C4TS: CO-ENCODING AND CO-DECODING VIA TEMPORAL–VISUAL CROSS-MODAL COLLABORATION FOR TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Visual representations of time series have been widely explored to improve forecasting. In temporal–visual feature-fusion approaches, interaction between parallel temporal and visual encoders feeds a single temporal decoding pathway, yet does not inherently reconcile the structural differences between temporal token sequences and spatially organized visual features. In temporal–visual prediction-fusion approaches, parallel decoding preserves both forecasting pathways, but globally shared weights cannot adapt their combination to individual input instances and future positions. We propose **C**o-Encoding and **C**o-Decoding via Temporal–Visual **C**ross-Modal **C**ollaboration for **T**ime **S**eries Forecasting (C4TS), a framework that coordinates two forecasting pathways through both collaborative encoding (co-encoding) and collaborative decoding (co-decoding). For co-encoding, Spatial-Aware Visual Adaptation bridges the structural gap between temporal and visual representations by adapting spatially organized visual features into a cross-attention memory, enabling visual features to enhance temporal representations. For co-decoding, Instance $\times$ Future-Patch Adaptive Fusion assigns distinct fusion weights to each input instance and future patch, adaptively combining temporal predictions with visual reconstruction forecasts. These weights are generated by learnable future-patch queries that retrieve historical evidence separately from the temporal and visual representations. Experiments on public time series benchmarks demonstrate that C4TS achieves state-of-the-art overall forecasting performance.

1 INTRODUCTION

Time series forecasting supports applications ranging from energy management to transportation. Temporal models typically encode historical numerical observations and map their representations to future values through a prediction head. Despite their architectural differences, these methods perform both encoding and prediction within the numerical time-series modality (Zeng et al., 2023; Nie et al., 2023; Liu et al., 2024a). Recently, visual reconstruction provides an alternative forecasting mechanism: historical observations are transformed into images, a visual model reconstructs future regions, and an inverse transformation recovers numerical predictions. VisionTS (Chen et al., 2025) demonstrates the effectiveness of pretrained visual masked autoencoders (MAEs) (He et al., 2022) in this setting. These approaches motivate collaboration between temporal and visual pathways in both representing historical observations and generating forecasts.

Existing methods realize different forms of temporal–visual collaboration. Time-VLM (Zhong et al., 2025) integrates visual and textual features with temporal representations before a single temporal prediction head, as illustrated in Figure 1(a). This interaction incorporates visual information but does not preserve visual reconstruction as a separate forecasting mechanism. DMMV (Shen et al., 2025) retains both prediction pathways in a design similar to Figure 1(b), assigning their roles through decomposition or a visual-backcast residual formulation. Its globally shared learnable scalar fusion weight does not vary across samples or future positions. These designs raise a question: how can the pathways interact during encoding while retaining their forecasting mechanisms and adapting their combination to each input and future position?

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.

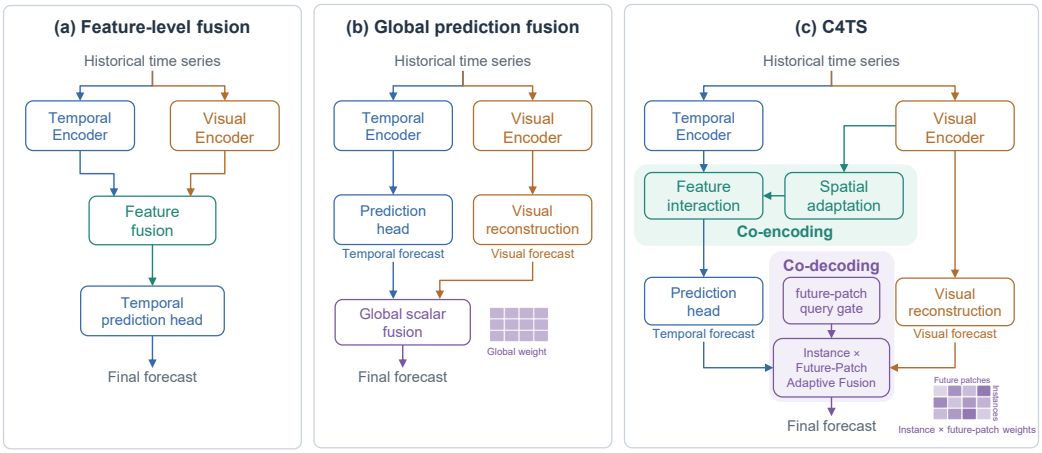


Figure 1: Representative temporal–visual forecasting designs. (a) Feature interaction between temporal and visual encoders supports a single temporal decoding pathway. (b) Parallel encoders lack direct feature interaction, while two decoding pathways produce forecasts combined using a globally shared scalar weight. (c) C4TS couples spatially adapted feature interaction for co-encoding with Instance $\times$ Future-Patch Adaptive Fusion for co-decoding. Green and purple shaded regions in (c) highlight the co-encoding and co-decoding components, respectively. Darker cells in the illustrative weight matrices indicate greater visual weight.

The first challenge concerns collaborative encoding, or *co-encoding*. Temporal representations are organized as sequences of tokens, whereas visual representations preserve a two-dimensional spatial layout. This structural mismatch makes direct interaction between the two representations nontrivial. To exploit visual information for temporal modeling, we introduce a spatial adaptation interface that extracts useful information from spatially organized visual features and constructs a memory to enhance temporal representations. Our ablation study supports the effectiveness of spatial adaptation (Section 4.3), while attention visualizations illustrate how it redistributes attention over visual history (Section 4.4).

The second challenge concerns collaborative decoding, or *co-decoding*. Global fusion weights cannot express instance-dependent or future-position-dependent preferences. Instance dependence allows the model to adjust the relative contributions of the two pathways to the characteristics of each historical input window. Future-position dependence allows their contributions to vary within the forecast horizon, which we implement at the future-patch level. Our Gate analysis reveals variation in visual weights both across input instances at the same future position and across future patches within an instance, supporting instance-dependent and position-dependent forecast combination (Section 4.4). These considerations motivate jointly conditioning fusion weights on input instances and future patches, supported by our ablation study (Section 4.3). Addressing this challenge requires a gating mechanism that uses historical representations from both pathways to generate fusion weights jointly conditioned on the input instance and future patch.

We propose **C4TS**, a framework for co-encoding and co-decoding via temporal–visual cross-modal collaboration (Figure 1(c)). Spatial-Aware Visual Adaptation enables visual-to-temporal feature interaction during encoding, while Instance $\times$ Future-Patch Adaptive Fusion combines the two forecasts during decoding, with each instance corresponding to a multivariate historical input window and each future patch to a segment of the forecast horizon. This collaboration retains numerical prediction and visual reconstruction as distinct forecasting mechanisms rather than reducing their outputs to a single prediction head.

Experiments on public benchmarks demonstrate that C4TS achieves state-of-the-art overall forecasting performance. On the evaluated ablation benchmarks, the fused forecast outperforms either constituent pathway prediction in aggregate. Ablation studies support the effectiveness of joint instance–future-patch weighting and spatially adapted feature interaction. Our contributions are:

- **C4TS Framework.** C4TS coordinates two complete-history forecasting pathways through co-encoding and co-decoding, preserving their respective prediction mechanisms while enabling collaboration at both levels.
- **Spatial-Aware Visual Adaptation.** An adapter exploits the two-dimensional neighborhoods of visible historical MAE tokens to construct a memory for temporal cross-attention, connecting visual reconstruction features to temporal modeling.
- **Instance $\times$ Future-Patch Adaptive Fusion.** Learnable future-patch queries retrieve evidence from both historical token memories to determine forecast combination weights jointly for each instance and future patch.

2 Related Work

2.1 Time Series Forecasting and Language Model Adaptation

Numerical time series forecasting models encode historical observations and map the resulting representations to future values. Transformer-based approaches explore efficient attention and frequency-domain modeling (Zhou et al., 2021; 2022). Autoformer (Wu et al., 2021) combines progressive decomposition with an auto-correlation mechanism, PatchTST (Nie et al., 2023) uses channel-independent patch tokens, and iTransformer (Liu et al., 2024a) applies attention across variate tokens to capture multivariate dependencies. DLinear (Zeng et al., 2023) demonstrates the competitiveness of decomposition-based linear forecasting. Other approaches explore wavelet-based patch mixing in WPMixer (Murad et al., 2025) and periodic and higher-frequency patterns in FreqCycle (Zhang et al., 2026). Recent temporal models also explore amplitude–phase reconstruction in APRNet (Liu et al., 2026a) and local cross-variate modeling over variate–patch representations in VPNet (Liu et al., 2026b). Two-dimensional temporal representations support intra-period and inter-period modeling in TimesNet (Wu et al., 2023). These methods develop representations within the numerical time-series modality, whereas C4TS explores collaboration between temporal modeling and visual reconstruction.

Language model adaptation spans parameter-efficient transfer and autoregressive reuse (Zhou et al., 2023; Liu et al., 2024b), LLM reprogramming in Time-LLM (Jin et al., 2024), and temporal–textual alignment (Liu et al., 2025b;a). However, Tan et al. (2024) find that removing or replacing LLM components in the evaluated methods often preserves or improves accuracy at lower computational cost. These findings motivate examining which pretrained capabilities transfer effectively to forecasting. Pretrained visual models offer another direction, which we discuss in Section 2.3.

2.2 Multi-View Modeling and Forecast Combination within the Temporal Modality

Within the numerical time-series modality, different scales, subtasks, and dependency structures provide complementary views for modeling and forecast combination. For example, TimeMixer (Wang et al., 2024) mixes multiscale representations and predictions, while TimeMixer++ (Wang et al., 2025) extracts patterns across scales and resolutions using two-dimensional temporal representations. Subtask-based approaches such as MAFS (Huang et al., 2025) coordinate specialized agents through information exchange and output voting. These methods exploit complementary representations or predictors within a single modality, whereas C4TS coordinates temporal modeling and visual reconstruction.

Online forecast ensembling adapts predictor combinations as new observations arrive. For example, OneNet (Zhang et al., 2023) combines predictors emphasizing temporal and cross-variate dependencies to address concept drift. These predictors capture different dependencies within the numerical time-series modality. C4TS instead infers weights within each forecast window without online parameter updates.

2.3 Vision-Based and Multimodal Time Series Forecasting

Beyond pretrained language models, visual representations offer another route to time series analysis (Ni et al., 2025). Image-based forecasting can either map visual features directly to numerical

predictions, as in ForCNN (Semenoglou et al., 2023), or generate future content in image space, as in VisualAE and ViTime (Sood et al., 2021; Yang et al., 2025). Within the latter category, VisionTS (Chen et al., 2025) transfers a pretrained MAE to reconstruct future regions of period-structured images and maps them back to numerical forecasts. C4TS retains this reconstruction mechanism alongside a temporal predictor; its Spatial-Aware Visual Adaptation connects encoded visual history to temporal cross-attention.

Feature-level multimodal fusion enriches temporal representations before prediction. For example, Time-VLM (Zhong et al., 2025) integrates temporal, visual, and textual representations. Such multimodal encoding supports a single temporal prediction head. C4TS also performs feature interaction but preserves visual reconstruction as a separate forecasting mechanism.

Prediction-level temporal–visual fusion retains distinct forecasting mechanisms and combines their outputs. A representative is DMMV (Shen et al., 2025), whose pathways are assigned seasonal–trend components (DMMV-S) or roles defined by a visual-backcast residual formulation (DMMV-A). Its globally shared learnable scalar cannot vary across samples or future positions. C4TS supplies both pathways with complete histories and couples feature interaction with Instance $\times$ Future-Patch Adaptive Fusion.

3 METHOD

3.1 PROBLEM FORMULATION

Given historical observations $\mathbf{X} \in \mathbb{R}^{L\times C}$, we learn f_θ to predict $\hat{\mathbf{Y}} = f_\theta(\mathbf{X}) \in \mathbb{R}^{H\times C}$, where L, H, and C denote history length, forecasting horizon, and variable count. The target is $\mathbf{Y} \in \mathbb{R}^{H\times C}$, and θ denotes the model parameters. The batch dimension is omitted throughout. We distinguish temporal input patches of length P_t, used to tokenize historical observations, from future patches of length P_f, which define the temporal granularity of forecast fusion. Each pathway normalizes its input using historical statistics and inverse-transforms its forecast to the target scale before fusion.

3.2 C4TS FRAMEWORK

C4TS operates on numerical and visual views of the same complete history (Figure 2(a)). Collaborative encoding (co-encoding) adapts visual features for temporal interaction; collaborative decoding (co-decoding) combines the outputs of a temporal prediction head and a visual reconstruction pathway. Both produce forecasts for all H steps and C variables.

The temporal and visual encoders produce historical representations $\mathbf{T} = E_\mathrm{t}(\mathcal{P}_\mathrm{t}(\mathbf{X}))$ and $\mathbf{V} = E_\mathrm{v}(\mathcal{I}(\mathbf{X}))$, respectively, where $\mathcal{P}_\mathrm{t}$ constructs temporal input patches and $\mathcal{I}$ constructs the masked visual input. The representations $\mathbf{T}$ and $\mathbf{V}$ contain temporal and visible visual tokens. Spatial adaptation $\mathcal{A}$ and feature interaction $\mathcal{F}$ enable co-encoding, yielding the temporal forecast:

$$\hat{\mathbf{Y}}_\mathrm{T} = \mathcal{H}_\mathrm{t}(\mathcal{F}(\mathbf{T}, \mathcal{A}(\mathbf{V}))). \tag{1}$$

Co-decoding adaptively combines the two pathway forecasts:

$$\hat{\mathbf{Y}} = (\mathbf{1} - \boldsymbol{\alpha}) \odot \hat{\mathbf{Y}}_\mathrm{T} + \boldsymbol{\alpha} \odot \hat{\mathbf{Y}}_\mathrm{V}, \tag{2}$$

where $\hat{\mathbf{Y}}_\mathrm{V} = \mathcal{R}_\mathrm{v}(\mathbf{V})$ is the visual reconstruction forecast. Both the temporal prediction mapping $\mathcal{H}_\mathrm{t}$ and the visual reconstruction mapping $\mathcal{R}_\mathrm{v}$ include inverse normalization. The symbols $\odot$ and $\mathbf{1}$ denote element-wise multiplication and an all-ones tensor. The tensor $\boldsymbol{\alpha}$ contains visual fusion weights expanded to the forecast shape, as specified in Section 3.4. The weights $\boldsymbol{\alpha}$ depend on the original historical representations $\mathbf{T}$ and $\mathbf{V}$ and vary by instance and future patch.

Temporal pathway. Inspired by the patch tokenization strategy of PatchTST (Nie et al., 2023), we divide each variable's history into overlapping patches of length P_t after applying reversible instance normalization (RevIN) (Kim et al., 2022). Linear embedding, learnable positional encoding, and a Transformer encoder with parameters shared across variables produce temporal tokens $\mathbf{T}_c \in \mathbb{R}^{N_\mathrm{t}\times d}$, where $c \in \{1, \ldots, C\}$ indexes variables, N_t is the temporal token count, and d is the temporal hidden dimension. After visual interaction, enhanced tokens $\mathbf{T}_c^+$ are flattened and mapped to H future values by a shared linear prediction head. Inverse normalization then yields the c-th variable's forecast in $\hat{\mathbf{Y}}_\mathrm{T}$. The original temporal tokens $\mathbf{T}_c$ also serve as inputs to the Gate.

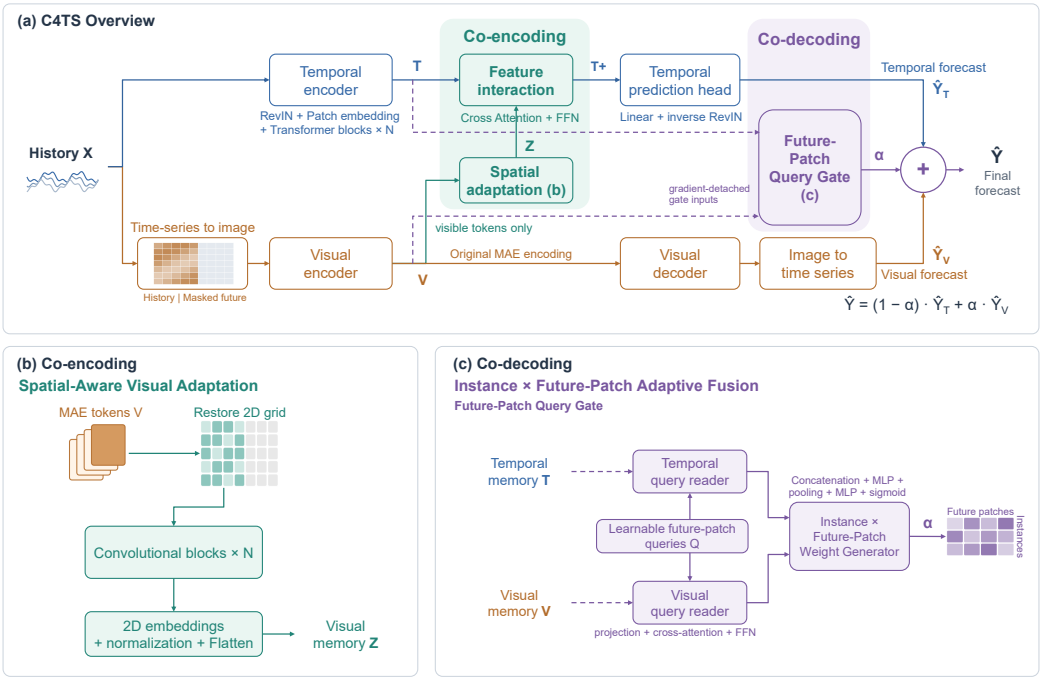


Figure 2: C4TS: (a) dual-path forecasting with feature interaction and prediction fusion; dashed purple arrows indicate stop-gradient Gate inputs; (b) spatial adaptation of visible historical MAE tokens for co-encoding; (c) shared future-patch queries and separate temporal/visual readers for instance-dependent future-patch weight generation in co-decoding. Green and purple shaded regions in (a) highlight the co-encoding and co-decoding components, respectively.

Visual pathway. Following VisionTS (Chen et al., 2025), we transform each variable's normalized history into a period-structured image, with the region corresponding to the forecast horizon masked. A pretrained MAE (He et al., 2022) encodes the visible historical patches into tokens $\mathbf{V}_c \in \mathbb{R}^{N_\mathrm{v} \times d_\mathrm{v}}$ and reconstructs the masked future region, where N_v and d_v denote the visual token count and hidden dimension. The reconstructed image is mapped back to a numerical sequence and inverse-normalized to obtain the c-th variable's forecast in $\hat{\mathbf{Y}}_\mathrm{V}$. The historical patch tokens $\mathbf{V}_c$ also support spatial adaptation and the Gate.

3.3 Spatial-Aware Visual Adaptation

Visual features preserve a two-dimensional spatial layout, whereas temporal tokens form a one-dimensional sequence. The adapter (Figure 2(b)) bridges this structural difference by extracting spatial information from visual features and organizing it into a token-sequence memory for temporal cross-attention. For variable c, patch-position metadata restores $\mathbf{V}_c$ to a visible historical grid of R rows and S columns, with $N_\mathrm{v} = RS$. Future cells are excluded. A pointwise projection precedes two spatial residual blocks:

$$\mathbf{G}_c^{(0)} = \mathrm{Conv}_{1\times1}^{\mathrm{in}}(\mathrm{Grid}(\mathbf{V}_c)), \tag{3}$$

$$\mathbf{G}_c^{(r)} = \mathbf{G}_c^{(r-1)} + \mathcal{B}_r(\mathbf{G}_c^{(r-1)}), \quad r \in \{1, 2\}. \tag{4}$$

Here, Grid restores spatial token order, and $\mathrm{Conv}_{1\times1}^{\mathrm{in}}$ projects input channels to d. The grid $\mathbf{G}_c^{(r)} \in \mathbb{R}^{d\times R\times S}$ is the output after block r, with $r = 0$ denoting the projected input. Each $\mathcal{B}_r$ applies group normalization, a 3×3 depthwise convolution with dilation r, GELU, and pointwise convolution, preserving grid resolution. The two blocks capture immediate and extended neighborhoods. Output projection and learnable row/column embeddings yield

$$\widetilde{\mathbf{Z}}_c = \mathrm{Flatten}_{\mathrm{sp}}\left[\mathrm{Conv}_{1\times1}^{\mathrm{out}}(\mathbf{G}_c^{(2)}) + \mathbf{E}_{\mathrm{row}} + \mathbf{E}_{\mathrm{col}}\right] \in \mathbb{R}^{N_\mathrm{v}\times d}. \tag{5}$$

Here, $\mathrm{Conv}_{1\times1}^{\mathrm{out}}$ is the output pointwise projection, $\mathbf{E}_{\mathrm{row}}$ and $\mathbf{E}_{\mathrm{col}}$ are row and column embeddings broadcast along their respective axes, and $\mathrm{Flatten}_{\mathrm{sp}}$ converts the spatial grid into a token sequence. Adding a learnable visual-view embedding $\mathbf{e}_{\mathrm{view}} \in \mathbb{R}^d$, shared across variables and broadcast to all N_{v} historical tokens, produces the memory:

$$\mathbf{Z}_c = \widetilde{\mathbf{Z}}_c + \mathbf{e}_{\mathrm{view}}. \tag{6}$$

Temporal tokens query this memory through $\mathbf{T}_c^+ = \mathcal{F}(\mathbf{T}_c, \mathbf{Z}_c)$:

$$\mathbf{U}_c = \mathbf{T}_c + \mathrm{MHA}(\mathbf{T}_c, \mathbf{Z}_c, \mathbf{Z}_c), \tag{7}$$

$$\mathbf{T}_c^+ = \mathbf{U}_c + \mathrm{FFN}(\mathbf{U}_c). \tag{8}$$

Here, MHA denotes multi-head attention, with arguments ordered as queries, keys, and values; $\mathbf{U}_c$ is the intermediate temporal representation. The FFN denotes a feed-forward network. This interaction requires no one-to-one patch alignment. The visual decoder retains the original MAE encoding; adaptation affects only the feature-interaction path.

3.4 Instance × Future-Patch Adaptive Fusion

Instance × Future-Patch Adaptive Fusion combines the two pathway forecasts with weights conditioned jointly on a forecasting instance and a future patch. An instance is one multivariate historical input window, and a future patch is a segment of its forecast horizon. Here, "×" denotes the Cartesian product of the instance and future-patch index sets, with one fusion weight assigned to each instance–future-patch pair. Co-decoding divides the forecast window into $K = \lceil H/P_{\mathrm{f}} \rceil$ patches, allowing a shorter final patch. A Future-Patch Query Gate (Figure 2(c)) retrieves historical evidence to infer a visual weight for each instance and future patch. All quantities below are defined for a single input instance; fusion weights are computed from its historical representations and therefore vary across instances.

Historical evidence retrieval. The Gate receives pre-interaction temporal tokens $\mathbf{T}_c$ and original visible MAE tokens $\mathbf{V}_c$, rather than the adapted memory $\mathbf{Z}_c$. Separate projections map stop-gradient inputs to routing dimension d_{g}, yielding $\mathbf{M}_c^{\mathrm{t}} = \phi_{\mathrm{t}}(\mathrm{sg}(\mathbf{T}_c))$ and $\mathbf{M}_c^{\mathrm{v}} = \phi_{\mathrm{v}}(\mathrm{sg}(\mathbf{V}_c))$. Here, sg denotes stop-gradient, ϕ_{t} and ϕ_{v} each apply linear projection and GELU, and $\mathbf{M}_c^{\mathrm{t}}$ and $\mathbf{M}_c^{\mathrm{v}}$ are temporal and visual routing memories. Upright t and v label pathways. Learnable queries $\mathbf{Q} = [\mathbf{q}_1, \ldots, \mathbf{q}_K]^\top \in \mathbb{R}^{K\times d_{\mathrm{g}}}$ are shared across instances, variables, and pathways. Readers have separate parameters for $j \in \{\mathrm{t}, \mathrm{v}\}$ and compute

$$\mathbf{B}_c^j = \mathbf{Q} + \mathrm{MHA}^j(\mathbf{Q}, \mathbf{M}_c^j, \mathbf{M}_c^j), \tag{9}$$

$$\mathbf{A}_c^j = \mathbf{B}_c^j + \mathrm{FFN}^j(\mathbf{B}_c^j). \tag{10}$$

Here, $\mathbf{B}_c^j$ is the intermediate query representation. For $k \in \{1, \ldots, K\}$, $\mathbf{q}_k$ is the query for future patch k, and $\mathbf{A}_{c,k}^j$ is row k of the retrieved feature matrix $\mathbf{A}_c^j \in \mathbb{R}^{K\times d_{\mathrm{g}}}$: query identity supplies position information, while historical memory supplies instance dependence. The queries contain no future observations.

Weight generation. For each variable and future patch, an MLP ψ combines retrieved representations and their absolute difference. The difference term explicitly captures element-wise discrepancies between temporal and visual representations to inform fusion-weight generation:

$$\mathbf{f}_{c,k} = \psi([\mathbf{A}_{c,k}^{\mathrm{t}}; \mathbf{A}_{c,k}^{\mathrm{v}}; |\mathbf{A}_{c,k}^{\mathrm{t}} - \mathbf{A}_{c,k}^{\mathrm{v}}|]). \tag{11}$$

Here, $[\,;\,]$ and $|\cdot|$ denote concatenation and element-wise absolute value. We add learnable variable embeddings $\mathbf{e}_c$ and average over variables to obtain the patch-specific summary $\bar{\mathbf{f}}_k = \frac{1}{C}\sum_{c=1}^{C}(\mathbf{f}_{c,k} + \mathbf{e}_c)$. Averaging these summaries over future patches yields the window-level summary $\mathbf{g} = \frac{1}{K}\sum_{k=1}^{K}\bar{\mathbf{f}}_k$. The visual fusion weight combines these two levels of information:

$$\alpha_k = \mathrm{sigmoid}(\rho([\bar{\mathbf{f}}_k; \mathbf{g}])). \tag{12}$$

Both ψ and ρ are multilayer perceptrons (MLPs). The final linear layer of ρ has zero-initialized weights and bias, so the sigmoid initially assigns equal visual and temporal weights, $\alpha_k = 0.5$. Combining patch-specific evidence with the window summary produces instance-dependent future-patch weights from history rather than predicted values.

Prediction combination. For future step $t \in \{1, \ldots, H\}$, define its patch index $k(t) = 1 + \lfloor (t-1)/P_{\mathrm{f}} \rfloor$. The final forecast is

$$\hat{Y}_{t,c} = (1 - \alpha_{k(t)})\hat{Y}_{\mathrm{T},t,c} + \alpha_{k(t)}\hat{Y}_{\mathrm{V},t,c}. \tag{13}$$

Here, $\hat{Y}_{\mathrm{T},t,c}$ and $\hat{Y}_{\mathrm{V},t,c}$ are temporal and visual predictions for step t and variable c of the input instance. The scalar α_k is the visual weight for future patch k of the input instance, with $1 - \alpha_k$ weighting the temporal forecast. The tensor $\boldsymbol{\alpha}$ in Section 3.2 is formed by assigning $\alpha_{k(t)}$ to each future step t and broadcasting across variables, yielding an instance-dependent, piecewise-constant combination. Both pathways produce complete forecasts and remain active during inference.

3.5 Training Objective and Optimization

We supervise both pathways and their fused forecast using mean squared error (MSE) over future numerical values on the restored target scale:

$$\mathcal{L} = \lambda_{\mathrm{t}}\mathcal{L}_{\mathrm{T}} + \lambda_{\mathrm{v}}\mathcal{L}_{\mathrm{V}} + \lambda_{\mathrm{f}}\mathcal{L}_{\mathrm{F}}, \tag{14}$$

$$\mathcal{L}_j = \mathrm{MSE}(\hat{\mathbf{Y}}_j, \mathbf{Y}) = \frac{1}{HC}\sum_{t=1}^{H}\sum_{c=1}^{C}\left(\hat{Y}_{j,t,c} - Y_{t,c}\right)^2, \tag{15}$$

where $j \in \{\mathrm{T}, \mathrm{V}, \mathrm{F}\}$ labels temporal, visual, and fused forecasts, respectively, and $\hat{\mathbf{Y}}_{\mathrm{F}} = \hat{\mathbf{Y}}$. Here, $\hat{Y}_{j,t,c}$ and $Y_{t,c}$ are the predicted and target values at future step t for variable c. The loss is defined per input instance and averaged over training instances. The coefficients λ_{t}, λ_{v}, and λ_{f} weight the three losses. All newly introduced components are trained jointly; within the pretrained MAE, only encoder and decoder layer-normalization parameters are updated. Stop-gradient blocks the Gate's input-gradient paths to the encoders, while the fused loss still trains both pathways through their predictions. Temporal supervision also trains visual-to-temporal interaction through the adapter and the trainable visual encoder parameters.

4 Experiments

4.1 Experimental Setup

Datasets and evaluation. We evaluate C4TS on seven multivariate forecasting benchmarks: the four ETT datasets (ETTh1, ETTh2, ETTm1, and ETTm2) (Zhou et al., 2021), Weather, Electricity (ECL), and Traffic. Dataset sources and descriptions are provided in Appendix A. We use the versions of all seven datasets provided by the Time-Series Library (TSLib) and follow its chronological data splits and preprocessing protocol.[1] The ETT datasets use 12, 4, and 4 months for training, validation, and testing, respectively; the remaining datasets use a 7:1:2 split. All methods use a history length of 512 and forecast horizons of $H \in \{96, 192, 336, 720\}$. We report mean squared error (MSE) and mean absolute error (MAE), with lower values indicating better performance.

Baselines and result sources. We compare C4TS with nine baselines spanning temporal, language-model-based, and visual forecasting. We use the original papers' results with a history length of 512 for PatchTST (Nie et al., 2023), WPMixer (Murad et al., 2025), APRNet (Liu et al., 2026a), and Time-VLM (Zhong et al., 2025). For Time-LLM (Jin et al., 2024), we use the 512-step reproductions reported by Tan et al. (2024). The original experimental settings of DMMV (Shen et al., 2025), VisionTS (Chen et al., 2025), VPNet (Liu et al., 2026b), and FreqCycle (Zhang et al., 2026) use history lengths other than 512, including 96, 336, 1152, and other lengths. To ensure a fair comparison under a common history length, we reproduce these methods using their open-source code, changing only the input length to 512 and keeping all other hyperparameters unchanged.

Implementation. We implement C4TS in PyTorch and conduct experiments on a workstation equipped with two NVIDIA RTX PRO 4500 32GB GPUs. The visual pathway is initialized from pretrained MAE weights, with only its LayerNorm parameters fine-tuned. The temporal pathway, spatial adaptation, and fusion modules are trained jointly using the objective in Section 3.5. We train a separate model for each forecast horizon and select the checkpoint with the lowest validation MSE of the fused forecast.

[1] https://github.com/thuml/Time-Series-Library

Table 1: Forecasting results averaged over four horizons $(96, 192, 336, 720)$, with a history length of 512. Lower MSE and MAE are better. Red bold denotes the best and blue underline the second best; tied first places suppress second-place highlighting. Overall avg averages seven datasets. Average rank pools MSE and MAE ranks over only the seven dataset Avg rows (14 comparisons), excluding Overall avg; lower ranks are better. Ties use competition ranks at three-decimal precision. Full results appear in Table 3.

	Temporal–Visual					Visual		Temporal												
Dataset	**C4TS** (Ours)		Time-VLM (2025)		DMMV (2025)		VisionTS (2025)		APRNet (2026)		VPNet (2026)		FreqCycle (2026)		WPMixer (2025)		Time-LLM (2024)		PatchTST (2023)	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	**0.396**	0.416	0.405	0.420	0.441	0.447	0.398	**0.415**	0.411	0.427	0.429	0.435	0.414	0.435	0.422	0.423	0.417	0.432	0.413	0.434
ETTh2	0.344	0.386	0.341	0.391	0.574	0.523	0.344	0.383	0.336	0.384	0.359	0.391	0.357	0.401	0.355	0.387	0.360	0.396	**0.331**	**0.381**
ETTm1	**0.336**	**0.372**	0.350	0.377	0.364	0.384	0.344	**0.372**	0.348	0.378	0.376	0.383	0.356	0.383	0.376	0.388	0.356	0.377	0.353	0.382
ETTm2	0.252	0.315	**0.248**	0.311	0.429	0.431	0.270	0.328	0.252	0.314	0.256	**0.308**	0.256	0.318	0.271	0.317	0.260	0.315	0.256	0.317
Weather	**0.219**	0.258	0.224	0.263	0.228	0.292	0.229	0.262	0.223	0.262	0.234	**0.256**	0.228	0.267	0.243	0.269	0.243	0.270	0.226	0.264
Electricity	**0.155**	**0.249**	0.172	0.273	0.179	0.281	0.157	**0.249**	**0.155**	0.251	0.159	0.251	0.161	0.255	0.177	0.267	0.164	0.259	0.159	0.253
Traffic	0.386	0.257	0.419	0.303	0.492	0.338	0.388	0.255	0.388	0.276	**0.379**	0.263	0.398	**0.243**	0.489	0.297	0.421	0.281	0.391	0.264
Overall avg	**0.298**	**0.322**	0.308	0.334	0.387	0.385	0.304	0.323	0.302	0.327	0.313	0.327	0.310	0.329	0.333	0.335	0.317	0.333	0.304	0.328
Average rank	**2.071**		4.714		9.429		3.429		3.214		5.214		6.143		7.714		6.929		4.571	

4.2 Main Forecasting Results

Table 1 reports forecasting results averaged over four horizons. C4TS achieves the best horizon-averaged MSE on a majority of the datasets and the best average rank across MSE and MAE (2.071). Table 3 in Appendix B further reports the full results over 28 dataset–horizon settings. Across 70 MSE/MAE comparisons covering these settings and the seven dataset Avg rows, C4TS achieves the best average rank (2.243), the highest first-place count (24), and the highest top-two count (47), demonstrating strong overall forecasting performance.

4.3 Ablation Studies

We conduct ablations on the four ETT datasets with horizons $H \in \{96, 192, 336, 720\}$. Table 2 reports overall MSE averaged across all 16 settings. Overall ΔMSE measures the relative increase in mean MSE over C4TS; positive values indicate degradation. Appendix C provides configuration details and full results.

Co-encoding. Spatially adapted feature interaction improves overall accuracy, whereas directly injecting flattened and linearly projected MAE tokens performs worse than omitting feature interaction. Adding temporal-to-visual cross-attention also increases overall error, supporting the visual-to-temporal direction of interaction in our design.

Pathway predictions and co-decoding. The fused forecast outperforms either pathway output alone in aggregate. Uniform averaging and global scalar fusion perform similarly, while instance-wise and future-patch-wise weighting each improve accuracy. Joint instance–future-patch weighting performs best overall, supporting adaptation to both the input instance and forecast position.

4.4 Analysis of Temporal–Visual Collaboration

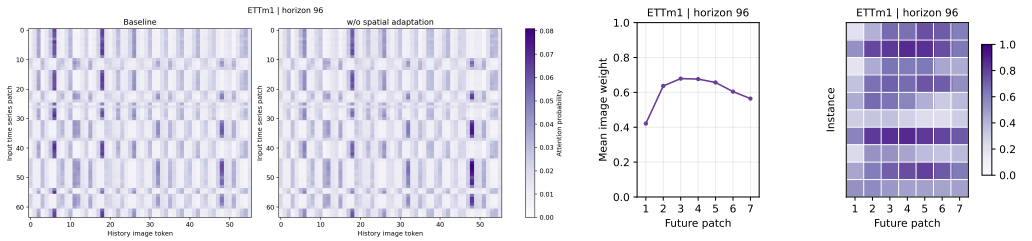
Co-encoding: Spatially adapted feature interaction. We visualize cross-attention weights for C4TS and the *w/o spatial adaptation* variant from Section 4.3 on ETTm1 (Figure 3(a)). Spatial adaptation changes how temporal queries attend to visual history, strengthening attention to some historical token positions while reducing it at others. Together with the ablations, this redistribution supports adapting spatial visual features before temporal interaction.

Co-decoding: Instance $\times$ Future-Patch Adaptive Fusion. Figure 3(b,c) visualizes fusion weights on ETTm1 for $H = 96$, with the remaining three horizons shown in Figure 4 of Appendix D. Along the instance dimension, the heatmaps show different visual weights at the same future patch, as well as distinct weight profiles across input instances. Along the future-patch di-

Table 2: Overall ablation results on the four ETT datasets. MSE averages all 16 dataset–horizon settings (lower is better). ΔMSE is the percentage change in overall MSE relative to C4TS; positive values indicate degradation. Full results are provided in Appendix C.

Variant	MSE	ΔMSE (%)
C4TS (full)	**0.3321**	–
Co-encoding ablations		
w/o spatial adaptation	0.3341	+0.603%
w/o spatial adaptation & feature interaction	0.3331	+0.302%
+ temporal-to-visual cross-attention‡	0.3412	+2.747%
Pathway predictions		
Temporal-path prediction	0.3535	+6.454%
Visual-path prediction	0.3481	+4.805%
Co-decoding ablations		
Uniform averaging†	0.3346	+0.748%
Global scalar fusion	0.3344	+0.678%
Instance-wise fusion	0.3334	+0.393%
Future-patch-wise fusion	0.3333	+0.356%

‡ Adds temporal-to-visual cross-attention to C4TS while retaining visual-to-temporal interaction.
† Uniform averaging uses equal prediction weights and omits the fused forecasting loss. Pathway predictions are outputs of the full model, not separately trained single-path models.



(a) Cross-attention comparison (b) Mean visual weights (c) Instance visual weights

Figure 3: Temporal–visual collaboration on ETTm1 ($H = 96$). (a) Cross-attention for the first instance: C4TS (left) and without spatial adaptation (right). (b) Visual fusion weights averaged over all test instances. (c) Weights for ten regularly spaced test instances; darker cells indicate larger visual weights. Visualization details and fusion weights for the remaining horizons are provided in Appendix D.

mension, instance-averaged visual weights show an early rise followed by horizon-dependent fluctuations. Weight variation across both instances and future patches further supports the use of fusion weights at the instance × future-patch granularity.

5 Conclusion

We presented C4TS, a temporal–visual forecasting framework that combines collaborative encoding and decoding while preserving two distinct prediction mechanisms. Spatial-Aware Visual Adaptation connects spatial visual features to temporal modeling, while Instance × Future-Patch Adaptive Fusion combines temporal predictions and visual reconstruction forecasts using instance-dependent, future-patch-dependent weights. Experiments demonstrate state-of-the-art overall forecasting performance. Ablations support spatially adapted visual-to-temporal interaction and joint instance–future-patch weighting, with the fused forecast outperforming either constituent pathway prediction in aggregate. Visualizations illustrate attention redistribution after spatial adaptation and fusion-weight variation across instances and future patches. These findings support temporal–visual collaboration through spatially adapted feature interaction and adaptive prediction fusion. Future work will investigate how these learned combinations generalize under distribution shifts and across different visual backbones.

AI USE STATEMENT

We used generative AI tools to aid or polish writing. Specifically, these tools were used for translation and for local editing to improve the clarity, grammar, and readability of author-written text, rather than to generate complete sections of the paper. We also used generative AI tools to format bibliographic references and complete or standardize citation metadata for literature already identified by the authors, rather than for literature retrieval, discovery, or review. All AI-assisted outputs were reviewed and verified by the authors, who take full responsibility for the final content of the paper.

REPRODUCIBILITY STATEMENT

Once the discussion forum opens, we will provide the reviewers and area chairs with access to an anonymized implementation through a private OpenReview comment. Upon acceptance, we will publicly release the source code, training configurations, and experiment scripts. All experiments use publicly available datasets in the versions provided by the Time-Series Library (TSLib), following its preprocessing protocol and chronological splits as described in Section 4.1 and Appendix A. Section 3 presents the model architecture and training objective, while Section 4.1 specifies the experimental settings and baseline result sources.

REFERENCES

Mouxiang Chen, Lefei Shen, Zhuo Li, Xiaoyun Joy Wang, Jianling Sun, and Chenghao Liu. VisionTS: Visual masked autoencoders are free-lunch zero-shot time series forecasters. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 8979–9007. PMLR, 2025. URL https://proceedings.mlr.press/v267/chen25be.html.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/He_Masked_Autoencoders_Are_Scalable_Vision_Learners_CVPR_2022_paper.html.

Qihe Huang, Zhengyang Zhou, Yangze Li, Kuo Yang, Binwu Wang, and Yang Wang. Many minds, one goal: Time series forecasting via sub-task specialization and inter-agent cooperation. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-5550. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/f34f0630c33be15b8c89426bb8056798-Abstract-Conference.html.

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*, 2024. URL https://openreview.net/forum?id=Unb5CVPtae.

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=cGDAkQo1C0p.

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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 18780–18788, 2025a. doi: 10.1609/aaai.v39i18.34067. URL https://ojs.aaai.org/index.php/AAAI/article/view/34067.

Hao Liu, Xiaobin Zhu, Chun Yang, Lirui Deng, and Xucheng Yin. Amplitude–phase reconstruction for non-stationary time-series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 38(8):4868–4882, 2026a. doi: 10.1109/TKDE.2026.3701273. URL https://doi.org/10.1109/TKDE.2026.3701273.

Kun Liu, Renjun Jia, Ruifeng Yang, Xirui Zeng, Yuqi Liang, and Cen Chen. Are global dependencies necessary? scalable time series forecasting via local cross-variate modeling. In *International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=CNVL194fO5.

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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 18915–18923, 2025b. doi: 10.1609/aaai.v39i18.34082. URL https://ojs.aaai.org/index.php/AAAI/article/view/34082.

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*, 2024a. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/2ea18fdc667e0ef2ad82b2b4d65147ad-Abstract-Conference.html.

Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. AutoTimes: Autoregressive time series forecasters via large language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024b. URL https://proceedings.nips.cc/paper_files/paper/2024/hash/dcf88cbc8d01ce7309b83d0ebaeb9d29-Abstract-Conference.html.

Md Mahmuddun Nabi Murad, Mehmet Aktukmak, and Yasin Yilmaz. WPMixer: Efficient multi-resolution mixing for long-term time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 19581–19588, 2025. doi: 10.1609/aaai.v39i18.34156. URL https://ojs.aaai.org/index.php/AAAI/article/view/34156.

Jingchao Ni, Ziming Zhao, ChengAo Shen, Hanghang Tong, Dongjin Song, Wei Cheng, Dongsheng Luo, and Haifeng Chen. Harnessing vision models for time series analysis: A survey. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 10612–10620, 2025. doi: 10.24963/ijcai.2025/1178. URL https://www.ijcai.org/proceedings/2025/1178. Survey Track.

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*, 2023. URL https://openreview.net/forum?id=Jbdc0vTOcol.

Artemios-Anargyros Semenoglou, Evangelos Spiliotis, and Vassilios Assimakopoulos. Image-based time series forecasting: A deep convolutional neural network approach. *Neural Networks*, 157: 39–53, 2023. doi: 10.1016/j.neunet.2022.10.006. URL https://www.sciencedirect.com/science/article/pii/S0893608022003902.

ChengAo Shen, Wenchao Yu, Ziming Zhao, Dongjin Song, Wei Cheng, Haifeng Chen, and Jingchao Ni. Multi-modal view enhanced large vision models for long-term time series forecasting. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-3119. URL https://proceedings.nips.cc/paper_files/paper/2025/hash/860c1c657deafe09f64c013c2888bd7b-Abstract-Conference.html.

Srijan Sood, Zhen Zeng, Naftali Cohen, Tucker Balch, and Manuela Veloso. Visual time series forecasting: An image-driven approach. In *Proceedings of the 2nd ACM International Conference on AI in Finance*. ACM, 2021. doi: 10.1145/3490354.3494387. URL https://doi.org/10.1145/3490354.3494387.

Mingtian Tan, Mike A. Merrill, Vinayak Gupta, Tim Althoff, and Thomas Hartvigsen. Are language models actually useful for time series forecasting? In *Advances in Neural Information Processing Systems*, volume 37, pp. 60162–60191, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/6ed5bf446f59e2c6646d23058c86424b-Abstract-Conference.html.

Artur Trindade. ElectricityLoadDiagrams20112014. UCI Machine Learning Repository, 2015. URL https://doi.org/10.24432/C58C86. Dataset.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y. Zhang, and Jun Zhou. TimeMixer: Decomposable multiscale mixing for time series forecasting. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/a7ac8a21e5a27e7ab31a5f42a0117bdb-Abstract-Conference.html.

Shiyu Wang, Jiawei Li, Xiaoming Shi, Zhou Ye, Baichuan Mo, Wenze Lin, Shengtong Ju, Zhixuan Chu, and Ming Jin. TimeMixer++: A general time series pattern machine for universal predictive analysis. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/2b187165e28fdfdc0ffb34d1bfff2b0c-Abstract-Conference.html.

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*, volume 34, 2021. URL https://papers.nips.cc/paper/2021/hash/bcc0d400288793e8bdcd7c19a8ac0c2b-Abstract.html.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. TimesNet: Temporal 2D-variation modeling for general time series analysis. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=ju_Uqw384Oq.

Luoxiao Yang, Yun Wang, Xinqi Fan, Israel Cohen, Jingdong Chen, and Zijun Zhang. ViTime: Foundation model for time series forecasting powered by vision intelligence. *Transactions on Machine Learning Research*, 2025. URL https://openreview.net/forum?id=XInsJDBIkp.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 11121–11128, 2023. doi: 10.1609/aaai.v37i9.26317. URL https://ojs.aaai.org/index.php/AAAI/article/view/26317.

Boya Zhang, Shuaijie Yin, Huiwen Zhu, and Xing He. FreqCycle: A multi-scale time-frequency analysis method for time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 28159–28167, 2026. doi: 10.1609/aaai.v40i33.40042. URL https://ojs.aaai.org/index.php/AAAI/article/view/40042.

Yi-Fan Zhang, Qingsong Wen, Xue Wang, Weiqi Chen, Liang Sun, Zhang Zhang, Liang Wang, Rong Jin, and Tieniu Tan. OneNet: Enhancing time series forecasting models under concept drift by online ensembling. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/dd6a47bc0aad6f34aa5e77706d90cdc4-Abstract.html.

Siru Zhong, Weilin Ruan, Ming Jin, Huan Li, Qingsong Wen, and Yuxuan Liang. Time-VLM: Exploring multimodal vision-language models for augmented time series forecasting. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 78478–78497. PMLR, 2025. URL https://proceedings.mlr.press/v267/zhong25a.html.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11106–11115, 2021. doi: 10.1609/aaai.v35i12.17325. URL https://ojs.aaai.org/index.php/AAAI/article/view/17325.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 27268–27286. PMLR, 2022. URL https://proceedings.mlr.press/v162/zhou22g.html.

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*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/86c17de05579cde52025f9984e6e2ebb-Abstract-Conference.html.

A Dataset Details

For all seven datasets, we use the preprocessed forecasting versions provided by TSLib, with the chronological splits described in Section 4.1. The descriptions below distinguish the original data sources from the benchmark versions used in our experiments.

ETT. The Electricity Transformer Temperature (ETT) datasets were introduced by Zhou et al. (2021). They contain load and oil-temperature measurements from two electricity transformers, covering approximately two years from July 2016. Each dataset contains seven variables: oil temperature and six power-load measurements. ETTh1 and ETTh2 provide hourly observations, while ETTm1 and ETTm2 provide observations at 15-minute intervals; the suffixes 1 and 2 identify the two transformers. The data are publicly available from the authors' official repository.[2]

Weather. Weather originates from the weather station of the Max Planck Institute for Biogeochemistry in Jena, Germany.[3] We use the forecasting benchmark containing 21 meteorological variables recorded at 10-minute intervals during 2020, including temperature, humidity, and air pressure.

Electricity. Electricity, also referred to as ECL in prior work, is derived from the UCI ElectricityLoadDiagrams20112014 dataset (Trindade, 2015). The original release contains measurements from 370 customers at 15-minute intervals. We use the preprocessed forecasting benchmark containing hourly electricity-consumption series for 321 customers from 2012 to 2014. We consistently use the name Electricity in the main text and result tables.

Traffic. Traffic originates from the Performance Measurement System (PeMS) maintained by the California Department of Transportation.[4] The forecasting benchmark contains hourly road-occupancy measurements from 862 sensors on San Francisco Bay Area freeways. The variables represent road occupancy rather than vehicle speed or traffic counts. We use the preprocessed multivariate forecasting version provided through TSLib.

B Full Forecasting Results

Table 3 reports all horizon-specific results corresponding to the averages in Table 1. It retains the same baseline sources and highlighting rules as Table 1. First-place counts and average ranks pool MSE and MAE over the 28 dataset–horizon rows and seven dataset Avg rows (70 comparisons), excluding Overall avg. This differs from the average rank in Table 1, which uses only the seven dataset Avg rows (14 comparisons across MSE and MAE). Both scopes weight the included metric-specific comparisons equally and use competition ranks at three-decimal precision.

C Additional Ablation Results

Tables 4 and 5 report MSE and relative changes for every dataset–horizon setting underlying Table 2. For each variant, ΔMSE is computed relative to C4TS in the same row: positive values (green) indicate degradation and negative values (red) indicate improvement. Nonzero changes are shown with explicit signs; no best- or second-best ranking is highlighted. The full-model reference is the same C4TS result used in the main forecasting table. Each dataset Avg is the arithmetic mean over four forecast horizons; Overall is the arithmetic mean over all 16 settings. The reported Overall ΔMSE is $100(\overline{\mathrm{MSE}}_{\mathrm{variant}}/\overline{\mathrm{MSE}}_{\mathrm{C4TS}} - 1)$, rather than the average of setting-specific percentage changes. All averages and percentage changes are calculated before rounding; MSE is displayed to four decimal places to retain small differences.

[2] https://github.com/zhouhaoyi/ETDataset

[3] https://www.bgc-jena.mpg.de/wetter/

[4] https://pems.dot.ca.gov/

Table 3: Full multivariate forecasting results underlying Table 1, with a history length of 512. Lower MSE and MAE are better. **Red bold** denotes the best result and blue underline the second best; no second result is marked when first place is tied. Avg averages four horizons; Overall avg averages seven dataset averages. The last two rows pool MSE and MAE over 28 dataset–horizon settings and seven dataset Avg rows (70 comparisons), excluding Overall avg. This ranking scope differs from Table 1, which uses only dataset Avg rows. Ties use competition ranks (e.g., 1, 1, 3) at three-decimal precision.

		Temporal–Visual				Visual		Temporal											
Dataset	H	**C4TS** (Ours)		Time-VLM (2025)		DMMV (2025)		VisionTS (2025)		APRNet (2026)		VPNet (2026)		FreqCycle (2026)		WPMixer (2025)		Time-LLM (2024)		PatchTST (2023)	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.348**	**0.377**	0.361	0.386	0.386	0.402	0.352	0.378	0.368	0.395	0.374	0.395	0.373	0.403	0.368	0.394	0.376	0.402	0.370	0.400
	192	0.386	0.405	0.397	0.415	0.418	0.429	**0.384**	**0.404**	0.405	0.417	0.412	0.422	0.404	0.425	0.420	0.418	0.407	0.421	0.413	0.429
	336	0.414	0.424	0.420	**0.421**	0.465	0.459	**0.411**	0.426	0.427	0.430	0.445	0.442	0.434	0.452	0.452	0.433	0.430	0.438	0.422	0.440
	720	0.436	0.455	0.441	0.458	0.495	0.496	0.447	0.454	**0.414**	0.463	0.484	0.481	0.443	0.461	0.449	**0.449**	0.457	0.468	0.447	0.468
	Avg	**0.396**	0.416	0.405	0.420	0.441	0.447	0.398	**0.415**	0.411	0.427	0.429	0.435	0.414	0.435	0.422	0.423	0.417	0.432	0.413	0.434
ETTh2	96	0.274	**0.331**	**0.267**	0.335	0.351	0.396	0.278	**0.331**	0.270	0.336	0.280	0.334	0.285	0.349	0.282	0.334	0.286	0.346	0.274	0.337
	192	0.338	0.375	**0.326**	**0.373**	0.481	0.478	0.345	0.377	0.330	0.374	0.350	0.382	0.353	0.394	0.359	0.385	0.361	0.391	0.341	0.382
	336	0.369	0.403	0.357	0.406	0.758	0.604	0.360	0.396	0.356	0.399	0.379	0.404	0.376	0.414	0.374	0.404	0.390	0.414	**0.329**	**0.384**
	720	0.398	0.434	0.412	0.449	0.707	0.613	0.392	0.427	0.386	0.427	0.426	0.442	0.416	0.447	0.405	0.427	0.405	0.434	**0.379**	**0.422**
	Avg	0.344	0.386	0.341	0.391	0.574	0.523	0.344	0.383	0.336	0.384	0.359	0.391	0.357	0.401	0.355	0.387	0.360	0.396	**0.331**	**0.381**
ETTm1	96	**0.279**	0.336	0.304	0.346	0.295	0.343	0.280	**0.329**	0.288	0.344	0.295	0.339	0.298	0.351	0.314	0.350	0.291	0.341	0.293	0.346
	192	**0.315**	**0.358**	0.332	0.366	0.342	0.368	0.334	0.364	0.330	0.366	0.339	0.363	0.336	0.374	0.358	0.375	0.341	0.369	0.333	0.370
	336	**0.347**	**0.379**	0.364	0.383	0.375	0.393	0.353	**0.379**	0.361	0.386	0.410	0.401	0.368	0.391	0.384	0.395	0.359	**0.379**	0.369	0.392
	720	0.404	0.414	**0.402**	**0.410**	0.446	0.431	0.408	0.416	0.415	0.418	0.459	0.430	0.422	0.417	0.448	0.432	0.433	0.419	0.416	0.420
	Avg	**0.336**	**0.372**	0.350	0.377	0.364	0.384	0.344	**0.372**	0.348	0.378	0.376	0.383	0.356	0.383	0.376	0.388	0.356	0.377	0.353	0.382
ETTm2	96	0.166	0.261	**0.160**	0.250	0.220	0.309	0.172	0.261	0.164	0.256	0.163	**0.245**	0.172	0.260	0.171	0.253	0.162	0.248	0.166	0.256
	192	0.221	0.294	**0.215**	0.291	0.358	0.400	0.248	0.314	0.221	0.293	0.221	**0.287**	0.224	0.299	0.234	0.294	0.235	0.304	0.223	0.296
	336	**0.269**	0.325	0.270	0.325	0.466	0.459	0.292	0.344	0.271	0.325	0.274	**0.321**	0.273	0.329	0.292	0.333	0.280	0.329	0.274	0.329
	720	0.350	0.381	**0.348**	**0.378**	0.671	0.554	0.370	0.393	0.355	0.384	0.363	**0.378**	0.356	0.383	0.387	0.390	0.366	0.382	0.362	0.385
	Avg	0.252	0.315	**0.248**	0.311	0.429	0.431	0.270	0.328	0.252	0.314	0.256	**0.308**	0.256	0.318	0.271	0.317	0.260	0.315	0.256	0.317
Weather	96	**0.140**	0.190	0.148	0.200	0.151	0.222	0.142	0.188	0.146	0.197	0.141	**0.178**	0.148	0.201	0.162	0.204	0.155	0.199	0.149	0.198
	192	**0.186**	0.236	0.193	0.240	0.198	0.273	0.190	0.236	0.190	0.240	0.194	**0.231**	0.190	0.241	0.209	0.246	0.223	0.261	0.194	0.241
	336	**0.238**	**0.276**	0.243	0.281	0.249	0.314	0.248	0.283	0.240	0.279	0.254	0.279	0.245	0.283	0.263	0.287	0.251	0.279	0.245	0.282
	720	0.314	**0.331**	**0.312**	0.332	0.315	0.358	0.335	0.339	0.314	**0.331**	0.346	0.336	0.331	0.343	0.339	0.338	0.345	0.342	0.314	0.334
	Avg	**0.219**	0.258	0.224	0.263	0.228	0.292	0.229	0.262	0.223	0.262	0.234	**0.256**	0.228	0.267	0.243	0.269	0.243	0.270	0.226	0.264
Electricity	96	**0.125**	0.220	0.142	0.245	0.148	0.247	0.127	**0.218**	**0.125**	0.221	0.129	0.220	0.130	0.225	0.150	0.241	0.137	0.233	0.129	0.222
	192	**0.143**	0.237	0.157	0.260	0.164	0.266	0.144	**0.235**	0.150	0.246	0.150	0.242	0.147	0.241	0.162	0.252	0.152	0.247	0.147	0.240
	336	**0.159**	**0.253**	0.174	0.276	0.178	0.283	0.162	0.255	0.160	0.257	0.166	0.259	0.164	0.260	0.179	0.270	0.169	0.267	0.163	0.259
	720	0.194	0.288	0.214	0.308	0.228	0.329	0.196	0.289	**0.185**	**0.279**	0.190	0.283	0.203	0.295	0.217	0.304	0.200	0.290	0.197	0.290
	Avg	**0.155**	**0.249**	0.172	0.273	0.179	0.281	0.157	**0.249**	**0.155**	0.251	0.159	0.251	0.161	0.255	0.177	0.267	0.164	0.259	0.159	0.253
Traffic	96	0.354	0.241	0.393	0.290	0.470	0.324	0.352	0.235	0.352	0.257	**0.341**	0.246	0.372	**0.232**	0.465	0.286	0.392	0.267	0.360	0.249
	192	0.375	0.251	0.405	0.296	0.475	0.327	0.374	0.245	0.376	0.269	**0.364**	0.257	0.379	**0.236**	0.475	0.290	0.409	0.271	0.379	0.256
	336	0.385	0.254	0.420	0.305	0.479	0.333	0.393	0.255	0.392	0.277	**0.383**	0.265	0.395	**0.243**	0.489	0.296	0.434	0.296	0.392	0.264
	720	0.431	0.282	0.459	0.323	0.545	0.369	0.434	0.283	0.433	0.299	**0.427**	0.284	0.447	**0.260**	0.527	0.318	0.451	0.291	0.432	0.286
	Avg	0.386	0.257	0.419	0.303	0.492	0.338	0.388	0.255	0.388	0.276	**0.379**	0.263	0.398	**0.243**	0.489	0.297	0.421	0.281	0.391	0.264
Overall avg		**0.298**	**0.322**	0.308	0.334	0.387	0.385	0.304	0.323	0.302	0.327	0.313	0.327	0.310	0.329	0.333	0.335	0.317	0.333	0.304	0.328
1st count		**24**		12		0		11		6		13		5		1		1		6	
Average rank		**2.243**		4.814		9.300		3.700		3.471		4.957		6.043		7.629		6.686		4.857	

Table 4: Co-encoding ablations on ETT. Each variant reports MSE and ΔMSE (%) relative to the C4TS reference in the same row. Positive changes (green) indicate degradation; negative changes (red) indicate improvement. Avg averages four horizons; Overall averages all 16 settings. Changes in average rows are computed from the corresponding mean MSE values.

Dataset	H	**C4TS**	w/o spatial adaptation		w/o adaptation & interaction		+ temporal-to-visual cross-attention‡	
		MSE	MSE	ΔMSE	MSE	ΔMSE	MSE	ΔMSE
ETTh1	96	0.3485	0.3496	+0.313%	0.3488	+0.097%	0.3627	+4.089%
	192	0.3857	0.3861	+0.118%	0.3851	-0.154%	0.3909	+1.360%
	336	0.4137	0.4118	-0.452%	0.4152	+0.359%	0.4113	-0.591%
	720	0.4362	0.4382	+0.449%	0.4327	-0.813%	0.4546	+4.213%
	Avg	0.3960	0.3964	+0.103%	0.3954	-0.146%	0.4049	+2.237%
ETTh2	96	0.2737	0.2754	+0.650%	0.2733	-0.122%	0.2949	+7.762%
	192	0.3377	0.3417	+1.177%	0.3371	-0.177%	0.3390	+0.374%
	336	0.3685	0.3708	+0.609%	0.3653	-0.886%	0.3596	-2.432%
	720	0.3977	0.3989	+0.306%	0.3963	-0.353%	0.3931	-1.154%
	Avg	0.3444	0.3467	+0.669%	0.3430	-0.407%	0.3466	+0.650%
ETTm1	96	0.2794	0.2796	+0.063%	0.2788	-0.224%	0.2941	+5.272%
	192	0.3152	0.3182	+0.935%	0.3178	+0.826%	0.3283	+4.144%
	336	0.3469	0.3518	+1.393%	0.3480	+0.318%	0.3534	+1.849%
	720	0.4041	0.4049	+0.201%	0.4017	-0.596%	0.3997	-1.076%
	Avg	0.3364	0.3386	+0.652%	0.3366	+0.050%	0.3439	+2.219%
ETTm2	96	0.1656	0.1655	-0.075%	0.1658	+0.109%	0.1651	-0.326%
	192	0.2213	0.2256	+1.959%	0.2263	+2.269%	0.2316	+4.679%
	336	0.2691	0.2732	+1.541%	0.2760	+2.569%	0.3085	+14.668%
	720	0.3504	0.3545	+1.165%	0.3615	+3.183%	0.3728	+6.413%
	Avg	0.2516	0.2547	+1.236%	0.2574	+2.312%	0.2695	+7.130%
Overall		0.3321	0.3341	+0.603%	0.3331	+0.302%	0.3412	+2.747%

‡ Adds temporal-to-visual cross-attention to the C4TS baseline while retaining visual-to-temporal interaction.

Co-encoding configurations. The variant *w/o spatial adaptation* removes convolutional spatial mixing and instead flattens the encoded visual history and applies a linear projection before cross-attention. Thus, visual-to-temporal feature interaction remains active. The variant *w/o spatial adaptation & feature interaction* removes both the adapter and cross-attention, while retaining the two forecasting pathways and prediction fusion. The variant + *temporal-to-visual cross-attention* adds reverse feature interaction to the full C4TS model while retaining its visual-to-temporal interaction, testing whether bidirectional interaction improves forecasting.

Pathway predictions. *Temporal-path prediction* and *Visual-path prediction* use the respective outputs $\hat{\mathbf{Y}}_{\mathrm{T}}$ and $\hat{\mathbf{Y}}_{\mathrm{V}}$ of the full, jointly trained model. These results measure the effect of selecting an individual output instead of the fused forecast; they do not represent separately trained single-path architectures.

Co-decoding configurations. *Uniform averaging* computes $\frac{1}{2}(\hat{\mathbf{Y}}_{\mathrm{T}} + \hat{\mathbf{Y}}_{\mathrm{V}})$ and omits the fused forecasting loss during training. *Global scalar fusion* uses a learnable scalar shared across samples and future patches. *Instance-wise fusion* adapts the weights to each input instance but shares them across future patches, whereas *Future-patch-wise fusion* learns future-patch-specific weights shared across instances. C4TS adapts the weights jointly to the input instance and future patch.

D Visualizations of Temporal–Visual Collaboration

Spatially adapted attention. Figure 3(a) compares cross-attention weights for the first input instance on ETTm1 with $H = 96$. The left panel shows full C4TS (labeled Baseline in the image); the right panel shows the same *w/o spatial adaptation* variant evaluated in Section 4.3, which replaces spatial adaptation with direct flattening and linear projection of MAE tokens. Both retain visual-to-temporal feature interaction. Rows index temporal input patches (queries), and columns index historical visual token positions in flattened spatial order (keys). In the full model, the keys

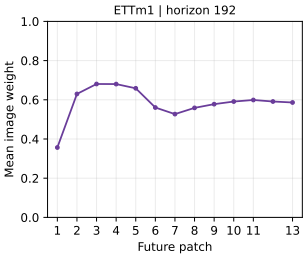
Table 5: Pathway predictions and co-decoding ablations on ETT. Each variant reports MSE and ΔMSE (%) relative to the C4TS reference in the same row. Positive changes (green) indicate degradation; negative changes (red) indicate improvement. Avg averages four horizons; Overall averages all 16 settings. Changes in average rows are computed from the corresponding mean MSE values.

Dataset	H	**C4TS**	Temporal path		Visual path		Uniform averaging†		Global scalar		Instance wise		Future-patch wise	
		MSE	MSE	ΔMSE	MSE	ΔMSE	MSE	ΔMSE	MSE	ΔMSE	MSE	ΔMSE	MSE	ΔMSE
ETTh1	96	0.3485	0.3828	+9.863%	0.3541	+1.602%	0.3515	+0.880%	0.3520	+1.009%	0.3488	+0.086%	0.3510	+0.722%
	192	0.3857	0.4183	+8.461%	0.3906	+1.266%	0.3916	+1.545%	0.3919	+1.616%	0.3854	-0.075%	0.3909	+1.357%
	336	0.4137	0.4482	+8.352%	0.4195	+1.399%	0.4143	+0.149%	0.4145	+0.188%	0.4108	-0.689%	0.4135	-0.054%
	720	0.4362	0.4936	+13.150%	0.4466	+2.364%	0.4449	+1.977%	0.4453	+2.070%	0.4373	+0.241%	0.4443	+1.840%
	Avg	0.3960	0.4358	+10.032%	0.4027	+1.677%	0.4006	+1.153%	0.4009	+1.235%	0.3956	-0.113%	0.3999	+0.982%
ETTh2	96	0.2737	0.2879	+5.209%	0.2898	+5.907%	0.2744	+0.267%	0.2741	+0.168%	0.2720	-0.610%	0.2731	-0.203%
	192	0.3377	0.3488	+3.288%	0.3528	+4.459%	0.3361	-0.467%	0.3364	-0.395%	0.3406	+0.843%	0.3354	-0.695%
	336	0.3685	0.3822	+3.721%	0.3842	+4.259%	0.3640	-1.225%	0.3637	-1.298%	0.3697	+0.330%	0.3627	-1.571%
	720	0.3977	0.4091	+2.858%	0.4125	+3.712%	0.4005	+0.688%	0.4002	+0.614%	0.3980	+0.082%	0.3992	+0.362%
	Avg	0.3444	0.3570	+3.661%	0.3598	+4.477%	0.3437	-0.191%	0.3436	-0.233%	0.3451	+0.197%	0.3426	-0.526%
ETTm1	96	0.2794	0.3007	+7.647%	0.2947	+5.490%	0.2798	+0.142%	0.2799	+0.171%	0.2806	+0.444%	0.2785	-0.302%
	192	0.3152	0.3375	+7.086%	0.3262	+3.502%	0.3164	+0.373%	0.3165	+0.418%	0.3188	+1.151%	0.3153	+0.030%
	336	0.3469	0.3721	+7.241%	0.3508	+1.110%	0.3460	-0.260%	0.3462	-0.209%	0.3478	+0.234%	0.3451	-0.546%
	720	0.4041	0.4272	+5.707%	0.4121	+1.975%	0.4046	+0.130%	0.4056	+0.364%	0.4034	-0.168%	0.4045	+0.095%
	Avg	0.3364	0.3594	+6.828%	0.3460	+2.840%	0.3367	+0.089%	0.3370	+0.189%	0.3377	+0.372%	0.3358	-0.168%
ETTm2	96	0.1656	0.1704	+2.872%	0.1965	+18.657%	0.1687	+1.893%	0.1685	+1.780%	0.1639	-1.038%	0.1675	+1.122%
	192	0.2213	0.2252	+1.754%	0.2624	+18.582%	0.2264	+2.301%	0.2246	+1.515%	0.2250	+1.661%	0.2236	+1.054%
	336	0.2691	0.2761	+2.625%	0.3048	+13.292%	0.2782	+3.397%	0.2736	+1.684%	0.2761	+2.635%	0.2726	+1.304%
	720	0.3504	0.3764	+7.421%	0.3715	+6.016%	0.3559	+1.577%	0.3567	+1.798%	0.3563	+1.680%	0.3556	+1.479%
	Avg	0.2516	0.2620	+4.144%	0.2838	+12.805%	0.2573	+2.275%	0.2559	+1.702%	0.2553	+1.484%	0.2548	+1.280%
Overall		0.3321	0.3535	+6.454%	0.3481	+4.805%	0.3346	+0.748%	0.3344	+0.678%	0.3334	+0.393%	0.3333	+0.356%

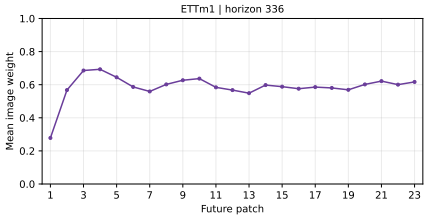
† Uniform averaging omits the fused forecasting loss. Temporal and visual pathway predictions are outputs of the full model, not separately trained single-path models.

are spatially adapted visual features. Darker colors indicate larger attention probabilities, with a shared color scale across panels. The comparison illustrates attention redistribution rather than a direct measure of token usefulness.

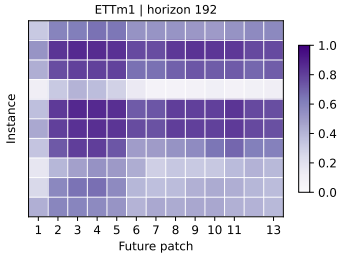
Adaptive prediction fusion. Figure 3(b,c) presents the $H = 96$ visual fusion weights on ETTm1, and Figure 4 extends the analysis to $H \in \{192, 336, 720\}$. Each curve averages the weights over all test instances. Each heatmap shows ten instances sampled at regular intervals from the test sequence. With a future-patch length of 15, the four horizons contain 7, 13, 23, and 48 patches, respectively; a shorter final patch is retained when needed. The mean curves show positional preferences, while the heatmaps expose variation across instances that averaging can obscure. The selected instances illustrate within-horizon variation and are not assumed to match across horizons.



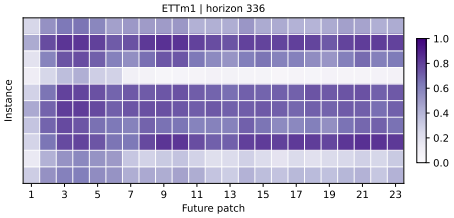
(a) Mean weights, $H = 192$



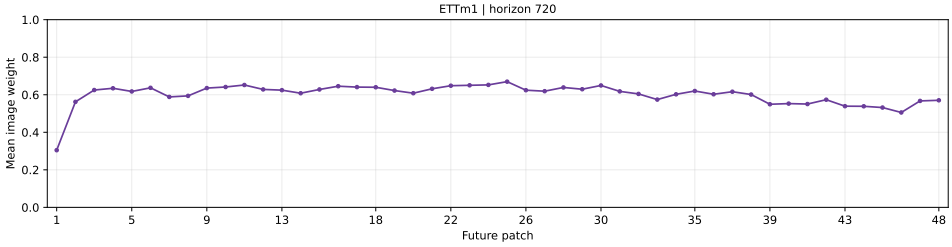
(b) Mean weights, $H = 336$



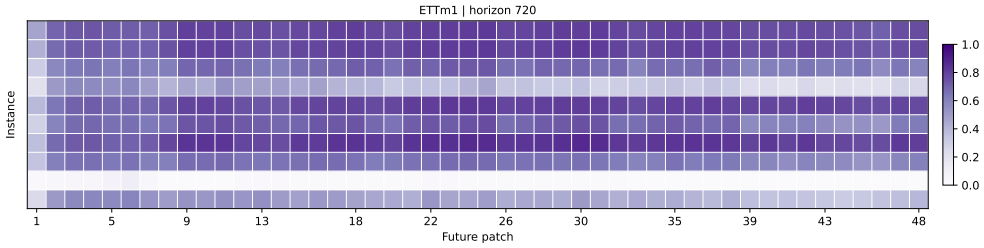
(c) Instance weights, $H = 192$



(d) Instance weights, $H = 336$



(e) Mean weights, $H = 720$



(f) Instance weights, $H = 720$

Figure 4: Visual fusion weights on ETTm1 for $H = 192, 336, 720$. Panels (a,b) show visual weights averaged over all test instances for $H = 192, 336$, respectively; panels (c,d) show the corresponding heatmaps for ten regularly spaced test instances. Panels (e,f) show the mean weights and heatmap for $H = 720$. Heatmap rows index instances and columns index future patches. Darker cells indicate larger visual weights on the shared scale $[0, 1]$; the temporal weight is the complement. Results for $H = 96$ appear in Figure 3(b,c).