same normalized post-backbone representations as PTA but matches patches at identical temporal positions:

$$\mathcal{L}_{\text{Index}} = \frac{1}{UN} \sum_{u=1}^{U} \sum_{i=1}^{N} \left\| \tilde{h}_{s,i}^{(u)} - \text{sg}\left(\tilde{h}_{t,i}^{(u)} \right) \right\|_2^2, \tag{17}$$

where $U = BM$, N is the number of patch tokens, and $\text{sg}(\cdot)$ denotes stop-gradient. Table 5 shows that the relative benefit of PTA over index-wise alignment is larger when FC is included. Without FC, PTA and index-wise alignment yield similar horizon-averaged MSE: 0.4307 versus 0.4325 on ETTh1 and 0.3660 versus 0.3652 on ETTm1. When both are paired with FC, PTA achieves lower MSE and MAE on both datasets, with relative MSE reductions of 2.1% and 2.6%, respectively. These controlled results support coupling flexible patch correspondence with forecast-level consistency under the evaluated settings.

Table 5: Comparison of patch correspondence and forecast consistency on ETTh1 and ETTm1 averaged over H.

	ETTh1		ETTm1	
Model Variant	MSE	MAE	MSE	MAE
Supervised only	0.4434	0.4581	0.3715	0.4014
FC only	0.4332	0.4455	0.3671	0.3912
Index only	0.4325	0.4443	0.3652	0.3883
Index + FC	0.4183	0.4357	0.3585	0.3876
PTA only	0.4307	0.4426	0.3660	0.3895
PTA + FC (CLAIR)	**0.4094**	**0.4287**	**0.3493**	**0.3820**

Table 6: Component ablation study evaluating the contributions of modules on ETTh1 and ETTm1 averaged over H.

	ETTh1		ETTm1	
Model Variant	MSE	MAE	MSE	MAE
No EMA ($\mu = 0$)	0.4357	0.4457	0.3681	0.3923
w/o Gaussian Noise	0.4272	0.4409	0.3598	0.3876
w/o Soft Prompts	0.4119	0.4302	0.3516	0.3829
w/o Structural Anchor	0.4129	0.4300	0.3528	0.3835
Frozen WPE+LN	0.4156	0.4333	0.3577	0.3854
Full Model (CLAIR)	**0.4094**	**0.4287**	**0.3493**	**0.3820**

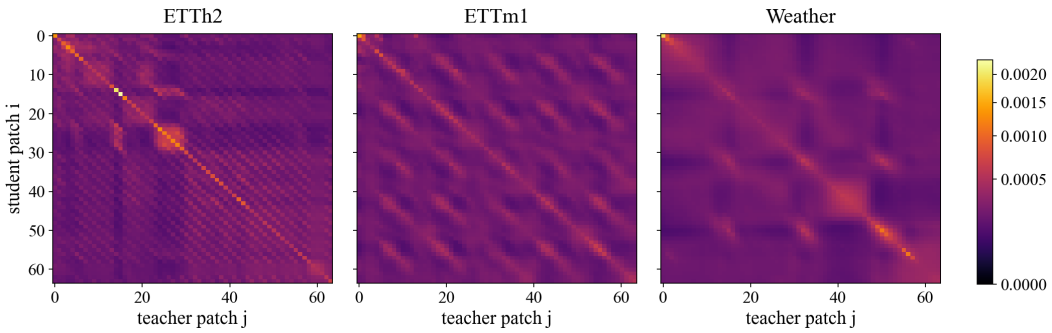


Figure 3: Batch-averaged unconstrained PTA transport plans π at horizon $H = 96$ on ETTh2 (left), ETTm1 (middle), and Weather (right). Rows and columns index student and teacher patches, respectively.

4.4 Ablation Study

Table 6 summarizes component ablations on ETTh1 and ETTm1, excluding the controlled PTA and FC comparisons reported in Section 4.3. The full model achieves the lowest horizon-averaged MSE and MAE among the evaluated variants on both datasets. Disabling EMA averaging ($\mu = 0$) produces the largest degradation, while removing student-side Gaussian noise also worsens performance. Removing the soft prompts or structural anchor results in smaller but consistent increases in error, supporting the contribution of task-level and sequence-specific conditioning. Freezing GPT-2's positional embeddings and layer-normalization parameters also degrades performance, indicating that selectively adapting these backbone parameters is beneficial in the evaluated settings.

4.5 Temporal Structure of Unconstrained PTA

We examine whether unconstrained PTA exhibits temporal structure despite using neither an explicit temporal penalty nor a local-window constraint. We first visualize mean transport plans on ETTh2, ETTm1, and Weather at H=96, and then analyze locality and lag–mass structure on ETTh2 across forecasting horizons $H \in \{96, 192, 336, 720\}$.

Mean Transport Plans across Datasets. At the fixed forecast horizon H=96, we average the student–teacher transport plans over the evaluated sequences separately for ETTh2, ETTm1, and Weather. Figure 3 shows the resulting dataset-specific mean transport plans, denoted by $\bar{\Pi}$. Although the detailed patterns vary across datasets, the maps exhibit near-diagonal concentrations together with structured off-diagonal mass. These patterns illustrate that PTA assigns mass to both local and non-local correspondences rather than restricting alignment to identical temporal positions.

Quantitative Locality on ETTh2. We focus on ETTh2 for a detailed analysis of the near-diagonal concentration and repeated off-diagonal bands visible in Figure 3. We first quantify locality across forecasting horizons using the expected patch-index displacement, $\mathbb{E}_{\bar{\Pi}}[|i-j|] = \sum_{i,j} \bar{\Pi}_{ij}|i-j|$,

Table 7: Batch-averaged locality metrics of unconstrained PTA on ETTh2.

H	$\mathbb{E}[\lvert i-j\rvert]$	Unif. baseline	Band $\lvert i-j\rvert \leq 1$	Band ≤ 8
96	19.19	21.33	0.097	0.326
192	18.94	21.33	0.100	0.331
336	18.95	21.33	0.101	0.331
720	18.85	21.33	0.104	0.334

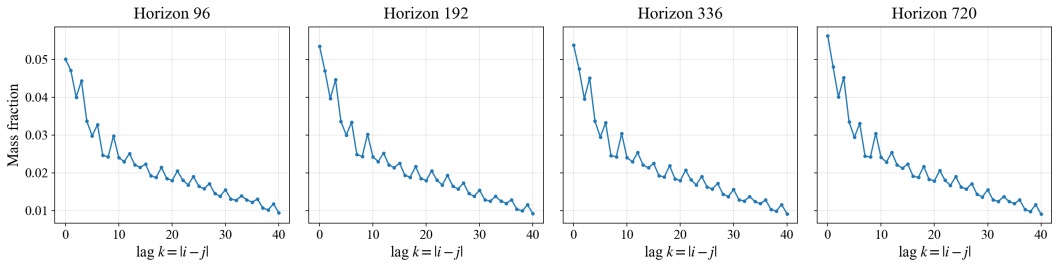


Figure 4: Transport mass fraction vs. temporal lag $k{=}|i{-}j|$ under the batch-averaged plan π on ETTh2.

and the transport mass within near-diagonal bands $|i-j| \leq b$. Table 7 shows that the expected displacement remains below the uniform-matching baseline across the evaluated horizons. At the same time, the narrow band $|i-j| \leq 1$ accounts for only a small fraction of the total transport mass, and substantial mass remains outside the broader band $|i-j| \leq 8$. Together, these results indicate a modest preference for nearby patches relative to uniform matching, while retaining substantial non-local correspondence. The similar locality statistics across horizons suggest that this balance extends beyond the single forecast horizon visualized in Figure 3.

Lag–Mass Structure on ETTh2. To characterize the temporal organization of the averaged transport plans, we aggregate transport mass by absolute patch-index lag: $M(k) = \sum_{|i-j|=k} \overline{\Pi}_{ij}$. Figure 4 shows an overall decreasing trend with recurring peaks at approximately three-patch intervals across the evaluated horizons. The overall decline should be interpreted cautiously because the number of available patch pairs also decreases at larger lags. Under the hourly sampling rate of ETTh2 and patch stride $S{=}8$, a three-patch separation corresponds to 24 hours. The recurring peaks are therefore consistent with daily-scale structure in the averaged transport patterns. These diagnostics characterize the temporal organization of PTA, complementing the controlled comparisons that evaluate the predictive effects of correspondence and forecast-level consistency.

5 Conclusion

This paper presents CLAIR, an asymmetric teacher-guided reprogramming framework that couples flexible patch correspondence with forecast-level consistency for LLM-based time-series forecasting. A clean-view EMA teacher guides a perturbed student through PTA and FC, complemented by soft prompts and a multi-scale structural anchor, while preserving the ordered patch states used for forecasting. CLAIR achieves the lowest horizon-averaged MSE on five of the six long-term benchmarks and the lowest aggregate OWA on M4 among the evaluated methods. Controlled comparisons show that FC enhances the relative benefit of flexible over index-wise correspondence, while ablations support the contributions of teacher guidance and input conditioning. Transport diagnostics further reveal both local and non-local correspondence patterns, including recurring lag structures consistent with daily-scale behavior on ETTh2. Together, these results support coupling flexible patch correspondence with forecast-level consistency in the evaluated LLM adaptation setting. The EMA teacher and Sinkhorn alignment add training-time overhead but are removed at inference. Future work includes more efficient training, explicit cross-variate modeling, and evaluating the predictive effects of temporally constrained alignment.

AI USE STATEMENT

In this work, we used generative AI tools to assist with implementing and debugging software code, including the proposed method; language editing for clarity and conciseness; drafting and refining selected manuscript sections, including the abstract, main-text descriptions, and figure captions; retrieval and discovery of related work; research ideation and execution; and organizing the presentation of technical content. We have not used generative AI tools to generate synthetic data, formulate mathematical claims or proofs, or generate or modify scientific figures; the remaining disclosure tasks are not applicable to this work. We have reviewed all AI-assisted work: all AI-assisted code was inspected, integrated, tested by the authors, and verified against the proposed method and intended functionality; references found via AI-assisted retrieval were read and verified before citation; AI-suggested ideas were critically evaluated and validated; and the reported results and interpretations were independently checked. We also checked the AI-assisted text for unsupported claims, fabricated results, plagiarism, and unintended changes to the technical meaning. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies large language model-based time-series forecasting on publicly available benchmark datasets. The experiments do not involve human subjects, personal data, or sensitive information, and the proposed method is not deployed to make decisions about individuals or high-stakes real-world situations. Although forecasting errors may pose risks in potential downstream applications, such applications are outside the scope of this study. We therefore do not identify any direct societal harms arising from the experiments presented in this paper.

REPRODUCIBILITY STATEMENT

All datasets used in our experiments, including the ETT, Weather, Electricity, and M4 benchmarks, are publicly available. Deep learning baselines use the corresponding open-source implementations in the Time-Series-Library, while LLM-based baselines use their publicly released codebases whenever available. We follow released task-specific configurations when provided; otherwise, we use comparable configurations under the common evaluation protocol described in the paper. For methods without an official implementation, we follow the methodological descriptions in their respective papers. Detailed experimental settings, hyperparameters, ablation results, and additional analyses are provided in the paper. The source code and complete experimental configurations will be released upon acceptance to facilitate reproduction and independent verification of our results.

REFERENCES

Lukas Braun, Erin Grant, and Andrew M Saxe. Not all solutions are created equal: An analytical dissociation of functional and representational similarity in deep linear neural networks. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 5355–5382, 13–19 Jul 2025.

Ching Chang, Wei-Yao Wang, Wen-Chih Peng, and Tien-Fu Chen. LLM4TS: aligning pre-trained llms as data-efficient time-series forecasters. *ACM Trans. Intell. Syst. Technol.*, 16(3):60:1–60:20, 2025.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pp. 15750–15758, 2021.

Xiao Cui, Mo Zhu, Yulei Qin, Liang Xie, Wengang Zhou, and Houqiang Li. Multi-level optimal transport for universal cross-tokenizer knowledge distillation on language models. In *Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025*, pp. 23724–23732, 2025.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pp. 2292–2300, 2013.

Jinxiao Fan, Pengfei Wang, Liang Liu, and Huadong Ma. Self-optimizing teacher and auto-matching student framework for change-point representation learning in time series forecasting. *ACM Trans. Intell. Syst. Technol.*, 16(3):72:1–72:25, 2025.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent - A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Yuxiao Hu, Qian Li, Dongxiao Zhang, Jinyue Yan, and Yuntian Chen. Context-alignment: Activating and enhancing llms capabilities in time series. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*, 2025.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pp. 4163–4174, 2020.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-llm: Time series forecasting by reprogramming large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*, 2024.

Chenxi Liu, Qianxiong Xu, Hao Miao, Sun Yang, Lingzheng Zhang, Cheng Long, Ziyue Li, and Rui Zhao. Timecma: Towards llm-empowered multivariate time series forecasting via cross-modality alignment. In *Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025*, pp. 18780–18788, 2025a.

Hao Liu, Zhang xiaoxing, Chun Yang, and Xiaobin Zhu. Semantic-enhanced time-series forecasting via large language models. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 101404–101425, 2026.

Peiyuan Liu, Hang Guo, Tao Dai, Naiqi Li, Jigang Bao, Xudong Ren, Yong Jiang, and Shu-Tao Xia. CALF: aligning llms for time series forecasting via cross-modal fine-tuning. In *Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025*, pp. 18915–18923, 2025b.

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 *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*, 2024.

Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. The m4 competition: Results, findings, conclusion and way forward. *International Journal of Forecasting*, 34(4):802–808, 2018. ISSN 0169-2070.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, 2023.

Boris N. Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-BEATS: neural basis expansion analysis for interpretable time series forecasting. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.

Zijie Pan, Yushan Jiang, Sahil Garg, Anderson Schneider, Yuriy Nevmyvaka, and Dongjin Song. S2IP-LLM: semantic space informed prompt learning with LLM for time series forecasting. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 39135–39153, 2024.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

Wadie Skaf, Mitra Baratchi, and Holger H. Hoos. Time series representations classroom (TSRC): a teacher-student-based framework for interpretability-enhanced unsupervised time series representation learning. *Mach. Learn.*, 114(12):276, 2025.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 1195–1204, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5998–6008, 2017.

Pengfei Wang, Huanran Zheng, Qi'ao Xu, Silong Dai, Yiqiao Wang, Wenjing Yue, Wei Zhu, Tianwen Qian, and Liang Zhao. TS-HTFA: advancing time-series forecasting via hierarchical text-free alignment with large language models. *Symmetry*, 17(3):401, 2025.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 22419–22430, 2021.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, 2023.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pp. 11121–11128, 2023.

Shubao Zhao, Ming Jin, Zhaoxiang Hou, Chengyi Yang, Zengxiang Li, Qingsong Wen, and Yi Wang. Himtm: Hierarchical multi-scale masked time series modeling with self-distillation for long-term forecasting. In Edoardo Serra and Francesca Spezzano (eds.), *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, CIKM 2024, Boise, ID, USA, October 21-25, 2024*, pp. 3352–3362, 2024.

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 *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pp. 11106–11115, 2021.

Tian Zhou, Peisong Niu, Xue Wang, Liang Sun, and Rong Jin. One fits all: Power general time series analysis by pretrained lm. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

A Detailed Experimental Setup

A.1 Implementation Details

All baselines are evaluated within a common experimental pipeline built on Time-Series-Library (Wu et al., 2023). Deep learning baselines follow the library's official implementations, whereas LLM-based methods use their publicly released codebases, with input lengths, prediction horizons, and evaluation metrics matched under the same protocol. CLAIR uses the first six Transformer layers of GPT-2 small (Radford et al., 2019), with patch length P=16 and stride S=8.

In both long-term and few-shot forecasting, FSCA uses a patch length of 16 and a stride of 8. To use the same 512-step input length during both training and evaluation, SE-LLM uses a 512-step input and 64-step tokens, with a 448-step overlap between the input and the shifted training target. Its backbone is replaced with GPT-2 small to reduce backbone-related differences across methods. Experiments are run on NVIDIA L40S GPUs and an AMD EPYC 9654 CPU (96 cores). Each configuration is repeated three times, and we report the averaged results.

A.2 Baselines

- **DLinear** (Zeng et al., 2023): a linear forecasting model that decomposes series into trend and seasonal components.
- **PatchTST** (Nie et al., 2023): a Transformer that tokenizes series into overlapping patches and applies temporal self-attention.
- **iTransformer** (Liu et al., 2024): a Transformer variant that applies inverted attention across variables.
- **Time-LLM** (Jin et al., 2024): reprograms time-series patches through text prototypes and Prompt-as-Prefix conditioning with a frozen LLM.
- **One Fits All (OFA)** (Zhou et al., 2023): unified LM adaptation by projecting patched continuous signals into soft tokens.
- **S^{2}IP-LLM** (Pan et al., 2024): semantic-space-informed prompt learning with LLM embedding anchors.
- **CALF** (Liu et al., 2025b): cross-modal fine-tuning that aligns numerical and language representations.
- **FSCA** (Hu et al., 2025): context-level alignment of time-series and language inputs using dual-scale GNNs and few-shot prompting.
- **SE-LLM** (Liu et al., 2026): semantic enhancement of LLM inputs with temporal patterns and short- and long-term dependency modeling.

A.3 Evaluation Metrics

For long-term and few-shot forecasting we report MSE and MAE. For short-term forecasting on M4 we report SMAPE, MASE, and OWA (Oreshkin et al., 2020), following the official M4 protocol:

$$\mathrm{MSE} = \frac{1}{H}\sum_{h=1}^{H}(Y_h - \hat{Y}_h)^2, \qquad \mathrm{MAE} = \frac{1}{H}\sum_{h=1}^{H}|Y_h - \hat{Y}_h|,$$

$$\mathrm{SMAPE} = \frac{200}{H}\sum_{h=1}^{H}\frac{|Y_h - \hat{Y}_h|}{|Y_h| + |\hat{Y}_h|}, \qquad \mathrm{MAPE} = \frac{100}{H}\sum_{h=1}^{H}\frac{|Y_h - \hat{Y}_h|}{|Y_h|},$$

$$\mathrm{MASE} = \frac{\frac{1}{H}\sum_{h=1}^{H}|Y_h - \hat{Y}_h|}{\frac{1}{T-s}\sum_{t=s+1}^{T}|X_t - X_{t-s}|}, \qquad \mathrm{OWA} = \frac{1}{2}\left[\frac{\mathrm{SMAPE}}{\mathrm{SMAPE}_{\text{Naïve2}}} + \frac{\mathrm{MASE}}{\mathrm{MASE}_{\text{Naïve2}}}\right],$$

where H is the prediction horizon, T is the length of the in-sample series, and s is the seasonal periodicity.

Table 8: Dataset statistics for long-term and short-term forecasting. Dataset size is reported as (training, validation, testing).

Tasks	Datasets	Dim.	Forecast Horizon	Dataset Size	Frequency
Long-term	Weather	21	$\{96, 192, 336, 720\}$	(36792, 5271, 10540)	10 min
	ETTh1	7	$\{96, 192, 336, 720\}$	(8545, 2881, 2881)	1 hour
	ETTh2	7	$\{96, 192, 336, 720\}$	(8545, 2881, 2881)	1 hour
	ETTm1	7	$\{96, 192, 336, 720\}$	(34465, 11521, 11521)	15 min
	ETTm2	7	$\{96, 192, 336, 720\}$	(34465, 11521, 11521)	15 min
	Electricity	321	$\{96, 192, 336, 720\}$	(18317, 2633, 5261)	1 hour
Short-term	M4-Yearly	1	6	(23000, 0, 23000)	Yearly
	M4-Quarterly	1	8	(24000, 0, 24000)	Quarterly
	M4-Monthly	1	18	(48000, 0, 48000)	Monthly
	M4-Weekly	1	13	(359, 0, 359)	Weekly
	M4-Daily	1	14	(4227, 0, 4227)	Daily
	M4-Hourly	1	48	(414, 0, 414)	Hourly

A.4 Dataset Descriptions

Table 8 summarizes dataset statistics. For long-term and few-shot forecasting we use Weather and Electricity (Wu et al., 2021) and four ETT datasets (ETTh1, ETTh2, ETTm1, ETTm2) (Zhou et al., 2021), with input length L=512 and horizons $H \in \{96, 192, 336, 720\}$. For short-term forecasting we use the M4 benchmark (Makridakis et al., 2018).

A.5 Teacher-guided Hyperparameter Configuration

Default shared settings. We use $n_p = 4$ soft prompt tokens, $K = 3$ temporal scales in the structural anchor, EMA momentum $\mu = 0.9999$, warm-up $E_{\text{warm}} = 5$, student Gaussian noise $\sigma = 0.01$, student dropout 0.05, and Sinkhorn parameters $\epsilon_{\text{OT}} = 0.1$ with 20 iterations.

Loss weights. We search both the PTA and FC loss weights, λ_{PTA} and λ_{FC}, over $\{0.05, 0.10, 0.15, 0.20\}$. The PTA weight is fixed at λ_{PTA}=0.10 across all six long-term forecasting datasets. The FC weight is set to λ_{FC}=0.10 for ETTh1 and λ_{FC}=0.05 for ETTh2, ETTm1, ETTm2, Weather, and Electricity.

A.6 Multi-scale Structural Anchor Configuration

The structural anchor aggregates temporal information across three scales, specified by window lengths in time steps rather than fixed calendar periods. For ETTh1, ETTh2, Weather, and Electricity, we use window lengths $\{12, 24, 48\}$ with corresponding strides $\{6, 12, 24\}$. For ETTm1 and ETTm2, we use window lengths $\{24, 48, 96\}$ with corresponding strides $\{12, 24, 48\}$. At each scale, each window is encoded by a scale-specific two-layer MLP with a GELU activation between the layers. The resulting features are averaged across windows within each scale, rescaled by a learnable scalar gain α_k, and combined using softmax-normalized weights π_k derived from learnable logits β_k, yielding a single structural anchor token.

B Full Long-term Forecasting Results

Table 9 reports horizon-wise multivariate long-term forecasting results corresponding to the averaged scores in the main paper.

Table 9: Multivariate long-term forecasting results (MSE / MAE) under full training (100%).

Dataset	H	CLAIR MSE	CLAIR MAE	Time-LLM MSE	Time-LLM MAE	S^{2}IP-LLM MSE	S^{2}IP-LLM MAE	OFA MSE	OFA MAE	CALF MSE	CALF MAE	FSCA MSE	FSCA MAE	SE-LLM MSE	SE-LLM MAE	DLinear MSE	DLinear MAE	iTransformer MSE	iTransformer MAE	PatchTST MSE	PatchTST MAE
Weather	96	0.1498	**0.2001**	0.1663	0.2200	0.1506	0.2005	0.1498	0.2020	0.1571	0.2071	**0.1492**	0.2025	0.1521	0.2013	0.1704	0.2298	0.1708	0.2212	0.1551	0.2123
	192	**0.1923**	**0.2417**	0.2024	0.2501	0.1941	0.2426	0.1944	0.2429	0.2036	0.2494	0.1935	0.2422	0.1977	0.2448	0.2140	0.2719	0.2118	0.2564	0.2122	0.2636
	336	0.2469	0.2848	0.2500	0.2855	**0.2450**	**0.2810**	0.2482	0.2848	0.2523	0.2873	0.2474	0.2856	0.2505	0.2854	0.2596	0.3112	0.2604	0.2933	0.2486	0.2857
	720	**0.3152**	**0.3301**	0.3197	0.3338	0.3237	0.3383	0.3216	0.3369	0.3322	0.3437	0.3221	0.3407	0.3251	0.3387	0.3225	0.3634	0.3341	0.3391	0.3192	0.3378
ETTh1	96	0.3695	0.3982	0.4136	0.4292	0.3711	0.4007	0.3833	0.4049	0.3863	0.4135	0.3755	0.3997	0.3783	0.4058	**0.3682**	**0.3981**	0.3981	0.4227	0.3834	0.4122
	192	**0.4018**	**0.4192**	0.4240	0.4337	0.4026	0.4217	0.4201	0.4273	0.4176	0.4323	0.4119	0.4217	0.4264	0.4369	0.4025	0.4205	0.4306	0.4450	0.4696	0.4675
	336	**0.4167**	**0.4300**	0.4503	0.4583	0.4229	0.4442	0.4356	0.4375	0.4315	0.4432	0.4324	0.4397	0.4378	0.4387	0.4311	0.4425	0.4429	0.4560	0.4712	0.4687
	720	**0.4497**	**0.4673**	0.4757	0.4854	0.4649	0.4719	0.4670	0.4726	0.4769	0.4844	0.4521	0.4717	0.4606	0.4703	0.4718	0.4932	0.5224	0.5212	0.5158	0.5089
ETTh2	96	**0.2840**	**0.3432**	0.3068	0.3627	0.2856	0.3456	0.2912	0.3458	0.2864	0.3462	0.2886	0.3448	0.3086	0.3567	0.3025	0.3676	0.3038	0.3594	0.3154	0.3715
	192	**0.3530**	**0.3890**	0.3700	0.3979	0.3597	0.3977	0.3656	0.3950	0.3734	0.4001	0.3571	0.3922	0.3818	0.4051	0.3941	0.4261	0.3755	0.4049	0.3947	0.4234
	336	0.3715	**0.4099**	0.3804	0.4143	**0.3662**	0.4121	0.4032	0.4284	0.3867	0.4158	0.3797	0.4109	0.4115	0.4298	0.4900	0.4895	0.4182	0.4345	0.4166	0.4415
	720	0.4171	0.4461	0.4173	0.4473	0.4381	0.4654	0.4369	0.4567	0.4199	**0.4458**	**0.4134**	0.4493	0.4274	0.4535	0.8112	0.6381	0.4410	0.4628	0.4876	0.4855
ETTm1	96	**0.2902**	**0.3467**	0.3303	0.3685	0.2928	0.3496	0.2996	0.3547	0.3006	0.3503	0.3186	0.3714	0.3067	0.3545	0.3037	0.3471	0.3104	0.3639	0.3135	0.3625
	192	0.3330	0.3715	0.3581	0.3843	**0.3324**	0.3722	0.3395	0.3802	0.3370	0.3761	0.3482	0.3771	0.3463	0.3819	0.3363	**0.3670**	0.3481	0.3874	0.3640	0.3981
	336	**0.3616**	**0.3905**	0.3869	0.3993	0.3625	0.3919	0.3681	0.3932	0.3682	0.3946	0.3675	0.3992	0.3817	0.4046	0.3714	0.3909	0.3797	0.4045	0.3918	0.4135
	720	**0.4125**	**0.4192**	0.4407	0.4344	0.4160	0.4203	0.4312	0.4315	0.4262	0.4268	0.4353	0.4315	0.4390	0.4381	0.4236	0.4223	0.4375	0.4396	0.4388	0.4449
ETTm2	96	0.1671	0.2603	0.1809	0.2756	**0.1665**	**0.2563**	0.1728	0.2662	0.1721	0.2615	0.1675	0.2606	0.1757	0.2629	0.1676	0.2633	0.1801	0.2718	0.1768	0.2693
	192	**0.2230**	**0.2991**	0.2310	0.3069	0.2247	0.3040	0.2334	0.3074	0.2294	0.2993	0.2301	0.3003	0.2404	0.3064	0.2257	0.3056	0.2421	0.3132	0.2414	0.3107
	336	**0.2740**	**0.3330**	0.2795	0.3384	0.2868	0.3426	0.2862	0.3445	0.2829	0.3337	0.2752	0.3359	0.2992	0.3401	0.2987	0.3618	0.2920	0.3452	0.2928	0.3460
	720	0.3609	0.3877	0.3682	0.3882	0.3622	0.3916	0.3721	0.3969	0.3669	**0.3867**	**0.3594**	0.3881	0.3907	0.4025	0.4150	0.4380	0.3728	0.3944	0.4079	0.4152
Electricity	96	0.1358	0.2326	0.1463	0.2502	0.1378	0.2376	0.1373	0.2367	0.1373	0.2323	**0.1297**	**0.2251**	0.1367	0.2330	0.1406	0.2410	0.1435	0.2426	0.1369	0.2408
	192	0.1518	0.2480	0.1587	0.2639	0.1567	0.2557	0.1534	0.2506	0.1571	0.2528	**0.1506**	**0.2437**	0.1536	0.2495	0.1544	0.2542	0.1551	0.2511	0.1533	0.2548
	336	0.1686	**0.2638**	0.1737	0.2720	0.1710	0.2693	0.1686	0.2668	**0.1677**	0.2648	0.1689	0.2664	0.1698	0.2657	0.1693	0.2709	0.1723	0.2719	0.1691	0.2699
	720	0.2059	0.2955	0.2089	0.3017	0.2094	0.3013	0.2061	0.2979	**0.2036**	0.2946	0.2047	**0.2942**	0.2151	0.3026	0.2040	0.3037	0.2108	0.3014	0.2058	0.3001

Table 10: Few-shot forecasting results with 10% training data (MSE / MAE).

Dataset	H	CLAIR MSE	CLAIR MAE	Time-LLM MSE	Time-LLM MAE	S^{2}IP-LLM MSE	S^{2}IP-LLM MAE	OFA MSE	OFA MAE	CALF MSE	CALF MAE	FSCA MSE	FSCA MAE	SE-LLM MSE	SE-LLM MAE	DLinear MSE	DLinear MAE	iTransformer MSE	iTransformer MAE	PatchTST MSE	PatchTST MAE
Weather	96	0.1626	0.2176	**0.1623**	**0.2144**	0.1676	0.2226	0.1650	0.2155	0.1808	0.2319	0.1684	0.2227	0.1716	0.2189	0.1750	0.2335	0.1934	0.2446	0.1739	0.2223
	192	**0.2085**	**0.2577**	0.2256	0.2731	0.2107	0.2604	0.2155	0.2599	0.2304	0.2747	0.2134	0.2600	0.2142	0.2579	0.2147	0.2672	0.2489	0.2893	0.2327	0.2783
	336	**0.2566**	**0.2915**	0.2759	0.3118	0.2584	0.2917	0.2666	0.3000	0.2779	0.3126	0.2614	0.2981	0.2699	0.3031	0.2579	0.3036	0.2975	0.3288	0.2704	0.3066
	720	0.3212	**0.3394**	0.3325	0.3504	0.3247	0.3402	0.3248	0.3416	0.3401	0.3555	0.3267	0.3452	0.3489	0.3576	**0.3202**	0.3534	0.4175	0.4027	0.3346	0.3498
ETTh1	96	**0.4503**	**0.4407**	0.5204	0.4903	0.5936	0.5230	0.4582	0.4591	0.5457	0.5153	0.4669	0.4663	0.4831	0.4600	0.4508	0.4530	0.5952	0.5270	0.5631	0.5158
	192	0.5006	0.4727	0.5790	0.5156	0.6828	0.5388	0.5363	0.4890	0.5978	0.5425	0.5225	0.4865	0.5568	0.5020	**0.4808**	**0.4702**	0.6509	0.5607	0.7542	0.5990
	336	0.5337	0.4946	0.7554	0.5819	0.7217	0.5437	0.6292	0.5487	0.7365	0.5843	0.6684	0.5605	0.6145	0.5379	**0.4994**	**0.4875**	0.7080	0.5744	0.7644	0.6183
	720	0.7508	0.6174	0.8969	0.6461	0.9745	0.6981	0.7410	0.6083	0.9872	0.7020	0.8769	0.6424	0.9134	0.6707	**0.6924**	**0.5991**	0.8565	0.6526	0.8218	0.6532
ETTh2	96	**0.3033**	**0.3636**	0.3197	0.3791	0.3454	0.3934	0.3222	0.3788	0.3159	0.3771	0.3365	0.3773	0.3194	0.3770	0.3301	0.3834	0.3620	0.4029	0.3908	0.4323
	192	**0.3837**	0.4163	0.3955	0.4246	0.4149	0.4382	0.4235	0.4422	0.4218	0.4343	0.4212	0.4396	0.3888	**0.4133**	0.4205	0.4455	0.4167	0.4366	0.4677	0.4602
	336	**0.4102**	**0.4430**	0.4193	0.4468	0.4215	0.4476	0.4457	0.4599	0.4304	0.4484	0.4262	0.4482	0.4148	0.4438	0.4389	0.4590	0.4330	0.4536	0.4713	0.4851
	720	**0.4411**	**0.4676**	0.4821	0.4827	0.6771	0.5849	0.5883	0.5504	0.5148	0.5089	0.4609	0.4728	0.4572	0.4746	0.4651	0.4958	0.4929	0.4959	0.4924	0.4860
ETTm1	96	0.3881	0.3988	0.4331	0.4297	0.4165	0.4231	0.3940	0.4048	0.4109	0.4235	0.4393	0.4264	0.5337	0.4735	**0.3615**	**0.3955**	0.4635	0.4536	0.4423	0.4297
	192	0.4211	0.4196	0.4492	0.4386	0.4613	0.4495	0.4242	0.4200	0.4594	0.4463	0.4579	0.4294	0.6017	0.5145	**0.3862**	**0.4109**	0.4874	0.4716	0.5438	0.4773
	336	0.4490	0.4325	0.4762	0.4558	0.4719	0.4555	0.4591	0.4360	0.5345	0.4955	0.4607	0.4355	0.6444	0.5172	**0.4182**	**0.4298**	0.6144	0.5346	0.6570	0.5242
	720	0.5346	0.4830	0.5388	0.4894	0.5839	0.5117	0.5621	0.4954	0.5890	0.5230	0.6115	0.5248	0.7239	0.5557	**0.4864**	**0.4737**	0.7168	0.5804	0.7661	0.5819
ETTm2	96	**0.1704**	**0.2606**	0.2083	0.2978	0.2082	0.2881	0.1980	0.2780	0.1903	0.2750	0.1980	0.2764	0.2025	0.2820	0.2068	0.2928	0.2180	0.3011	0.2126	0.2942
	192	**0.2265**	**0.2997**	0.2609	0.3296	0.2770	0.3355	0.2537	0.3128	0.2550	0.3200	0.2526	0.3106	0.2477	0.3174	0.2649	0.3318	0.2745	0.3355	0.2813	0.3397
	336	**0.2827**	**0.3369**	0.3039	0.3542	0.3209	0.3656	0.2998	0.3432	0.3150	0.3599	0.3059	0.3559	0.2988	0.3490	0.3174	0.3656	0.3237	0.3676	0.3314	0.3679
	720	**0.3648**	**0.3896**	0.4338	0.4245	0.4247	0.4230	0.4081	0.4091	0.4173	0.4160	0.3883	0.3934	0.4076	0.4192	0.3969	0.4126	0.4424	0.4360	0.4942	0.4565
Electricity	96	0.1419	**0.2409**	0.1501	0.2522	0.1438	0.2444	**0.1418**	0.2414	0.1495	0.2523	0.1422	0.2413	0.1509	0.2471	0.1945	0.3060	0.1505	0.2537	0.1494	0.2532
	192	0.1611	0.2593	0.1645	0.2675	0.1585	0.2568	**0.1580**	0.2554	0.1710	0.2729	0.1587	**0.2553**	0.1680	0.2629	0.2058	0.3159	0.1730	0.2740	0.1655	0.2680
	336	0.1821	0.2803	0.1861	0.2875	**0.1773**	0.2745	0.1774	**0.2733**	0.1944	0.2946	0.1826	0.2765	0.1923	0.2894	0.2211	0.3306	0.2043	0.3029	0.1842	0.2867
	720	0.2426	0.3307	0.2432	0.3310	**0.2355**	0.3216	**0.2355**	**0.3192**	0.2577	0.3437	0.2446	0.3316	0.2454	0.3323	0.2590	0.3625	0.2506	0.3347	0.2400	0.3284

C Few-shot Forecasting Protocol and Full Results

C.1 Protocol

In the few-shot setting, we evaluate robustness under data scarcity using the first 10% or 5% of the training split. For baselines with official few-shot scripts, we follow the released configurations. Otherwise, we retain the corresponding full-shot configuration, except for the reduced training data and an early-stopping patience of 10 epochs. CLAIR follows the same training-data restriction. Under the 5% setting, results for H=720 on ETTh1 and ETTh2 are omitted due to insufficient training samples for the input–output window, consistent with prior protocols such as OFA (Zhou et al., 2023).

C.2 Full Results

Tables 10 and 11 report horizon-wise few-shot forecasting results corresponding to the averaged scores in the main paper.

D Full Short-term Forecasting Results

Table 12 reports short-term forecasting results on the M4 benchmark corresponding to the averaged scores in the main paper.

Table 11: Few-shot forecasting results with 5% training data (MSE / MAE).

Dataset	H	CLAIR MSE	CLAIR MAE	Time-LLM MSE	Time-LLM MAE	S^{2}IP-LLM MSE	S^{2}IP-LLM MAE	OFA MSE	OFA MAE	CALF MSE	CALF MAE	FSCA MSE	FSCA MAE	SE-LLM MSE	SE-LLM MAE	DLinear MSE	DLinear MAE	iTransformer MSE	iTransformer MAE	PatchTST MSE	PatchTST MAE
Weather	96	0.1798	0.2381	**0.1775**	**0.2310**	0.1884	0.2423	0.1855	0.2349	0.1870	0.2412	0.1841	0.2338	0.1917	0.2439	0.1922	0.2598	0.2088	0.2617	0.1854	0.2390
	192	**0.2290**	**0.2796**	0.2617	0.2958	0.2438	0.2969	0.2377	0.2870	0.2401	0.2867	0.2322	0.2824	0.2389	0.2843	0.2413	0.3019	0.2569	0.3015	0.2492	0.2917
	336	**0.2896**	0.3284	0.2970	**0.3268**	0.3289	0.3531	0.2921	0.3309	0.2949	0.3288	0.2913	0.3302	0.3043	0.3296	0.2928	0.3418	0.3138	0.3480	0.3108	0.3437
	720	**0.3582**	**0.3723**	0.3748	0.3850	0.4061	0.4053	0.3728	0.3827	0.3814	0.3883	0.3605	0.3756	0.3629	0.3770	0.3632	0.3945	0.3875	0.3978	0.3928	0.3975
ETTh1	96	0.6033	0.5157	0.7049	0.5613	0.8031	0.5903	0.6220	0.5297	0.6659	0.5519	0.6313	0.5496	0.6403	0.5528	**0.5565**	**0.5057**	0.7338	0.5789	0.6462	0.5551
	192	0.6689	**0.5432**	0.8215	0.5879	0.8289	0.5942	0.8224	0.6094	0.9616	0.6758	0.7869	0.6020	0.8295	0.6274	**0.6305**	0.5550	0.8229	0.6188	0.9925	0.7022
	336	0.7436	0.5939	0.9112	0.6264	1.3839	0.8311	1.0763	0.7180	0.9742	0.6850	0.8076	0.6144	0.8558	0.6289	**0.6937**	**0.5828**	0.8844	0.6493	1.0182	0.7142
	720	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
ETTh2	96	**0.3390**	0.4061	0.4671	0.4686	0.4973	0.4827	0.4204	0.4665	0.3474	0.4019	0.3437	0.4092	0.3556	0.4058	0.4128	0.4588	0.3941	0.4262	0.3404	**0.3917**
	192	**0.3768**	**0.4236**	0.5475	0.5082	0.5978	0.5380	0.4533	0.4532	0.4845	0.4843	0.3806	0.4256	0.4796	0.4935	0.4611	0.4943	0.4413	0.4564	0.4118	0.4262
	336	0.4029	0.4408	0.6252	0.5532	0.7550	0.6154	0.5354	0.5335	0.4914	0.4912	**0.4018**	**0.4397**	0.5448	0.5428	0.5878	0.5739	0.4586	0.4722	0.4633	0.4643
	720	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
ETTm1	96	0.3884	**0.4006**	0.3942	0.4110	0.4083	0.4225	0.3932	0.4093	0.3986	0.4149	0.4515	0.4315	0.5534	0.4776	**0.3783**	0.4068	0.4227	0.4334	0.4318	0.4325
	192	0.4209	**0.4232**	0.4427	0.4397	0.4387	0.4382	0.4382	0.4341	0.4377	0.4405	0.4346	0.4286	0.6009	0.4983	**0.4103**	0.4272	0.4906	0.4698	0.5300	0.4912
	336	0.4701	0.4526	0.5577	0.5001	0.5167	0.4887	0.5048	0.4677	0.4960	0.4750	0.4849	0.4528	0.6024	0.5013	**0.4455**	**0.4444**	0.5697	0.5104	0.5532	0.4977
	720	0.5621	**0.5042**	0.6363	0.5406	0.7168	0.5945	0.6395	0.5389	0.7646	0.6094	0.6168	0.5389	0.6676	0.5490	**0.5514**	0.5068	0.7707	0.6052	0.7661	0.5975
ETTm2	96	0.2024	0.2847	0.2185	0.2962	0.2205	0.2941	**0.2006**	**0.2816**	0.2167	0.2963	0.2123	0.2899	0.2143	0.2965	0.2204	0.3031	0.2126	0.2987	0.2353	0.3089
	192	**0.2523**	**0.3142**	0.2814	0.3423	0.2785	0.3354	0.2600	0.3201	0.2865	0.3388	0.2576	0.3155	0.2754	0.3282	0.2789	0.3440	0.2673	0.3340	0.3113	0.3611
	336	**0.3024**	**0.3481**	0.3225	0.3606	0.3277	0.3723	0.3264	0.3607	0.3462	0.3832	0.3263	0.3630	0.3109	0.3580	0.3263	0.3751	0.3189	0.3665	0.3655	0.3866
	720	**0.4159**	**0.4151**	0.4687	0.4479	0.4527	0.4476	0.4824	0.4442	0.4294	0.4209	0.4294	0.4206	0.4324	0.4227	0.4795	0.4824	0.4352	0.4301	0.5010	0.4581
Electricity	96	**0.1443**	**0.2439**	0.1515	0.2549	0.1455	0.2452	0.1506	0.2538	0.1657	0.2701	0.1469	0.2461	0.1592	0.2557	0.2242	0.3338	0.1578	0.2616	0.1554	0.2602
	192	0.1616	0.2603	0.1708	0.2732	**0.1610**	**0.2596**	0.1655	0.2662	0.1926	0.2946	0.1624	0.2611	0.1757	0.2718	0.2403	0.3471	0.1843	0.2880	0.1707	0.2731
	336	0.1831	0.2799	0.1917	0.2912	**0.1818**	**0.2797**	0.1865	0.2858	0.2272	0.3248	0.1829	0.2818	0.1989	0.2963	0.2643	0.3666	0.2104	0.3129	0.1906	0.2914
	720	**0.2333**	**0.3185**	0.2418	0.3359	0.2442	0.3359	0.2376	0.3257	0.3268	0.4030	0.2363	0.3257	0.2528	0.3406	0.4542	0.5266	0.2882	0.3733	0.2446	0.3356

Table 12: Short-term forecasting results on the M4 dataset (SMAPE / MASE / OWA).

		CLAIR	Time-LLM	S^{2}IP-LLM	OFA	CALF	FSCA	SE-LLM	DLinear	iTransformer	PatchTST
Yearly	SMAPE	13.764	13.721	13.899	14.368	13.488	**13.413**	13.675	14.358	14.282	13.664
	MASE	3.161	3.036	3.093	3.397	3.043	**3.009**	3.140	3.128	3.202	3.086
	OWA	0.819	0.802	0.814	0.867	0.796	**0.789**	0.813	0.833	0.840	0.806
Quarterly	SMAPE	**10.116**	10.861	10.545	10.444	10.697	10.213	10.191	10.500	10.740	10.916
	MASE	**1.178**	1.293	1.239	1.237	1.233	1.227	1.193	1.239	1.280	1.289
	OWA	**0.889**	0.965	0.931	0.925	0.935	0.911	0.898	0.929	0.954	0.966
Monthly	SMAPE	**12.858**	13.445	13.122	12.871	13.068	12.963	13.121	13.394	15.062	14.180
	MASE	**0.953**	1.027	0.987	**0.953**	0.968	0.987	0.983	1.007	1.170	1.127
	OWA	**0.894**	0.949	0.919	**0.894**	0.908	0.913	0.917	0.938	1.072	1.021
Others	SMAPE	**4.848**	6.122	5.763	5.321	5.011	4.893	5.124	5.111	5.342	6.089
	MASE	**3.225**	4.060	4.023	3.566	3.418	3.534	3.508	3.665	3.613	3.941
	OWA	**1.019**	1.284	1.241	1.122	1.066	1.072	1.092	1.116	1.132	1.262

E Sensitivity Analysis

We analyze the sensitivity of CLAIR on ETTh1 with input length L=512 across prediction horizons $H \in \{96, 192, 336, 720\}$. We evaluate model sensitivity across three categories of hyperparameters: architectural parameters, including the soft-prompt length and the number of structural-anchor scales; training parameters, including the EMA momentum coefficient μ and the auxiliary loss weights λ_{PTA} and λ_{FC}; and optimal-transport parameters, including the Sinkhorn entropic regularization $\varepsilon_{\mathrm{OT}}$ and the number of Sinkhorn iterations.

E.1 Number of Soft Prompt Tokens

Varying n_p shows that n_p=4 consistently provides a strong trade-off. Fewer prompts limit task-level conditioning capacity, whereas substantially larger values introduce redundant prompt tokens and can degrade accuracy. We therefore adopt n_p=4 as the default.

E.2 Number of Scales in the Structural Anchor

Starting from the ordered candidate set $\{12, 24, 48, 96\}$, we construct the structural anchor using one to four temporal scales. Three scales yield the best overall performance: performance degrades with fewer scales, while adding a fourth scale provides little or no additional gain. Accordingly, we use K=3 by default, with dataset-specific window sets as described in Appendix A.6.

E.3 Sinkhorn Regularization and Iterations

An intermediate $\varepsilon_{\mathrm{OT}}$=0.1 balances alignment sharpness and numerical stability. Increasing Sinkhorn iterations beyond a moderate range (around 20) yields diminishing returns. Overall, CLAIR is not overly sensitive to narrow hyperparameter choices around the selected defaults.

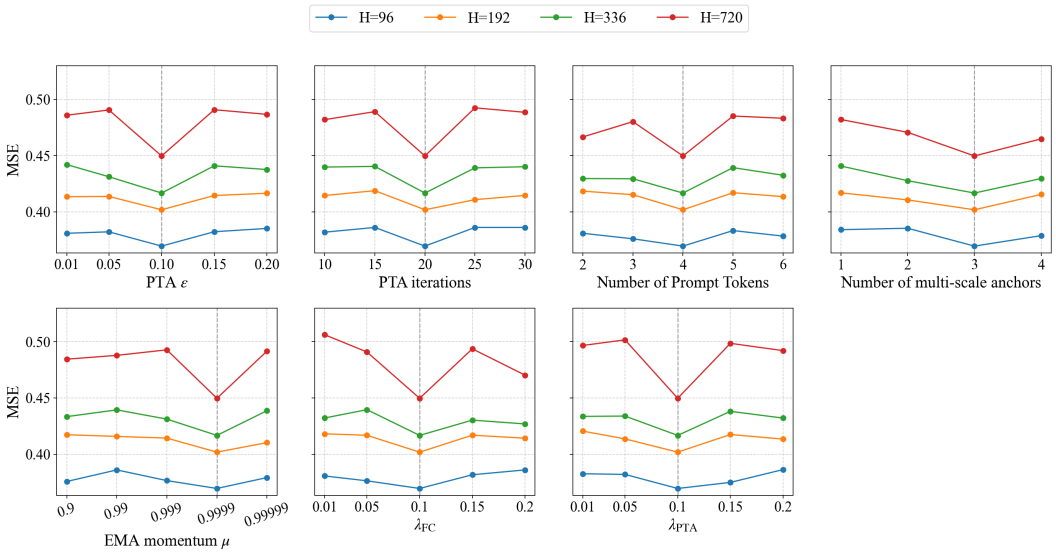


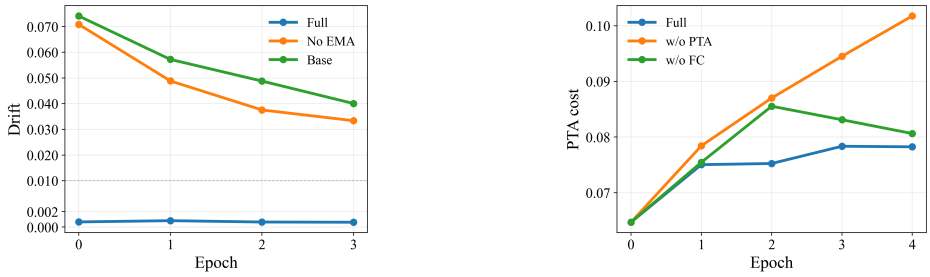
Figure 5: Sensitivity of ETTh1 MSE to seven hyperparameters. Top row, from left to right: PTA ϵ, PTA iterations, number of prompt tokens, and number of scales in the structural anchor. Bottom row, from left to right: EMA momentum μ, λ_{FC}, and λ_{PTA}. Vertical dashed lines mark the selected values.

E.4 EMA Momentum Coefficient μ.

We evaluate the effect of the EMA momentum coefficient μ on ETTh1 while keeping all other hyperparameters fixed. $\mu = 0.9999$ consistently achieves the lowest MSE across all prediction horizons. Both smaller momentum values ($\mu \leq 0.999$) and the excessively large value $\mu = 0.99999$ result in degraded performance. This suggests that a moderate EMA update rate provides a stable yet sufficiently responsive teacher representation. These results support our choice of $\mu = 0.9999$ as the default setting.

E.5 Auxiliary Loss Weights λ_{PTA} and λ_{FC}

Both λ_{FC} and λ_{PTA} show a consistent trend across prediction horizons on ETTh1, with 0.1 providing the best overall performance for both weights. Smaller or larger values generally lead to higher errors, indicating the importance of appropriately balancing the auxiliary objectives with the primary forecasting loss. We therefore use $\lambda_{\mathrm{FC}} = \lambda_{\mathrm{PTA}} = 0.1$ on ETTh1, while the dataset-specific settings for the other benchmarks are provided in Appendix A.5.



(a) Reference drift

(b) Student–reference PTA cost

Figure 6: Reference stability and patch-level alignment during training. (a) Reference drift on a fixed validation batch. (b) Student–reference PTA cost. Lower values indicate a more stable reference and better patch-level alignment, respectively.

F Stability and Alignment Analysis

Beyond forecasting accuracy, we analyze whether CLAIR stabilizes the adaptation process from two complementary perspectives: latent-space stability and output-level robustness. We first examine whether the EMA teacher provides a temporally stable reference and whether PTA keeps student representations aligned with this reference. We then evaluate prediction sensitivity under input perturbations.

Reference Stability and Patch-level Alignment. We first examine whether the proposed teacher-guided regularization provides a stable reference and maintains student–teacher representation alignment during training. All diagnostics are computed on a fixed validation batch using clean inputs in evaluation mode.

To measure reference stability, we compute the epoch-to-epoch drift of normalized patch-level reference hidden states:

$$\Delta_{\mathrm{ref}}(e) = \frac{1}{UN} \sum_{u=1}^{U} \sum_{i=1}^{N} \left\| \tilde{h}_{\mathrm{ref},i}^{(u,e)} - \tilde{h}_{\mathrm{ref},i}^{(u,e-1)} \right\|_2, \tag{18}$$

where $U = BM$ and N denotes the number of patch tokens. For the full model, the reference is given by the EMA teacher. For non-EMA variants, we use the stop-gradient student representation as the reference. As shown in Figure 6(a), the full model exhibits substantially lower reference drift than the No EMA and Base variant. This supports the role of the EMA teacher as a temporally stable alignment reference during adaptation.

We further measure whether student patch representations remain aligned with the EMA teacher by computing the student–reference PTA cost:

$$\mathcal{C}_{\mathrm{PTA}}(e) = \frac{1}{U} \sum_{u=1}^{U} \left\langle \Pi^{\star(u,e)}, C^{(u,e)} \right\rangle, \tag{19}$$

where $C^{(u,e)}$ is the student–teacher patch cost matrix and $\Pi^{\star(u,e)}$ is the corresponding Sinkhorn transport plan. Figure 6(b) shows that the full model maintains a lower PTA cost than the w/o PTA variant. Without PTA, the student–reference transport cost increases as training proceeds, suggesting that student patch representations drift farther from the EMA teacher reference. The w/o FC variant also shows a higher PTA cost than the full model, suggesting that FC may indirectly support representation-level stability. Together, these results show that the EMA teacher provides a stable reference, while PTA helps maintain patch-level alignment with that reference.

Noise Sensitivity of Predictions. We analyze output-level stability under input perturbations. Given Gaussian noise $\epsilon_\sigma \sim \mathcal{N}(0, \sigma^2)$, we construct a noisy input

$$X_\sigma = X + \epsilon_\sigma, \tag{20}$$

and measure prediction drift as

$$D_{\mathrm{drift}}(\sigma) = \frac{1}{BHM} \left\| \hat{Y}(X_\sigma) - \hat{Y}(X) \right\|_2^2. \tag{21}$$

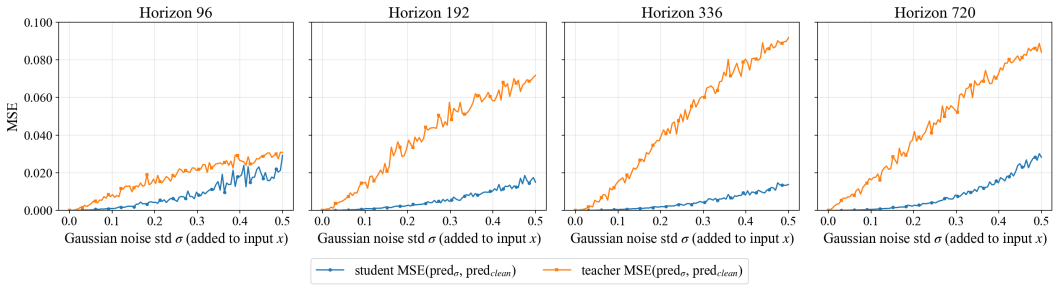


Figure 7: Noise sensitivity of student and EMA teacher predictions. Prediction drift MSE is measured between clean-input and noisy-input predictions across forecasting horizons. Lower values indicate stronger output stability under input perturbations.

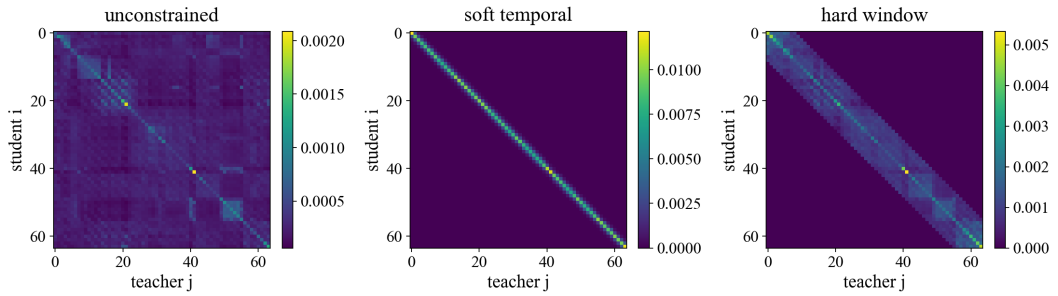


Figure 8: Mean transport plans on ETTh2 at forecast horizon $H = 96$ under three alignment settings: unconstrained (left), soft temporal (middle), and hard-window (right).

This metric measures sensitivity to input noise rather than forecasting error against the ground-truth target. As shown in Figure 7, the student predictions exhibit lower drift than the EMA teacher predictions across prediction horizons. This suggests that although the EMA teacher serves as a stable clean-view reference during training, the student learns a more perturbation-tolerant forecasting function. The gap becomes larger as the noise level increases, indicating that the asymmetric teacher–student design helps improve output stability under input perturbations.

G Additional Diagnostics for Unconstrained PTA

G.1 Post-hoc Temporally Constrained OT

We further examine how explicit temporal constraints alter the transport geometry induced by the learned patch representations. Using fixed ETTh2 full-model checkpoints, we recompute the Sinkhorn plans without retraining the forecasting model. We compare unconstrained PTA with a soft temporal penalty,

$$C_{ij}^{\text{soft}} = C_{ij} + \lambda |i - j|^2, \tag{22}$$

and a hard local window that permits transport only when $|i - j| \leq 8$. Figure 9 shows that unconstrained PTA retains substantially larger expected patch-index displacement than either temporally constrained alternative across forecasting horizons. The corresponding plans in Figure 8 show that the soft penalty concentrates transport near matching

Table 13: ETTh2: post-hoc constrained OT locality at H=96 (other horizons are similar).

OT variant	$\mathbb{E}[\lvert i-j\rvert]$	Diag. mass
Unconstrained	19.19	0.048
Soft $+ 0.05\,\lvert i-j\rvert^2$	0.52	0.562
Hard $\lvert i-j\rvert \leq 8$	3.48	0.147

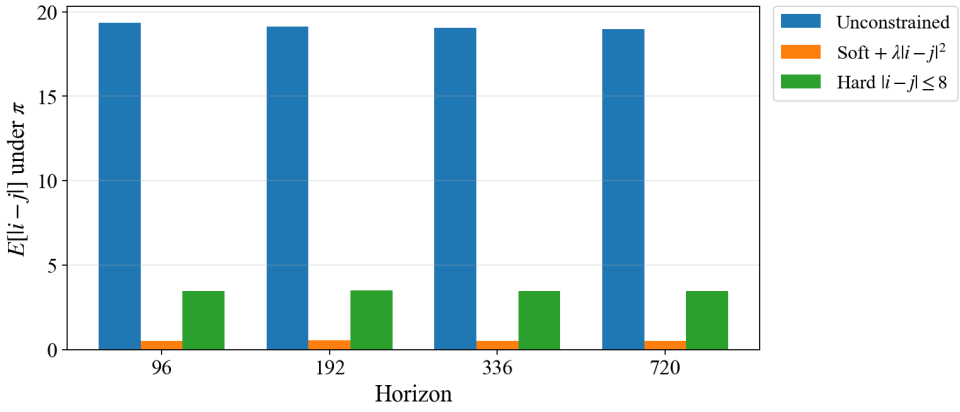


Figure 9: ETTh2: Post-hoc expected transport lag $\mathbb{E}_\pi[|i-j|]$ for unconstrained PTA, soft temporal penalty ($C_{ij}^{\text{soft}} = C_{ij} + \lambda|i-j|^2$), and hard local window ($|i-j|\leq 8$).

Table 14: Trainable parameter comparison averaged over prediction horizons $H \in \{96, 192, 336, 720\}$.

Method	ETTh1/ETTh2	ETTm1/ETTm2	ECL	Weather
Time-LLM	55.41M	55.50M	55.41M	55.41M
S^2IP-LLM	57.04M	58.50M	57.04M	57.04M
OFA	17.33M	9.34M	17.33M	17.33M
CALF	18.53M	18.53M	18.53M	18.53M
FSCA	31.75M	35.60M	31.32M	30.33M
SE-LLM	97.33M	97.33M	100.31M	98.61M
CLAIR	19.18M	19.24M	19.18M	19.18M

indices, whereas the hard window produces band-limited correspondence while retaining flexibility within the permitted range. Table 13 provides the corresponding locality statistics at $H = 96$. Together, these post-hoc results place unconstrained PTA between uniform matching and explicitly localized transport. It exhibits a modest temporal preference but retains considerably more non-local correspondence than the constrained alternatives. Because the transport plans are recomputed from fixed representations without retraining, this analysis characterizes correspondence geometry rather than forecasting performance. The soft-penalty setting is therefore treated as a locality stress test rather than as a tuned replacement for unconstrained PTA.

H Additional Diagnostic Analyses

This section provides extended diagnostic analyses and profiling to substantiate the teacher-guided design. We analyze the parameter and inference efficiency of CLAIR. The EMA teacher receives no gradient updates and is discarded at inference. Thus, trainable and inference-time parameter counts include only the student branch. CLAIR uses substantially fewer trainable parameters than Time-LLM and S^2IP-LLM and remains comparable to CALF, although OFA is lighter under this setting. The EMA teacher and Sinkhorn transport introduce training-time overhead, requiring $1.44\times$ training time per iteration and $1.24\times$ peak GPU memory compared with the student-only base. However, these components are removed at inference, where CLAIR uses only the student branch and achieves $1.00\times$ inference latency relative to the student-only base.

H.1 Profiling Setup

Teacher-guided training, including the EMA teacher, Sinkhorn-based PTA, and forecast consistency, incurs additional computational cost only during training, with no overhead at inference time. At deployment CLAIR evaluates the student branch alone: input perturbation, teacher forwards, Sinkhorn transport, and the consistency loss are disabled, and teacher weights can be deleted from memory. We profile this claim on the ETTm1 Full checkpoint without retraining (L=512, M=7, GPT-2 truncated to 6 layers, patch size P=16, and stride S=8; multi-scale patches of sizes $24/48/96$ with strides $12/24/48$; λ_{PTA}=0.10 and λ_{FC}=0.05). Unless noted, training steps use batch size 128 and inference follows the test loader (batch size 1). Timings are median CUDA-event measurements after warmup on an NVIDIA L40S.

H.2 Trainable vs. stored parameters

Table 15: Training cost on ETTm1. Params are student-trainable counts averaged over H. Relative columns are vs. the student-only Base step.

Setting	Params	ms/iter	Peak GPU	Rel. time / mem
CLAIR Full	19.24M	1107.7	35.73 GB	$1.44\times$ / $1.24\times$
Student Base	19.24M	771.4	28.92 GB	$1.00\times$ / $1.00\times$

We distinguish gradient-updated student parameters from the total model weights stored during training and inference. As shown in Table 14, CLAIR uses 19.18–19.24M trainable parameters across the evaluated dataset configurations, averaged over $H \in \{96, 192, 336, 720\}$. For ETTm1, the horizon-averaged count is 19.24M, whereas the smaller forecasting head at H=96 reduces the number of trainable parameters to 7.44M. Together with 81.11M frozen GPT-2 weights, the student stores 88.55M parameters at this horizon. The EMA teacher has the same parameter count but receives no gradient updates. Its weights corresponding to the trainable student parameters are updated by EMA. Figure 10 makes the distinction between training-time storage and deployed model size explicit. At H=96, training retains separate student and teacher copies, corresponding to approximately 177.1M stored parameters in total. After discarding the teacher, inference requires only the 88.55M student model. Thus, the teacher increases weight storage only during training and does not enlarge the deployed model. In terms of trainable parameters, CLAIR is comparable to CALF and remains smaller than FSCA, Time-LLM, S^{2}IP-LLM, and SE-LLM, although OFA uses fewer trainable parameters.

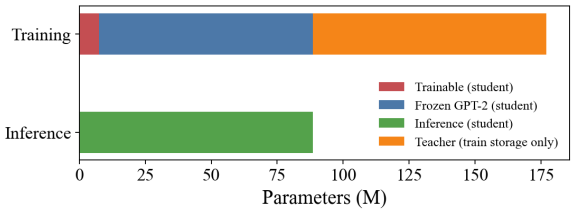


Figure 10: Parameter composition at H=96 on ETTm1. Training retains trainable student parameters (red), frozen GPT-2 weights (blue), and a separate EMA teacher copy (orange). At inference, the teacher is discarded and only the student is retained (green).

H.3 Student-only inference

Returned forecasts have always been produced by the student. The profiled change is that evaluation no longer executes the teacher or Sinkhorn. With teacher weights discarded, ETTm1 inference at H=96 takes 4.39 ms/sample and 0.46 GB peak memory (Table 16), matching the student-only Base architecture (**1.00**× latency). If the teacher forward is left in the eval graph, latency rises to 10.32 ms and peak memory to 0.76 GB, about 2.35× slower, which illustrates why the skip must be in the forward path rather than only in the reported prediction. Student-forward FLOPs at batch 1 are 42.97 G. Latency stays in a narrow band across $H \in \{96, 192, 336, 720\}$ (4.39, 3.57, 3.64, and 3.84 ms; Figure 11), because the GPT-2 trunk dominates and only the linear forecast head depends on H.

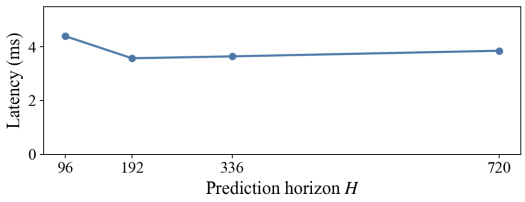


Figure 11: Student-only inference latency versus forecast horizon on ETTm1 (batch 1). Variation is small relative to the backbone cost.

H.4 Training-time overhead

We compare a Full training step (student forward, no-grad teacher forward, Sinkhorn PTA, FC loss, backward, EMA update) to a student-only Base step (student forward and backward only). Table 15 reports 1107.7 ms/iter and 35.73 GB peak memory for Full versus 771.4 ms/iter and 28.92 GB for Base, i.e. **1.44**× wall-clock and **1.24**× peak GPU memory. These ratios closely match the ETTh1 and Weather microbenchmarks (1.44× time; 1.23–1.24× memory), indicating that the teacher-guided overhead is structural rather than dataset-specific. Figure 12 decomposes the same steps

Table 16: Inference cost on ETTm1. Student-only deploy discards teacher weights. "Eval w/ teacher" is the previous eval path that still ran the teacher and Sinkhorn.

Setting	Latency	Peak GPU	Rel. latency
Student-only	4.39 ms	0.46 GB	1.00×
Eval w/ teacher	10.32 ms	0.76 GB	2.35×

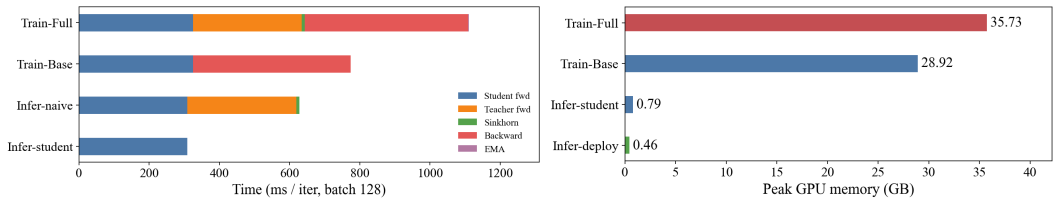


Figure 12: Left: Per-iteration runtime breakdown at batch size 128 on ETTm1. Train-Full includes the teacher forward pass, Sinkhorn computation, and EMA update, while the teacher forward pass is also retained in the teacher-based evaluation path. Infer-student performs only a single student forward pass. Right: Peak GPU memory usage across training and inference configurations. Infer-deploy removes the teacher modules after training.

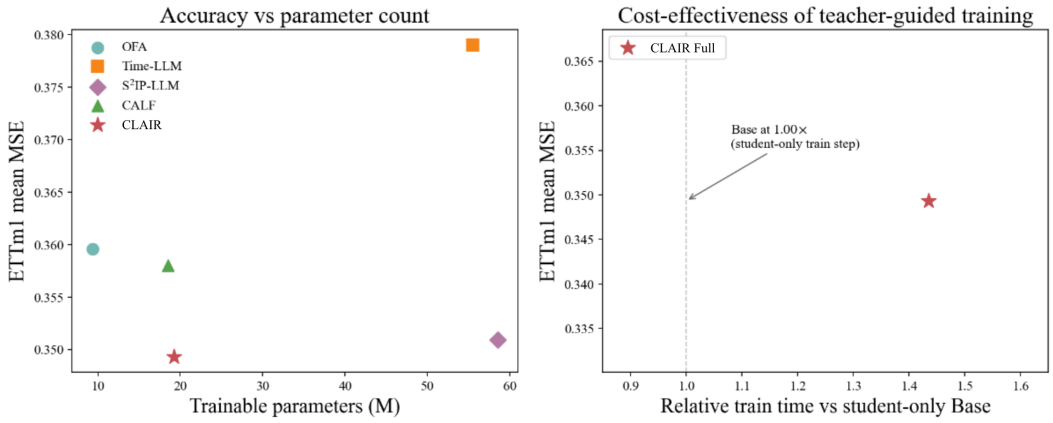


Figure 13: Left: ETTm1 mean MSE versus trainable parameters for LLM-TS baselines and CLAIR. Right: Measured Full versus Base training-time ratio; inference uses the student only in both cases.

at batch 128. The extra Full cost is almost entirely a second (teacher) forward ($\approx$ 308 ms) plus additional backward work on the student graph. Sinkhorn PTA remains cheap: $\approx$ 8.7 ms per iteration ($< 1\%$ of a Full step). The additional computation introduced by PTA is modest. Constructing the pairwise student–teacher cost matrix and performing 20 log-domain Sinkhorn iterations requires an analytically estimated 8.76 GFLOPs at the training token width. For reference, a single forward pass through the student model requires 42.35 GFLOPs at batch size 1. The EMA teacher update adds little further overhead, taking only approximately 2.4 ms.

H.5 Cost-effectiveness versus LLM baselines

Figure 13 (left) plots ETTm1 mean MSE (full-shot average over four horizons) against trainable parameter count. CLAIR attains 0.3493 MSE at 19.24M trainable parameters, improving on Time-LLM (0.3790 / 55.50M), S^{2}IP-LLM (0.3509 / 58.50M), OFA (0.3596 / 9.34M), and CALF (0.3580 / 18.53M). CLAIR achieves lower horizon-averaged MSE than these four baselines, with a trainable parameter count close to CALF, below Time-LLM and S2IP-LLM, and above OFA. These counts describe trainable parameters. Total stored weights and measured training costs are reported separately in Sections H.2 and H.4. The right panel of Figure 13 places CLAIR's Full training at $1.44\times$ the step time of the student-only Base. This additional cost is incurred only during optimization, since both configurations use the student branch alone at inference.