STCFORMER: ADAPTIVE SPATIO-TEMPORAL MODELING WITH DYNAMIC CLUSTER TRANSFORMER FOR STATION-BASED WEATHER FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Station-based weather forecasting supports daily life and economic activity, yet accurate forecasts require modeling complex spatial dependencies among stations. Recent clustering-based selective modeling offers a promising alternative to dense inter-station interactions. However, a grouping shared across an observation window may obscure local changes in station relationships, while intra-cluster interactions alone may miss important global context. The theoretical advantages of selective interactions over dense connectivity also remain insufficiently understood. We therefore propose STCFormer, an adaptive spatio-temporal Transformer that dynamically groups stations according to their local evolution within each temporal patch. Its Cluster-Guided Attention Block combines fine-grained local attention within clusters and global attention over regional state summaries, allowing each station to access information beyond its own cluster. We further show that a derived Lipschitz upper bound for cluster-conditioned local attention is no larger than its fully connected counterpart, explaining a potential robustness benefit and motivating the design of InfoLoss. Experiments on three real-world weather datasets spanning eight temperature and wind forecasting tasks show that STCFormer achieves the lowest 24-hour mean squared error on all eight tasks and ranks first or second in 47 of 48 comparisons across metrics and forecasting horizons. Ablations and case studies further confirm the benefits of locally adaptive grouping and complementary local–global interactions.

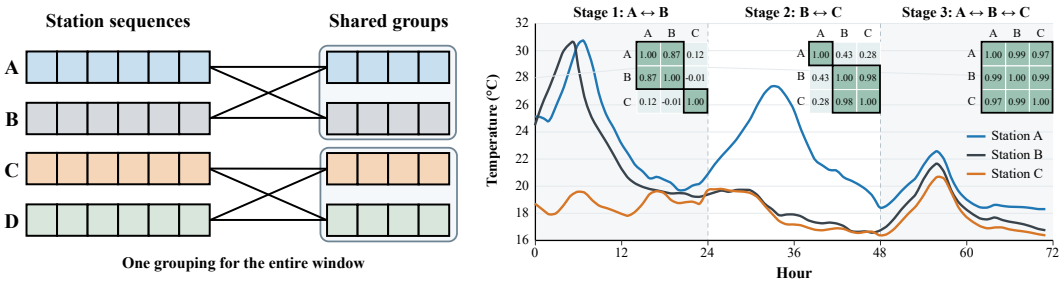
1 INTRODUCTION

Station-based weather forecasting supports agriculture, transportation, and many other applications Lazo et al. (2009); Freebairn & Zillman (2002). The widespread deployment of automatic weather stations has increased the demand for accurate station-level predictions, particularly for fine-grained urban forecasting Baklanov et al. (2018); Rolnick et al. (2022). Shaped by local environments and atmospheric processes, station observations exhibit complex temporal variations and spatial relationships. Effectively capturing these spatio-temporal dependencies remains a key challenge in station-based weather forecasting.

Existing studies approach this challenge through Spatio-Temporal (ST) and Multivariate Time Series (MTS) forecasting. ST methods incorporate spatial structures, spatio-temporal priors, or physical processes Li et al. (2018); Wu et al. (2023). MTS methods typically treat stations as channels, modeling them independently or capturing their dependencies through channel mixing Nie et al. (2023); Chen et al. (2024); Qiu et al. (2025). Because geography and atmospheric processes induce heterogeneous dependencies, dense, fine-grained interactions between all station pairs are not always necessary. Recent clustering-based methods offer a promising strategy for selective dependency modeling by organizing related sequences into groups Chen et al. (2024); Qiu et al. (2025); Hu et al. (2025). Figure 1(a) illustrates this grouping-based perspective.

Despite the promising progress, two key challenges remain. **(1) Window-level grouping can obscure local relationship changes and restrict useful information exchange.** Groups estimated from complete sequences and shared across a window may overlook changing station dependencies as weather processes evolve. In the real French-temperature dataset case in Figure 1(b), A and B are

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.



(a) Shared grouping over a window **(b) Relationships evolve within the window**

Figure 1: Motivation for locally adaptive station grouping. (a) A single grouping is shared across an observation window. (b) A real 72-hour case from the French temperature dataset shows the strongest relationship shifting from A–B to B–C before all three stations become highly correlated. Insets report Pearson correlations within each 24-hour segment.

strongly correlated in the first 24-hour segment, B and C in the second, and all three stations in the third. A single grouping can therefore blend distinct local relationships. Moreover, when groups restrict interactions to their members, useful cross-region information may be lost: atmospheric circulation and propagating weather systems can connect stations with different local evolution patterns. Effective selective modeling should thus adapt to local relationship changes while retaining access to global context. **(2) Clustering-based selective interaction still lacks sufficient theoretical support.** Its effectiveness is primarily established empirically, leaving its advantages over dense interaction and potential robustness benefits insufficiently explained. A theoretical account is needed to complement empirical evidence and inform the design of selective interactions.

To address these challenges, we propose **STCFormer**, an adaptive spatio-temporal Transformer. Joint spatio-temporal patch embedding represents observations at a local temporal granularity. Within each temporal patch, efficient EM-style clustering uses evolution representations to dynamically organize stations into regional units. The Cluster-Guided Attention Block (CGAB) first performs hard-routed local attention within clusters, then aggregates the updated state features into regional content centers. Global station-to-region attention allows each station to access summaries from all regions, combining selective local interactions with global context. Theoretically, we show that the derived Lipschitz upper bound for fixed-partition local attention is no larger than the corresponding fully connected bound. We further connect this bound to station–cluster mutual information through the largest cluster size, motivating our mutual-information clustering regularizer, InfoLoss. Experiments on three real-world weather datasets demonstrate strong predictive performance and support the proposed design.

Our main contributions are as follows:

(1) We propose STCFormer, combining evolution-driven grouping at each temporal patch with cluster-guided local–global attention to capture changing station dependencies while preserving global context.
(2) We establish a conditional sensitivity analysis of clustering-based attention, compare its derived Lipschitz upper bound with the dense counterpart, and connect the bound to mutual information to motivate InfoLoss.
(3) Experiments on three real-world datasets covering eight forecasting tasks demonstrate competitive performance against 15 baselines, while ablations and case studies validate the benefits of locally adaptive grouping and complementary local–global interactions.

2 STCFormer

Let $\mathcal{X}_H \in \mathbb{R}^{T\times N\times C}$ denote C meteorological variables observed at N stations over the past T time steps. Our goal is to predict the next O steps, $\mathcal{Y}_F \in \mathbb{R}^{O\times N\times C}$. Figure 2 gives an overview of STCFormer. The model combines locally adaptive station grouping with access to global spatial context. Spatio-Temporal Patch Embedding forms compact tokens from local observations and geographic coordinates. Each subsequent block constructs two temporal views: a state view provides

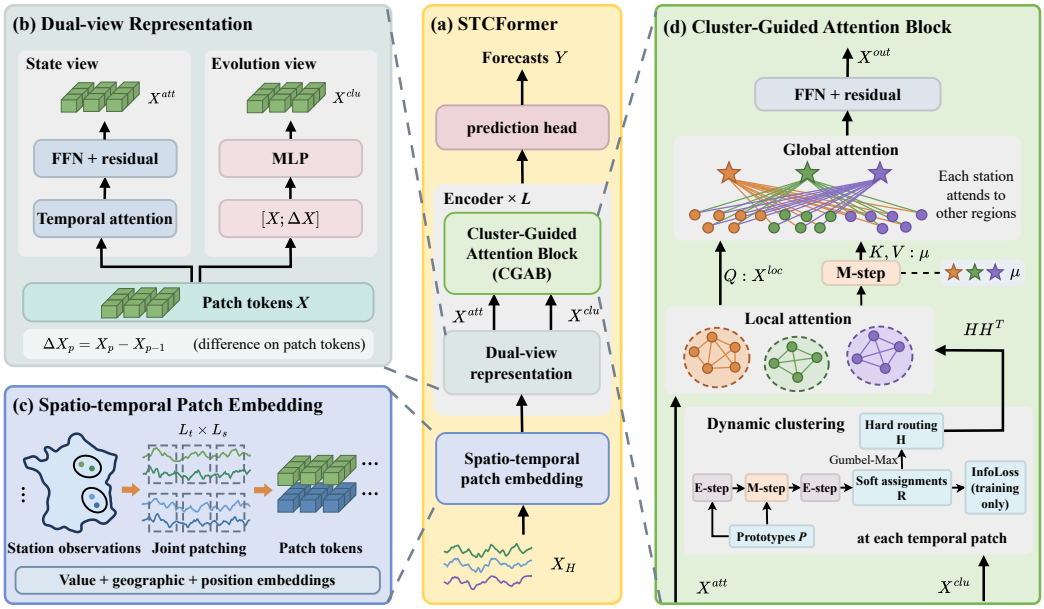


Figure 2: Overview of STCFormer. (a) The forecasting pipeline stacks CGAB encoders over patch embeddings and dual-view features. (b) Dual-view representation separates temporal state and local evolution. (c) Spatio-temporal patch embedding forms joint station–time tokens. (d) CGAB combines dynamic clustering, local attention within clusters, and global attention over regional summaries.

contextualized features for prediction, while an evolution view guides grouping at each temporal patch. The Cluster-Guided Attention Block (CGAB) first models dependencies within the resulting groups, then enables each station to attend to content summarized from all regions. A direct prediction head maps the encoded features to future observations. We then analyze how grouping controls a local-attention sensitivity bound and use the resulting mutual-information connection to motivate InfoLoss for within-patch assignments.

2.1 Spatio-Temporal Patch Embedding

Patch embedding is an effective way to reduce sequence length in multivariate forecasting Nie et al. (2023); Huang et al. (2024), while conventional temporal patches encode each variable separately without explicitly incorporating neighboring stations. We construct joint spatio-temporal patches to retain local spatial context. Before patching, stations are grouped into latitude bands and ordered by longitude in alternating directions across adjacent bands. This snake-like ordering limits jumps at band boundaries and places geographically nearby stations at nearby indices Wu et al. (2023). The temporal and station axes are padded to multiples of the patch lengths when necessary.

Let L_t and L_s be the temporal and spatial patch lengths, and let T' and N' be the resulting numbers of patches. The final feature or variable axis is implicit in all slices. The (p, s)-th patch is

$$\mathcal{X}^{(\mathrm{pa})}_{p,s} = \mathcal{X}_H[(p-1)L_t : pL_t, (s-1)L_s : sL_s] \in \mathbb{R}^{L_t \times L_s \times C}. \tag{1}$$

A value projector encodes each flattened patch, while a lightweight geographic projector embeds the normalized latitude and longitude coordinates $\mathbf{c}_s$ within each spatial patch. Together with a learnable temporal-patch position embedding $\mathbf{e}^{(\mathrm{pos})}_p$, the input representation is

$$\mathcal{X}_{p,s} = f_{\mathrm{val}}\Big(\mathrm{vec}(\mathcal{X}^{(\mathrm{pa})}_{p,s})\Big) + f_{\mathrm{geo}}(\mathbf{c}_s) + \mathbf{e}^{(\mathrm{pos})}_p. \tag{2}$$

The embedded tensor $\mathcal{X} \in \mathbb{R}^{T' \times N' \times D}$ reduces the token grid from $T \times N$ to $T' \times N'$ while retaining local spatio-temporal and geographic context for subsequent attention.

2.2 DUAL-VIEW TEMPORAL REPRESENTATION

We construct two temporal views to support predictive information exchange and station grouping according to local evolution. For spatial patch s, the temporal slice $\mathcal{X}_{:,s} \in \mathbb{R}^{T' \times D}$ contains its features across all temporal patches, where a colon denotes all entries along an axis. Specifically, the state view uses temporal self-attention to capture temporal dependencies across the observation window, followed by a feed-forward update:

$$\mathcal{X}^{\text{temp}}_{:,s} = \text{SelfAttention}(\mathcal{X}_{:,s}), \qquad \mathcal{X}^{\text{att}}_{:,s} = \mathcal{X}^{\text{temp}}_{:,s} + \text{FFN}(\mathcal{X}^{\text{temp}}_{:,s}). \tag{3}$$

The resulting state view $\mathcal{X}^{\text{att}}$ retains temporal context, providing the feature content for subsequent local and global spatial attention. To make grouping responsive to changes within the observation window, we additionally construct an evolution view at each temporal patch. It incorporates local changes by concatenating each input token with its first-order temporal difference:

$$\Delta\mathcal{X}_{p,s} = \mathcal{X}_{p,s} - \mathcal{X}_{p-1,s}, \qquad \mathcal{X}^{\text{clu}}_{p,s} = \text{MLP}_{\text{clu}}([\mathcal{X}_{p,s}; \Delta\mathcal{X}_{p,s}]). \tag{4}$$

Here the difference is defined for $p = 2, \ldots, T'$, with $\Delta\mathcal{X}_{1,s} = \mathbf{0}$, and concatenation is along the feature axis. The patch-aligned views give CGAB temporally contextualized state features for attention and local evolution features for patch-specific grouping.

2.3 CLUSTER-GUIDED ATTENTION BLOCK (CGAB)

We propose the Cluster-Guided Attention Block to combine time-varying station interactions with global spatial context. CGAB comprises two modules: Dynamic Station Clustering and Cluster-Guided Local–Global Attention. For notational convenience, we take temporal patch p as an example and write $\mathcal{X}^{\text{clu}}_{p,:} = \mathbf{X}^{\text{clu}} = [\mathbf{x}^{\text{clu}}_1, \ldots, \mathbf{x}^{\text{clu}}_{N'}]^\top$ and $\mathcal{X}^{\text{att}}_{p,:} = \mathbf{X}^{\text{att}} = [\mathbf{x}^{\text{att}}_1, \ldots, \mathbf{x}^{\text{att}}_{N'}]^\top$, both in $\mathbb{R}^{N' \times D}$. Each row corresponds to a spatial patch, referred to as a station token, and the time index is omitted below.

2.3.1 DYNAMIC STATION CLUSTERING

To capture relationship changes within an observation window, we group the N' station tokens into K clusters separately at each temporal patch using a differentiable EM-style procedure Zhou et al. (2024); Li et al. (2024). Each layer maintains learnable prototypes $\mathbf{P} = [\mathbf{p}_1, \ldots, \mathbf{p}_K]^\top \in \mathbb{R}^{K \times D}$. These shared prototypes provide a common reference across patches, while patch-specific updates allow a station to change groups as its local evolution changes. We perform one E–M update, followed by a final E-step to obtain the assignments used for spatial interaction.

Initial E-step. Using the prototypes as initial centroids, we compute provisional soft assignments from the evolution descriptors:

$$\widetilde{r}_{s,k} = \text{softmax}_k\left(\frac{\langle \mathbf{x}^{\text{clu}}_s, \mathbf{p}_k \rangle}{\tau}\right), \tag{5}$$

where $\tau > 0$ is a learned temperature. **M-step.** The assignments produce a data-dependent center for each cluster, which is blended with its learnable prototype through an EMA-style update:

$$\bar{\mathbf{c}}_k = \frac{\sum_{s=1}^{N'} \widetilde{r}_{s,k} \mathbf{x}^{\text{clu}}_s}{\max(\sum_{s=1}^{N'} \widetilde{r}_{s,k}, \epsilon)}, \qquad \mathbf{c}_k = \eta \bar{\mathbf{c}}_k + (1-\eta)\mathbf{p}_k. \tag{6}$$

Here $\eta \in [0, 1]$ balances the current patch's evolution information and the learned prototype, while $\epsilon > 0$ prevents division by zero. This blend is computed within the current patch and does not carry centroids across time. Finally, we repeat the E-step in Eq. (5) with the updated centers $\mathbf{c}_k$ replacing $\mathbf{p}_k$, obtaining the final soft assignments $\mathbf{R} = [r_{s,k}] \in [0, 1]^{N' \times K}$ for InfoLoss, hard local routing, and soft regional aggregation. During training, Gumbel-Max samples the cluster index

$$\kappa_s = \arg\max_k \big(\log r_{s,k} + g_{s,k}\big), \qquad g_{s,k} \sim \text{Gumbel}(0, 1). \tag{7}$$

The hard assignment matrix $\mathbf{H} \in \{0, 1\}^{N' \times K}$ has $H_{s,k} = 1$ iff $k = \kappa_s$. During training, a straight-through Gumbel–Softmax estimator preserves hard routing in the forward pass while enabling

backpropagation through the assignment probabilities Huijben et al. (2023). At inference, we omit the noise and set $\kappa_s = \arg\max_k r_{s,k}$. Recomputing this E–M–E procedure at each temporal patch lets groups track local changes within the observation window. Its hard assignments determine local attention neighborhoods, while the corresponding soft assignments carry graded membership into regional aggregation.

2.3.2 Cluster-Guided Local–Global Attention

Tokens are grouped according to similarity in their local evolution representations. We express local attention over the state features $\mathbf{X}^{\text{att}}$ using $\mathbf{H}\mathbf{H}^\top$ as a mask; our experiments implement the same hard partition with efficient packed attention (Appendix B):

$$\mathbf{X}^{\text{loc}} = \mathbf{X}^{\text{att}} + \text{Softmax}\left(\frac{(\mathbf{X}^{\text{att}}\mathbf{W}_Q^{\text{loc}})(\mathbf{X}^{\text{att}}\mathbf{W}_K^{\text{loc}})^\top}{\sqrt{D}} + \log(\mathbf{H}\mathbf{H}^\top)\right)(\mathbf{X}^{\text{att}}\mathbf{W}_V^{\text{loc}}). \tag{8}$$

At each temporal patch, the mask restricts this attention step to station tokens assigned to the same region. The state view can therefore model fine-grained dependencies among stations with similar recent evolution.

Stations with different local evolution patterns may still provide useful predictive context. We therefore complement local attention with global station-to-region attention over all regional content centers. Specifically, the regional M-step aggregates the locally refined state features $\mathbf{X}^{\text{loc}}$ with the soft assignments $\mathbf{R}$. Let $\mathbf{D}_R = \text{Diag}(\max(\mathbf{R}^\top\mathbf{1}, \epsilon))$, where $\epsilon = 10^{-6}$ and the maximum is elementwise. We then perform cross-attention with $\mathbf{X}^{\text{loc}}$ as queries and $\boldsymbol{\mu}$ as keys and values:

$$\boldsymbol{\mu} = \mathbf{D}_R^{-1}\mathbf{R}^\top\mathbf{X}^{\text{loc}}, \qquad \mathbf{X}^{\text{glo}} = \mathbf{X}^{\text{loc}} + \text{Softmax}\left(\frac{(\mathbf{X}^{\text{loc}}\mathbf{W}_Q^{\text{glo}})(\boldsymbol{\mu}\mathbf{W}_K^{\text{glo}})^\top}{\sqrt{D}}\right)(\boldsymbol{\mu}\mathbf{W}_V^{\text{glo}}). \tag{9}$$

The regional centers $\boldsymbol{\mu}$ summarize state content, whereas the updated clustering centers $\{\mathbf{c}_k\}_{k=1}^K$ represent evolution for grouping. Soft weights retain each token's degree of membership when forming these centers, even though local attention uses hard cluster boundaries. Each station queries all regional centers, accessing global context beyond its own group without all-to-all station attention. We then apply a feed-forward update for nonlinear transformation:

$$\mathbf{X}^{\text{out}} = \mathbf{X}^{\text{glo}} + \text{FFN}(\mathbf{X}^{\text{glo}}). \tag{10}$$

Thus, local attention captures fine-grained dependencies within evolving clusters, while global attention shares information across clusters through compact regional summaries. This separation preserves selective local interactions without isolating stations from other regions. The following sensitivity analysis focuses on local attention; the equations above use single-head notation for clarity.

Direct prediction. For spatial patch s, we flatten the encoded temporal features $\mathbf{F}_s \in \mathbb{R}^{T' \times D}$ and predict all OL_s future values for one target variable:

$$\widehat{\mathbf{Y}}_s = \text{Reshape}_{O \times L_s}(\text{MLP}_{\text{pred}}(\text{LN}(\text{vec}(\mathbf{F}_s)))). \tag{11}$$

The MLP uses two affine layers with GELU and dropout between them. The patch outputs are assembled, unpadded, and reordered to recover forecasts in the original station order.

2.4 Lipschitz Bounds and Mutual Information Regularization

In this section, we analyze the potential sensitivity advantage of cluster-conditioned local attention over dense attention. Following bounded-domain analyses of attention (Castin et al., 2024), we compare the two operators under identical projection weights and feature-norm constraints. At one temporal patch, let $n = N'$, fix a hard partition H, and denote its largest cluster size by $m(H)$. Let $\mathcal{A}_H$ be the single-head residual local-attention operator in Eq. (8), and $\mathcal{A}_{\text{dense}}$ its fully connected counterpart. Features use the Frobenius norm on $\Omega_{R_X} = \{U : \max_s \|u_s\|_2 \le R_X\}$, and weights use the spectral norm.

Theorem 2.1 (Grouping-dependent sensitivity bound)**.** *Let* $v = \|W_V^{\text{loc}}\|_2$ *and*

$$\beta = \frac{R_X^2 \|W_Q^{\text{loc}}(W_K^{\text{loc}})^\top\|_2}{\sqrt{D}}.$$

For $q \geq 1$, define the nondecreasing bound

$$B(q) = 1 + v \left\lfloor \beta + (1+\beta)\sqrt{\frac{qe^{2\beta}}{q-1+e^{2\beta}}} \right\rfloor . \tag{12}$$

Then the derived Lipschitz upper bounds satisfy

$$B_{\text{ours}}(H) := B(m(H)) \leq B(n) =: B_{\text{dense}}. \tag{13}$$

Moreover, $\text{Lip}_F(\mathcal{A}_H) \leq B_{\text{ours}}(H)$ *and* $\text{Lip}_F(\mathcal{A}_{\text{dense}}) \leq B_{\text{dense}}$.

Since $m(H) \leq n$ and B is nondecreasing, the derived local-attention bound is no larger than the dense bound and is strictly tighter when $m(H) < n$ and $v, \beta > 0$. The maximum cluster size therefore controls the local-attention certificate. This observation motivates a loss that favors informative, non-degenerate assignments rather than allowing cluster collapse.

Mutual information clustering loss. Let $R \in [0,1]^{n \times K}$ be row-stochastic, with $1 \leq K \leq n$. Let X be a station token sampled uniformly within the patch and Y its cluster label, with $\Pr(Y = k \mid X = \mathbf{x}_s^{\text{clu}}) = R_{s,k}$. Using natural logarithms, we maximize the within-patch mutual information $I(X;Y) = H(Y) - H(Y \mid X)$ (Gomes et al., 2010). Restoring the temporal-patch index p, write

$$\pi_{p,k} = \frac{1}{N'} \sum_s R_{p,s,k}, \qquad H(Y_p \mid X_p) = \frac{1}{N'} \sum_s H(\mathbf{R}_{p,s,:}), \tag{14}$$

where $H(\mathbf{r}) = -\sum_k r_k \log r_k$ and $H(Y_p) = H(\boldsymbol{\pi}_p)$. We minimize the negative mutual information

$$\mathcal{L}_{\text{info}} = -\sum_{\ell=1}^{L} \frac{1}{T'} \sum_{p=1}^{T'} I(X_p^{(\ell)}; Y_p^{(\ell)}) = \sum_{\ell=1}^{L} \frac{1}{T'} \sum_{p=1}^{T'} \left[\frac{1}{N'} \sum_s H(\mathbf{R}_{p,s,:}^{(\ell)}) - H(\boldsymbol{\pi}_p^{(\ell)}) \right]. \tag{15}$$

Here L is the number of encoder layers: InfoLoss is computed at every layer, averaged over its temporal patches, and summed over layers. Lower conditional entropy encourages confident assignments, while higher marginal entropy encourages balanced cluster use. We train with $\mathcal{L} = \mathcal{L}_{\text{pred}} + \lambda \mathcal{L}_{\text{info}}$, using MSE for $\mathcal{L}_{\text{pred}}$ and a single regularization weight λ. The following theorem formalizes why this objective is relevant: increasing station–cluster mutual information tightens a probabilistic certificate on the largest sampled group, which in turn tightens the derived local-attention bound.

Theorem 2.2 (Mutual-information control of sampled grouping)**.** *Conditional on R, sample token labels independently according to its rows, as in training-time Gumbel-Max. For $\delta \in (0,1)$, define*

$$M_\delta(I) = \min\left\{ n,\ \frac{n}{K} + \sqrt{\left(\frac{n^2}{2} + \frac{n}{\delta}\right)(\log K - I)} \right\}. \tag{16}$$

Then, with conditional probability at least $1-\delta$,

$$m(H) \leq M_\delta(I(X;Y)), \qquad B_{\text{ours}}(H) \leq B(M_\delta(I(X;Y))) . \tag{17}$$

At fixed n, K, v, β, δ, increasing mutual information tightens or preserves this probabilistic certificate. Together, the two results connect the training objective to the sensitivity analysis: InfoLoss promotes informative assignments, the mutual-information theorem controls the largest sampled group, and the grouping theorem maps that size to the local-attention bound. The failure probability δ is an analysis parameter, not a training hyperparameter. Appendix D gives the proofs and a deterministic argmax counterpart; Appendix E discusses further loss properties.

3 Experiments

3.1 Experimental Setup

Datasets and Baselines. We evaluate STCFormer on three real-world weather datasets—French, Hunan, and Global—covering eight temperature and wind forecasting tasks. All datasets use a 6:2:2 training/validation/test split and Z-score normalization based on training-set statistics. We compare with 15 baselines: six spatio-temporal methods and nine multivariate time series methods. Dataset descriptions and the complete baseline list are provided in Appendices F.1 and F.2.

Table 1: 24-hour forecasting results (MSE, ↓) with a 48-hour look-back window. Values are mean ± standard deviation over three runs. Bold and underlined means indicate the best and second-best reported values in each column; ties share the same marking.

Method	French			Hunan			Global	
	Temp.	U-Wind	V-Wind	Temp.	U-Wind	V-Wind	Temp.	Wind
Spatio-Temporal Methods								
DCRNN	6.428±0.062	6.111±0.159	8.091±0.133	<u>4.921</u>±2.835	1.947±0.007	2.815±0.039	8.401±0.138	3.675±0.005
Corrformer	7.911±0.161	4.700±0.068	6.411±0.001	5.594±0.723	2.044±0.006	2.640±0.030	7.709±0.009	3.889±0.004
EasyST	7.823±0.697	4.735±0.092	5.884±0.047	6.673±1.390	2.026±0.034	2.680±0.039	7.333±0.028	3.715±0.007
CDPNet	7.592±0.136	4.478±0.049	4.998±0.065	5.432±0.006	1.979±0.000	2.590±0.047	<u>7.029</u>±0.018	3.638±0.001
STELLA	26.347±3.948	9.844±1.123	9.345±1.033	21.410±2.085	2.411±0.044	3.541±0.179	8.371±0.021	3.835±0.003
S^2Transformer	7.807±0.053	4.373±0.131	5.112±0.372	6.864±0.494	1.912±0.001	2.492±0.070	7.149±0.002	3.708±0.015
Multivariate Time Series Methods								
DLinear	6.902±0.013	4.676±0.002	5.335±0.001	5.564±0.012	2.080±0.000	2.448±0.000	9.180±0.005	4.114±0.004
PatchTST	6.921±0.030	4.788±0.019	5.627±0.027	5.509±0.058	1.952±0.001	2.546±0.009	8.177±0.001	4.139±0.010
Crossformer	7.288±0.027	4.172±0.022	4.998±0.036	5.061±0.173	1.900±0.025	2.523±0.030	7.774±0.046	<u>3.609</u>±0.017
TimeXer	<u>6.388</u>±0.134	4.348±0.033	5.338±0.015	5.390±0.321	1.947±0.010	2.447±0.014	7.855±0.029	3.872±0.005
CCM	6.491±0.019	4.159±0.011	5.146±0.001	4.958±0.032	1.922±0.002	<u>2.434</u>±0.002	8.161±0.006	3.981±0.002
DUET	6.833±0.138	4.507±0.039	<u>4.997</u>±0.097	5.882±0.020	<u>1.895</u>±0.022	2.859±0.058	7.699±0.098	3.708±0.070
TimeFilter	6.760±0.078	<u>4.151</u>±0.022	<u>4.997</u>±0.018	5.041±0.065	<u>1.895</u>±0.023	2.435±0.045	7.848±0.024	3.650±0.072
TQNet	6.613±0.005	4.317±0.017	5.172±0.051	4.978±0.024	1.920±0.002	2.469±0.006	7.734±0.011	3.812±0.002
xPatch	7.010±0.008	4.517±0.004	5.334±0.011	5.227±0.021	1.924±0.001	2.516±0.002	8.380±0.007	3.930±0.003
STCFormer	**6.291**±0.090	**4.045**±0.028	**4.883**±0.013	**4.743**±0.149	**1.832**±0.001	**2.402**±0.003	**6.917**±0.002	**3.584**±0.002

Implementation Details. All experiments use PyTorch and BasicTS Shao et al. (2025) on an NVIDIA A100 GPU (40 GB). We use a 48-hour look-back window to forecast the next 24 and 72 hours, with MSE as the training loss. Models are trained with Adam for up to 50 epochs, with an early-stopping patience of 5. We repeat each experiment three times and report mean and standard deviation. The main text presents 24-hour MSE; other metrics and 72-hour results appear in the appendix. Full training and model configurations are given in Appendix F.3.

3.2 Main Results

Table 1 reports 24-hour MSE on eight dataset–variable tasks. The corresponding 24-hour MAE and SEDI, together with all 72-hour metrics, are reported in Appendix G. STCFormer achieves the lowest mean 24-hour MSE on all eight tasks, consistently outperforming both spatio-temporal and multivariate forecasting methods. Relative to the second-best mean in each task, it reduces MSE by 2.2% on average. The additional results in Appendix G further demonstrate this consistency.

Across all 48 dataset–variable–horizon–metric combinations, STCFormer ranks first in 42 and second in five. It also outperforms the clustering-based methods DUET and CCM and the dynamic interaction-selection method TimeFilter on every task in the main table. We attribute this advantage to locally adaptive clustering, which tracks changing station relationships, together with global attention over regional summaries, which restores information exchange across clusters. Overall, the results support a stable predictive advantage across heterogeneous station networks.

3.3 Ablation Studies

We evaluate six variants: one grouping shared across the observation window (**Window-shared**), full spatial attention instead of CGAB (**Full Attention**), fixed geographic grouping (**Static Clustering**), and removal of **InfoLoss**, **global attention**, or **local attention**. Table 2 reports 24-hour MSE; the corresponding MAE and SEDI results appear in Appendix H.

The full model achieves the lowest mean MSE on all eight tasks. **(1)** Sharing one grouping across the window increases MSE on all tasks, supporting grouping updates within the window. Averaged over the eight tasks, Window-shared raises MSE by 2.9%, whereas replacing CGAB with full attention

Table 2: 24-hour ablation results (MSE, ↓) with a 48-hour look-back window. Values are mean ± standard deviation over three runs. Bold means indicate the best reported values in each column; ties are both bold.

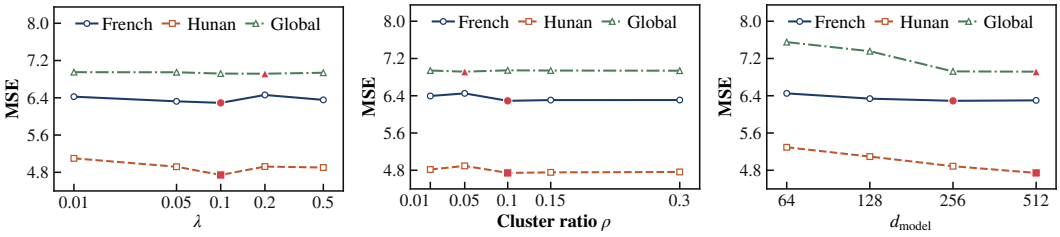
Variant	French			Hunan			Global	
	Temp.	U-Wind	V-Wind	Temp.	U-Wind	V-Wind	Temp.	Wind
Window-shared	6.516±0.093	4.181±0.041	4.996±0.081	4.891±0.078	1.913±0.024	2.428±0.032	6.954±0.027	3.758±0.012
Full Attention	6.334±0.059	4.197±0.118	5.044±0.183	4.930±0.118	1.932±0.004	2.518±0.012	6.976±0.019	3.787±0.004
Static Clustering	6.503±0.175	4.204±0.062	5.079±0.130	4.904±0.401	1.905±0.013	2.437±0.018	6.932±0.016	3.796±0.006
w/o InfoLoss	6.634±0.310	4.269±0.044	4.891±0.027	5.119±0.018	1.902±0.017	2.439±0.025	6.930±0.006	3.765±0.008
w/o Global Attn.	6.363±0.065	4.193±0.019	4.926±0.048	5.185±0.083	1.902±0.002	2.421±0.007	6.966±0.012	3.760±0.003
w/o Local Attn.	6.388±0.041	4.260±0.085	4.931±0.016	4.874±0.210	1.907±0.000	2.497±0.022	7.159±0.011	3.811±0.013
STCFormer	**6.291**±0.090	**4.045**±0.028	**4.883**±0.013	**4.743**±0.149	**1.832**±0.001	**2.402**±0.003	**6.917**±0.002	**3.584**±0.002

raises it by 3.6%. The latter result indicates that selective, structured interaction is beneficial beyond merely reducing attention cost. **(2)** Fixed geographic grouping produces a 3.3% average increase, showing that geographic proximity alone cannot capture the input-dependent relationships exposed by evolving weather conditions. Removing the within-patch InfoLoss regularizer also increases MSE, with task-dependent effects: Hunan temperature rises from 4.743 to 5.119, while Global temperature changes only from 6.917 to 6.930. This variation is consistent with InfoLoss acting most strongly when the task benefits from clearer assignment structure. **(3)** Removing either attention branch worsens all tasks. On Global wind, MSE rises from 3.584 to 3.760 without global attention and 3.811 without local attention, supporting their complementary roles. Together, these comparisons isolate the contributions of assignment adaptivity, regularized grouping, and cross-region information exchange rather than attributing the gain to a single component.

3.4 Model Analysis

3.4.1 Hyperparameter Sensitivity

We evaluate the InfoLoss weight λ, cluster ratio ρ, and embedding dimension d_{model} on the French, Hunan, and Global temperature tasks using 24-hour MSE. The cluster count is $K \approx \rho N'$, where N' denotes the station tokens after spatial patching. Look-back and patch-size studies are provided in Appendix I.3.



(a) MSE w.r.t. different λ. (b) MSE w.r.t. different ρ. (c) MSE w.r.t. different d_{model}.

Figure 3: Hyperparameter sensitivity on three temperature tasks (24-hour MSE). Red markers indicate the lowest reported MSE within each dataset's sweep.

Figure 3 shows moderate sensitivity across all three settings. The best λ is 0.1 on French and Hunan and 0.2 on Global, while larger weights bring no further gain. Intermediate cluster ratios are preferred: $\rho = 0.1$ performs best on French and Hunan, and 0.05 on Global, with nearby settings remaining competitive. French favors $d_{\text{model}} = 256$, whereas Hunan and Global favor 512; however, gains beyond 256 are small. Overall, STCFormer remains stable over practical ranges and does not require uniformly large regularization, cluster counts, or feature dimensions.

3.4.2 Efficiency Analysis

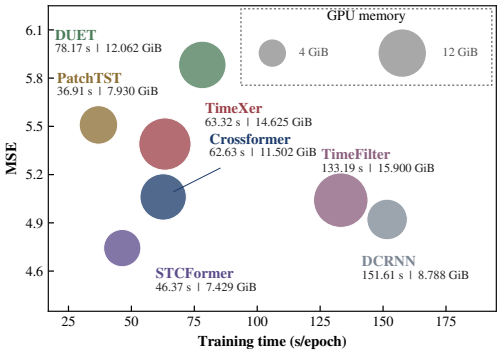


Figure 4: Efficiency on Hunan temperature (48-to-24-hour forecasting, batch size 8).

Figure 4 compares accuracy, mean training time per epoch, and peak memory on Hunan. Among the methods shown, STCFormer achieves the lowest reported MSE (4.743) and memory usage (7.429 GiB), taking 46.372 ± 0.095 s/epoch. Compared with Crossformer, it reduces training time by 26.0% and memory by 35.4%; it is $2.87\times$ faster than TimeFilter. PatchTST trains faster (36.913 s/epoch), but has higher MSE (5.509) and memory usage (7.930 GiB). The bubble sizes make clear that STCFormer's accuracy gain does not come with a larger memory footprint. This result is consistent with CGAB's selective local interactions and compact regional summaries, which retain cross-cluster information without dense station-wise attention. STCFormer thus provides a favorable accuracy–resource trade-off on the larger station network. Appendix I.4 reports the French comparison.

3.4.3 Dynamic Clustering Case Study

To test whether learned groups track within-window relationship changes, we evaluate all test windows from the French-temperature 48-to-24-hour forecasting task. For each window, we measure within-cluster evolution coherence using Pearson correlations between hourly temperature increments for valid co-clustered station pairs and average the scores over local temporal blocks. Higher values indicate that stations grouped together exhibit more consistent short-term evolution.

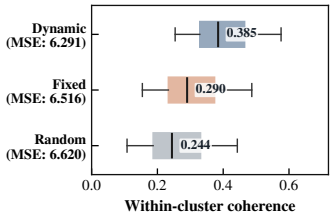


Figure 5: Dynamic, fixed, and random grouping on all French-temperature test windows ($48 \to 24$). Parentheses report MSE.

Figure 5 compares dynamic grouping with a fixed window-shared partition and size-matched random grouping. Dynamic grouping achieves the highest median coherence (0.385), compared with Fixed (0.290) and Random (0.244), while also obtaining the lowest forecasting MSE (6.291 versus 6.516 and 6.620). Fixed remains better than Random, indicating persistent station dependence; the additional gain from Dynamic shows that updating assignments within a window captures transient couplings that a single shared partition obscures.

4 Conclusion

We introduced STCFormer to model locally changing station dependencies while preserving global context for weather forecasting. Evolution-driven grouping adapts spatial interactions at each temporal patch, and CGAB combines fine-grained local attention within clusters and global attention over regional state summaries. Our theoretical analysis shows that the derived Lipschitz upper bound for fixed-partition local attention is no larger than its dense counterpart. Its connection to station–cluster mutual information motivates InfoLoss. Across three real-world datasets, STCFormer achieves the lowest 24-hour MSE on all eight tasks and ranks first or second in 47 of 48 metric–task–horizon comparisons. Ablations and case studies support the complementary roles of local grouping updates, global context, and clustering regularization. These findings support locally adaptive, selective spatial modeling for station-based weather forecasting.

AI use statement

Generative AI assisted the authors with preliminary exploration, literature review, and manuscript polishing and editing. The authors remain fully responsible for the accuracy of the text, the cited references, and the reported research.

REFERENCES

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Syndar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=gerNCVqqtR.

A Baklanov, C Sue B Grimmond, D Carlson, D Terblanche, X Tang, V Bouchet, B Lee, G Langendijk, R Kumar Kolli, and A Hovsepyan. From urban meteorology, climate and environment research to integrated city services. *Urban Climate*, 23:330–341, 2018.

Valérie Castin, Pierre Ablin, and Gabriel Peyré. How smooth is attention? In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 5817–5840. PMLR, 2024. URL https://proceedings.mlr.press/v235/castin24a.html.

Hongyi Chen, Xiucheng Li, Xinyang Chen, Yun Cheng, Jing Li, Kehai Chen, and Liqiang Nie. S^2Transformer: Scalable structured transformers for global station weather forecasting. *Transactions on Machine Learning Research*, 2026. URL https://openreview.net/forum?id=AL2VnKno5n.

Jialin Chen, Jan Eric Lenssen, Aosong Feng, Weihua Hu, Matthias Fey, Leandros Tassiulas, Jure Leskovec, and Rex Ying. From similarity to superiority: Channel clustering for time series forecasting. In *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

Abhimanyu Das, Weihao Kong, Andrew Leach, Shaan Mathur, Rajat Sen, and Rose Yu. Long-term forecasting with tide: Time-series dense encoder. *Trans. Mach. Learn. Res.*, 2023, 2023.

John W Freebairn and John W Zillman. Economic benefits of meteorological services. *Meteorological Applications*, 9(1):33–44, 2002.

Yisong Fu, Fei Wang, Zezhi Shao, Boyu Diao, Lin Wu, Zhulin An, Chengqing Yu, Yujie Li, and Yongjun Xu. On the integration of spatial-temporal knowledge: A lightweight approach to atmospheric time series forecasting. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://papers.nips.cc/paper_files/paper/2025/file/7691484a7a35d5e2742279c1d926b778-Paper-Conference.pdf.

Zhangyang Gao, Cheng Tan, Lirong Wu, and Stan Z. Li. Simvp: Simpler yet better video prediction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 3160–3170. IEEE, 2022.

Ryan Gomes, Andreas Krause, and Pietro Perona. Discriminative clustering by regularized information maximization. In *Advances in Neural Information Processing Systems*, volume 23, 2010. URL https://proceedings.neurips.cc/paper_files/paper/2010/file/42998cf32d552343bc8e460416382dca-Paper.pdf.

Vincent Le Guen and Nicolas Thome. Disentangling physical dynamics from unknown factors for unsupervised video prediction. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 11471–11481. Computer Vision Foundation / IEEE, 2020.

Pradeep Hewage, Ardhendu Behera, Marcello Trovati, Ella Pereira, Morteza Ghahremani, Francesco Palmieri, and Yonghuai Liu. Temporal convolutional neural (tcn) network for an effective weather forecasting using time-series data from the local weather station. *Soft Computing*, 24(21):16453–16482, 2020.

Yifan Hu, Guibin Zhang, Peiyuan Liu, Disen Lan, Naiqi Li, Dawei Cheng, Tao Dai, Shu-Tao Xia, and Shirui Pan. Timefilter: Patch-specific spatial-temporal graph filtration for time series forecasting. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, 2025.

Qihe Huang, Lei Shen, Ruixin Zhang, Jiahuan Cheng, Shouhong Ding, Zhengyang Zhou, and Yang Wang. Hdmixer: Hierarchical dependency with extendable patch for multivariate time series forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(11):12608–12616, 2024. doi: 10.1609/aaai.v38i11.29155.

Iris A. M. Huijben, Wouter Kool, Max B. Paulus, and Ruud J. G. van Sloun. A review of the gumbel-max trick and its extensions for discrete stochasticity in machine learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(2):1353–1371, 2023.

Renhe Jiang, Zhaonan Wang, Jiawei Yong, Puneet Jeph, Quanjun Chen, Yasumasa Kobayashi, Xuan Song, Shintaro Fukushima, and Toyotaro Suzumura. Spatio-temporal meta-graph learning for traffic 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. 8078–8086, 2023.

Do-Yeon Kim and Heung-Il Suk. SIGMAformer: a spatiotemporal gaussian mixture correlation transformer for global weather forecasting. *npj Climate and Atmospheric Science*, 9:113, 2026. doi: 10.1038/s41612-026-01385-w. URL https://www.nature.com/articles/s41612-026-01385-w.

Jeffrey K Lazo, Rebecca E Morss, and Julie L Demuth. 300 billion served: Sources, perceptions, uses, and values of weather forecasts. *Bulletin of the American Meteorological Society*, 90(6): 785–798, 2009.

Rongwen Li, Haiyang Hu, Liang Du, Jiarong Chen, Bingbing Jiang, and Peng Zhou. One-stage fair multi-view spectral clustering. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 1407–1416, 2024.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

Shengsheng Lin, Haojun Chen, Haijie Wu, Chunyun Qiu, and Weiwei Lin. Temporal query network for efficient multivariate time series forecasting. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 37797–37814. PMLR, 2025. URL https://proceedings.mlr.press/v267/lin25e.html.

Dachuan Liu, Jin Wang, Shuo Shang, and Peng Han. MSDR: multi-step dependency relation networks for spatial temporal forecasting. In *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, pp. 1042–1050, 2022. doi: 10.1145/3534678.3539397.

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*, 2024a.

Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer: Generative pre-trained transformers are large time series models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 32369–32399, 2024b. URL https://proceedings.mlr.press/v235/liu24cb.html.

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.

Xiangfei Qiu, Xingjian Wu, Yan Lin, Chenjuan Guo, Jilin Hu, and Bin Yang. DUET: dual clustering enhanced multivariate time series forecasting. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining, V.1, KDD 2025, Toronto, ON, Canada, August 3-7, 2025*, pp. 1185–1196, 2025.

David Rolnick, Priya L Donti, Lynn H Kaack, Kelly Kochanski, Alexandre Lacoste, Kris Sankaran, Andrew Slavin Ross, Nikola Milojevic-Dupont, Natasha Jaques, Anna Waldman-Brown, et al. Tackling climate change with machine learning. *ACM Computing Surveys (CSUR)*, 55(2):1–96, 2022.

Rajat Sen, Hsiang-Fu Yu, and Inderjit S. Dhillon. Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 4838–4847, 2019.

Zezhi Shao, Fei Wang, Yongjun Xu, Wei Wei, Chengqing Yu, Zhao Zhang, Di Yao, Tao Sun, Guangyin Jin, Xin Cao, Gao Cong, Christian S. Jensen, and Xueqi Cheng. Exploring progress in multivariate time series forecasting: Comprehensive benchmarking and heterogeneity analysis. *IEEE Trans. Knowl. Data Eng.*, 37(1):291–305, 2025.

Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. Time-MoE: Billion-scale time series foundation models with mixture of experts. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/558d48c1f08675daa636e09bfe94a89e-Abstract-Conference.html.

Artyom Stitsyuk and Jaesik Choi. xPatch: Dual-stream time series forecasting with exponential seasonal-trend decomposition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 20601–20609, 2025. URL https://github.com/stitsyuk/xPatch.

Jiabin Tang, Wei Wei, Lianghao Xia, and Chao Huang. Easyst: A simple framework for spatio-temporal prediction. 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. 2220–2229, 2024.

Yunbo Wang, Haixu Wu, Jianjin Zhang, Zhifeng Gao, Jianmin Wang, Philip S. Yu, and Mingsheng Long. Predrnn: A recurrent neural network for spatiotemporal predictive learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(2):2208–2225, 2023.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Guo Qin, Haoran Zhang, Yong Liu, Yunzhong Qiu, Jianmin Wang, and Mingsheng Long. Timexer: Empowering transformers for time series forecasting with exogenous variables. In *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

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, Hang Zhou, Mingsheng Long, and Jianmin Wang. Interpretable weather forecasting for worldwide stations with a unified deep model. *Nat. Mac. Intell.*, 5(6):602–611, 2023.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pp. 1907–1913, 2019.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pp. 753–763, 2020.

Chujie Xu, Yuqing Ma, Haoyuan Deng, Yajun Gao, Yudie Wang, Kai Lv, and Xianglong Liu. Continuous diffusive prediction network for multi-station weather prediction. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2025, Montreal, Canada, August 16-22, 2025*, pp. 6714–6722, 2025. doi: 10.24963/IJCAI.2025/747.

Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pp. 3634–3640, 2018.

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.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, 2023.

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.

Peng Zhou, Rongwen Li, and Liang Du. Fair kernel k-means: from single kernel to multiple kernel. In *Advances in Neural Information Processing Systems*, pp. 99686–99714, 2024.

A Related Work

Spatio-Temporal Forecasting: ST methods combine temporal modeling with spatial structures, learned relationships, or domain knowledge. Graph-based approaches represent locations as nodes and dependencies as edges Jiang et al. (2023); Liu et al. (2022). DCRNN Li et al. (2018) uses diffusion convolution and recurrent modeling, while STGCN Yu et al. (2018) combines graph convolutions with gated temporal convolutions. Graph WaveNet Wu et al. (2019) and MTGNN Wu et al. (2020) learn adjacency structures from data, although a graph learned during training need not adapt within each observation window. Weather models further incorporate domain structure: Corrformer Wu et al. (2023) organizes correlations hierarchically, and STELLA Fu et al. (2025) integrates spatial-temporal knowledge into a lightweight model. SIGMAformer Kim & Suk (2026) combines Gaussian-mixture pattern extraction with dynamic spatio-temporal correlation modeling for global station weather forecasting. Our focus is on grouping stations from local evolution at each temporal patch while retaining information from all groups.

Multivariate Time Series Forecasting: MTS methods treat station sequences as channels. Channel-dependent methods model series jointly, whereas channel-independent methods process them separately with shared parameters. Following recurrent and convolutional approaches Hewage et al. (2020); Sen et al. (2019), Informer Zhou et al. (2021) and Autoformer Wu et al. (2021) improve temporal modeling through sparse attention and decomposition. iTransformer Liu et al. (2024a) attends over variate tokens to capture cross-channel dependencies. Channel-independent models avoid unnecessary mixing Das et al. (2023); Nie et al. (2023): DLinear Zeng et al. (2023) uses decomposition and linear mappings, while PatchTST Nie et al. (2023) applies a shared Transformer to temporal patches. Independent processing, however, does not explicitly exploit other stations' predictive information.

Clustering balances shared modeling and channel specialization. CCM Chen et al. (2024) groups similar channels and learns cluster-specific shared parameters. DUET Qiu et al. (2025) combines temporal distribution clustering with frequency-based channel soft clustering. These mechanisms differ from fully isolated hard partitions. TimeFilter Hu et al. (2025) further adapts dependencies through patch-specific spatio-temporal graph filtration. Across these designs, grouping granularity and information coverage matter: window-shared assignments may obscure local relationship changes, while exclusively intra-group interactions can omit useful global context. STCFormer combines locally updated groups with attention over all regional content centers.

B Efficient Implementation of Intra-Cluster Local Attention

Equation (8) describes the hard-partition attention pattern. Our experiments evaluate it with packed local attention, avoiding an explicit $N' \times N'$ mask. Assignments are recomputed at every temporal patch and encoder layer.

Hard routing with a differentiable membership bias. Training obtains straight-through assignments $\mathbf{H}_{\mathrm{ST}}$ using `F.gumbel_softmax(log(R), hard=True)`. Let κ_s be the selected cluster index and $\mathcal{I}_k = \{s : \kappa_s = k\}$. The indices and their sorting are nondifferentiable. Crucially, we retain the selected membership $a_s = (H_{\mathrm{ST}})_{s,\kappa_s}$ as a differentiable tensor. For group k, gather the state features $\mathbf{X}_k = \mathbf{X}^{\mathrm{att}}_{\mathcal{I}_k,:}$ and the membership vector $\mathbf{a}_k = (a_s)_{s\in\mathcal{I}_k}$. Packed attention computes

$$\mathbf{Y}_k = \mathbf{X}_k + \mathrm{Softmax}\left(\frac{(\mathbf{X}_k\mathbf{W}_Q^{\mathrm{loc}})(\mathbf{X}_k\mathbf{W}_K^{\mathrm{loc}})^\top}{\sqrt{D}} + \mathbf{1}_{n_k}(\log \mathbf{a}_k)^\top\right)(\mathbf{X}_k\mathbf{W}_V^{\mathrm{loc}}). \tag{18}$$

The logarithmic membership is a key bias, broadcast across queries. Selected memberships equal one in the hard forward pass, so this bias is zero and each token attends to exactly its hard cluster. In the backward pass, the membership bias propagates gradients through the continuous Gumbel–Softmax relaxation to $\mathbf{R}$, despite nondifferentiable group indexing. At inference, routing uses deterministic $\arg\max_k R_{s,k}$.

C Computational Complexity Analysis

We analyze the current pipeline: patch embedding, dual-view temporal representation, dynamic clustering, intra-cluster attention, station-to-region attention, and a direct prediction head. Let $n = N' = \lceil N/L_s \rceil$ and $m = T' = \lceil T/L_t \rceil$, where N, T are the original station count and history length. Let D be the hidden width and $K \leq n$ the number of clusters. Clustering uses one E–M update followed by a final E-step. Costs below are per sample and per block, with a fixed number of heads and feed-forward widths proportional to D. For a batch, multiply operation counts by the batch size.

C.1 Module-wise computation

Patch embedding and dual views. Flattening and linearly projecting the value patches costs $\mathcal{O}(mnL_tL_sCD)$, where C is the number of observed variables; geographic projection and positional addition contribute separate lightweight embedding operations. The state view applies temporal attention to m tokens for each of the n spatial patches, costing $\mathcal{O}(nm^2D)$. Temporal differences cost $\mathcal{O}(mnD)$, and the evolution MLP, attention projections, and feed-forward updates cost $\mathcal{O}(mnD^2)$ for fixed expansion ratios. The evolution view introduces no additional all-pairs temporal attention.

Dynamic station clustering. At each temporal patch, the initial E-step computes $\mathbf{X}^{\mathrm{clu}}\mathbf{P}^\top$, the M-step aggregates $\widetilde{\mathbf{R}}^\top\mathbf{X}^{\mathrm{clu}}$ and blends the result with $\mathbf{P}$, and the final E-step computes $\mathbf{X}^{\mathrm{clu}}\mathbf{C}^\top$. Each matrix product costs $\mathcal{O}(nKD)$; prototype mixing and normalization are lower-order operations. Thus, the fixed E–M–E sequence over the window costs

$$\mathcal{C}_{\mathrm{clu}} = \mathcal{O}(mnKD). \tag{19}$$

Sampling or taking the row-wise argmax of the final assignments adds $\mathcal{O}(mnK)$. No station-to-station distance matrix or reuse of the previous patch's partition is required.

Intra-cluster attention. Let $n_{p,k} = |\mathcal{I}_{p,k}|$ and $S_p = \sum_{k=1}^{K} n_{p,k}^2$. The packed evaluation in Appendix B has attention-product cost

$$\mathcal{C}_{\mathrm{loc}} = \mathcal{O}\left(D\sum_{p=1}^{m} S_p\right), \qquad \frac{n^2}{K} \leq S_p \leq n^2. \tag{20}$$

The lower bound follows from Cauchy–Schwarz and is attained by equal group sizes when divisibility permits. Approximately balanced groups give $S_p \approx n^2/K$; collapse into one group gives $S_p = n^2$. InfoLoss encourages balanced soft masses but does not enforce equal hard group sizes.

Station-to-region global attention. Soft-assignment-weighted aggregation $\mathbf{R}^\top\mathbf{X}^{\mathrm{loc}}$ costs $\mathcal{O}(mnKD)$; computing and normalizing the regional soft masses costs $\mathcal{O}(mnK + mKD)$. Cross-attention between n station queries and K regional content centers costs $\mathcal{O}(mnKD)$, with an additional $\mathcal{O}(m(n+K)D^2)$ for projections. These are station-to-region interactions, rather than a $K \times K$ centroid self-attention stage. The centroids used in the E–M updates and the content centers used as attention keys and values are computed separately.

InfoLoss during training. At each temporal patch, computing $\boldsymbol{\pi}_p = n^{-1}\mathbf{R}_p^\top\mathbf{1}$ and the sum of row entropies takes $\mathcal{O}(nK)$ operations. The marginal entropy adds $\mathcal{O}(K)$, giving

$$\mathcal{C}_{\mathrm{info}} = \mathcal{O}(mnK). \tag{21}$$

No $K \times K$ joint matrix or adjacent-patch pairing is required. The loss averages over temporal patches within each layer and sums across layers; these reductions do not change the asymptotic cost. This term is absent at inference. Backpropagation through the differentiable clustering updates has the same asymptotic arithmetic order as their forward operations for the fixed E–M–E sequence; retained intermediate activations and the chosen assignment surrogate affect training memory.

Direct prediction head. For one target variable, the head in Eq. (11) flattens the $m \times D$ encoded features of each spatial patch. Layer normalization costs $\mathcal{O}(nmD)$, and the two projections cost $\mathcal{O}(nmDD_h + nD_hOL_s)$, where D_h is the head's hidden width and O the forecasting horizon. Reshaping, removing padding, and restoring station order add no learned pairwise interaction.

C.2 Overall cost and memory

Combining the representation and interaction modules gives

$$\mathcal{C}_{\text{block}} = \mathcal{O}\left(mnD^2 + nm^2D + mnKD + D\sum_{p=1}^{m} S_p \right). \tag{22}$$

With approximately balanced groups, the last term becomes mn^2D/K; in the worst case it becomes mn^2D. Clustering and global attention remain in both expressions and cannot be discarded merely because $K < n$. For B blocks with unchanged dimensions, the backbone cost is $B\mathcal{C}_{\text{block}}$. If dimensions differ across blocks, their costs must instead be summed explicitly. The full model additionally incurs patch embedding and direct-head costs; training adds InfoLoss wherever it is applied. No layer-wise temporal shrinking or autoregressive decoder is assumed.

Ignoring widths and rounding, the principal terms are

$$\underbrace{\frac{NT^2}{L_sL_t^2}}_{\text{temporal attention}}, \qquad \underbrace{\frac{TN^2}{L_tL_s^2K}}_{\text{balanced local attention}}, \qquad \underbrace{\frac{TNK}{L_tL_s}}_{\text{clustering and global attention}}. \tag{23}$$

Increasing K reduces balanced local-attention work but increases clustering, global-attention, and InfoLoss costs. In particular, setting K proportional to n does not make the full block linear in n: clustering and global attention remain quadratic, while within-patch InfoLoss costs $\mathcal{O}(mnK)$ and is quadratic in n at a fixed cluster ratio. Patching reduces the token grid, but these trade-offs and hardware utilization determine practical efficiency.

For conventional materialized attention, score storage is $\mathcal{O}(nm^2 + \sum_p S_p + mnK)$, while token features and assignments require $\mathcal{O}(mnD + mnK)$. The two E-steps and single M-step retain only a fixed number of clustering intermediates during training. Fused attention kernels can avoid storing the full score matrices without changing the arithmetic orders above. InfoLoss reduces the existing soft assignments into m cluster-marginal vectors and row-entropy sums. These summaries require $\mathcal{O}(mK + mn)$ storage if materialized together; a conventional elementwise entropy implementation may additionally retain $\mathcal{O}(mnK)$ intermediates for backward computation. Sequential or fused reductions can reduce forward workspace, with backward storage depending on checkpointing. These are activation and workspace costs, excluding model parameters and optimizer states.

C.3 Comparison with current baselines

Table 3 compares representative dependency-modeling operators from the baselines in our experiments. Hidden widths, head counts, and depth are fixed; each expression covers the operator named in the table, not end-to-end training or forecasting. Embedding, prediction heads, and auxiliary objectives are excluded. This scope makes explicit the distinction between dense station attention, temporal patch attention, compressed global interaction, and dynamically grouped interaction.

DCRNN's sparse diffusion cost depends on graph edges rather than necessarily on N^2 (Li et al., 2018). Corrformer's multi-correlation uses a spatial tree and temporal correlation to obtain linear spatial and log-linear temporal scaling (Wu et al., 2023). PatchTST attends independently along each channel's temporal patches (Nie et al., 2023), whereas Crossformer additionally uses r routers per temporal segment for cross-channel communication (Zhang & Yan, 2023). CCM's assignment and prototype cross-attention cost $\mathcal{O}(NK_{\text{CCM}})$ at fixed hidden width (Chen et al., 2024). We use DLinear as its backbone; the displayed term excludes that backbone and the cluster-aware prediction heads, and is not an end-to-end cost.

DUET's displayed $\mathcal{O}(N^2)$ term concerns fusion after temporal extraction and channel-mask construction; its frequency transform and learned distance metric incur additional costs (Qiu et al., 2025). TimeFilter constructs pairwise scores over NP_t patch tokens before neighbor selection and filtration, so a sparse final graph alone does not remove this construction cost (Hu et al., 2025). For STCFormer, substitute $\sum_p S_p \approx mn^2/K$ for balanced groups or $\sum_p S_p \le mn^2$ for the worst case. The matched runtime and memory measurements in Sections 3.4.2 and I.4 complement these asymptotic comparisons and show STCFormer's favorable accuracy–resource trade-off.

Table 3: Computational complexity of representative dependency-modeling operators among the current baselines. N, T denote stations and history steps. P_t is each baseline's actual temporal-token count (accounting for its patch length and stride), r is Crossformer's router count, K_{CCM} is CCM's cluster count, and $q, |\mathcal{E}|$ are DCRNN's diffusion order and graph-edge count. For STCFormer, $n = \lceil N/L_s \rceil$, $m = \lceil T/L_t \rceil$, with spatial and temporal patch sizes L_s, L_t. Widths are fixed; the stated scope excludes embeddings, output heads, and training losses.

Model	Operator compared	Time complexity
DCRNN	Diffusion recurrence over T steps	$\mathcal{O}(Tq(N + \lvert\mathcal{E}\rvert))$
Corrformer	Multi-correlation	$\mathcal{O}(NT \log T)$
PatchTST	Per-channel temporal attention	$\mathcal{O}(NP_t^2)$
Crossformer	Temporal attention and routers	$\mathcal{O}(NP_t^2 + NP_t r)$
CCM (DLinear)	Cluster assignment and prototype cross-attention	$\mathcal{O}(NK_{\mathrm{CCM}})$
DUET	Masked channel-attention fusion	$\mathcal{O}(N^2)$
TimeFilter	Dense patch-pair graph construction	$\mathcal{O}(N^2 P_t^2)$
STCFormer	Dual views, clustering, local–global attention	$\mathcal{O}(nm^2 + mnK + \sum_p S_p)$

D Proofs for Cluster-Conditioned Local Attention

D.1 Notation and Preliminaries

We analyze the single-head residual local-attention stage in Eq. (8), equivalently evaluated by the packed implementation in Appendix B. The evolution view determines the partition and the state view supplies the attention content. Holding the realized partition and projection weights fixed isolates how restricted interaction changes the sensitivity bound relative to dense attention.

Feature perturbations use the Frobenius norm; weights use the spectral norm. The convex domain is $\Omega_{R_X} = \{U \in \mathbb{R}^{n\times D} : \max_s \|u_s\|_2 \le R_X\}$. For a mapping f, define

$$\mathrm{Lip}_F(f) = \sup_{\substack{U,\widetilde{U}\in\Omega_{R_X}\\ U\neq\widetilde{U}}} \frac{\|f(U) - f(\widetilde{U})\|_F}{\|U - \widetilde{U}\|_F}. \tag{24}$$

Integration of the derivative along the segment joining two inputs gives

$$\mathrm{Lip}_F(f) \le \sup_{U\in\Omega_{R_X}} \|\mathrm{D}f(U)\|_{F\to F}, \qquad \|\mathrm{D}f(U)\|_{F\to F} = \sup_{E\neq 0} \frac{\|\mathrm{D}f(U)[E]\|_F}{\|E\|_F}, \tag{25}$$

for mappings smooth on a neighborhood of the domain. Also, $\mathrm{Lip}_F(\mathrm{Id} + f) \le 1 + \mathrm{Lip}_F(f)$. The hard assignment matrix H has one nonzero entry per row; empty clusters are permitted and produce no output rows. We use natural logarithms, $0 \log 0 = 0$, and $1 \le K \le n$.

D.2 Proof of Theorem 2.1

Write the residual operator and its attention part as

$$\begin{aligned} \mathcal{A}_H(U) = U + \Phi_H(U), \qquad \Phi_H(U) = \mathrm{Softmax}\big(UAU^\top + \log(HH^\top)\big)\, UW_V^{\mathrm{loc}}, \\ A = \frac{W_Q^{\mathrm{loc}}(W_K^{\mathrm{loc}})^\top}{\sqrt{D}}, \qquad v = \|W_V^{\mathrm{loc}}\|_2, \qquad \beta = R_X^2\|A\|_2. \end{aligned} \tag{26}$$

Here $\log 0 = -\infty$ implements the fixed mask. Every attention row contains its self-entry. Dense attention corresponds to a single group containing all n tokens.

Proof. **Step 1: a single attention block.** Let $\mathcal{I}$ be a nonempty cluster of size q. Write token rows as columns $a_i = u_i^\top$ and set $V = (W_V^{\mathrm{loc}})^\top$. Its probabilities, weighted mean, and output are

$$p_{ij} = \frac{\exp(a_i^\top A a_j)}{\sum_{\ell\in\mathcal{I}} \exp(a_i^\top A a_\ell)}, \qquad \mu_i = \sum_{j\in\mathcal{I}} p_{ij} a_j, \qquad g_i = V\mu_i. \tag{27}$$

The row-norm assumption implies $|a_i^\top A a_j| \le \beta$. Consequently,

$$p_{ij} \le \frac{e^{2\beta}}{q - 1 + e^{2\beta}}, \qquad \sum_{i\in\mathcal{I}} p_{ij} \le s_\beta(q) := \frac{q e^{2\beta}}{q - 1 + e^{2\beta}} \le \min\{q, e^{2\beta}\}. \tag{28}$$

The probability bound follows by placing its numerator at the largest admissible logit and the other $q-1$ logits at the smallest admissible value. It sharpens the generic column-sum bound q by accounting for bounded logits, consistent with the bounded-domain perspective of Castin et al. (2024).

For a perturbation with token columns e_i, differentiating the bilinear logits and softmax gives

$$\begin{aligned} \mathrm{D}g_i[E] &= V\big(T_i^{\mathrm{val}} + T_i^{\mathrm{qry}} + T_i^{\mathrm{key}}\big), \\ T_i^{\mathrm{val}} &= \sum_j p_{ij} e_j, \qquad T_i^{\mathrm{qry}} = C_i A^\top e_i, \\ T_i^{\mathrm{key}} &= \sum_j p_{ij}(a_i^\top A e_j)(a_j - \mu_i), \qquad C_i = \sum_j p_{ij}(a_j - \mu_i)(a_j - \mu_i)^\top. \end{aligned} \tag{29}$$

All sums in this step are over $\mathcal{I}$. The covariance matrix satisfies

$$\|C_i\|_2 \le \mathrm{tr}(C_i) = \sum_j p_{ij}\|a_j\|_2^2 - \|\mu_i\|_2^2 \le R_X^2. \tag{30}$$

Let $T^{\mathrm{val}}, T^{\mathrm{qry}}, T^{\mathrm{key}}$ collect the corresponding row contributions. Jensen's inequality, Eq. (28), and weighted Cauchy–Schwarz imply

$$\begin{aligned} \|T^{\mathrm{val}}\|_F^2 &\le \sum_{i,j} p_{ij}\|e_j\|_2^2 \le s_\beta(q)\|E_{\mathcal{I}}\|_F^2, \\ \|T^{\mathrm{qry}}\|_F^2 &\le R_X^4\|A\|_2^2 \sum_i \|e_i\|_2^2 = \beta^2\|E_{\mathcal{I}}\|_F^2, \\ \|T^{\mathrm{key}}\|_F^2 &\le \sum_i \left(\sum_j p_{ij}(a_i^\top A e_j)^2\right)\left(\sum_j p_{ij}\|a_j - \mu_i\|_2^2\right) \\ &\le R_X^4\|A\|_2^2 \sum_{i,j} p_{ij}\|e_j\|_2^2 \le \beta^2 s_\beta(q)\|E_{\mathcal{I}}\|_F^2. \end{aligned} \tag{31}$$

In the key term, $|a_i^\top A e_j| \le R_X\|A\|_2\|e_j\|_2$ and the second factor is at most R_X^2. Thus the non-residual block obeys

$$\|\mathrm{D}\Phi_{\mathcal{I}}(U_{\mathcal{I}})[E_{\mathcal{I}}]\|_F \le v\left[\beta + (1+\beta)\sqrt{s_\beta(q)}\right]\|E_{\mathcal{I}}\|_F. \tag{32}$$

Equation (25) therefore gives the same bound for $\mathrm{Lip}_F(\Phi_{\mathcal{I}})$.

Step 2: disjoint groups and the dense comparison. Let $q_k = |\mathcal{I}_k|$ and $m(H) = \max_k q_k$. Reordering tokens by their fixed groups preserves Frobenius norms and makes the mapping block separable. Hence

$$\begin{aligned} \|\Phi_H(U) - \Phi_H(\widetilde{U})\|_F^2 &= \sum_k \|\Phi_{\mathcal{I}_k}(U_{\mathcal{I}_k}) - \Phi_{\mathcal{I}_k}(\widetilde{U}_{\mathcal{I}_k})\|_F^2 \\ &\le \left(\max_k \mathrm{Lip}_F(\Phi_{\mathcal{I}_k})\right)^2 \|U - \widetilde{U}\|_F^2. \end{aligned} \tag{33}$$

The maximum is over nonempty clusters. For real $q \geq 1$,

$$s'_\beta(q) = \frac{e^{2\beta}(e^{2\beta}-1)}{(q-1+e^{2\beta})^2} \geq 0. \tag{34}$$

Adding the residual and applying monotonicity gives

$$\mathrm{Lip}_F(\mathcal{A}_H) \leq 1 + v\left[\beta + (1+\beta)\sqrt{s_\beta(m(H))}\right] = B(m(H)) \leq B(n). \tag{35}$$

Applying exactly the same single-block calculation to $q = n$ establishes $\mathrm{Lip}_F(\mathcal{A}_{\mathrm{dense}}) \leq B(n)$. Defining $B_{\mathrm{ours}}(H) = B(m(H))$ and $B_{\mathrm{dense}} = B(n)$ gives Eq. (13). Throughout this appendix, $B(\cdot)$ denotes the derived upper bound rather than the exact Lipschitz constant in Eq. (24). □

For $v > 0$, $\beta > 0$, and $m(H) < n$, the comparison is strict: $B(m(H)) < B(n)$. Since $s_\beta(q) \leq q$, $1 + v[\beta + (1+\beta)\sqrt{q}]$ is also valid, while the bounded-logit expression is tighter because $s_\beta(q)$ saturates at $e^{2\beta}$. The numerical diagnostics in Appendix I.6 confirm that the predicted cluster-size dependence is visible in trained models.

D.3 Proof of Theorem 2.2

Proof. Fix a temporal patch and condition on its pre-draw soft assignments $R = (r_{s,k})$. Sample token labels independently with $\Pr(\kappa_s = k \mid R) = r_{s,k}$. This is the hard forward distribution of Gumbel-Max, $\kappa_s = \arg\max_k(\log r_{s,k} + g_{s,k})$, with fresh independent standard Gumbel noises across tokens and labels. Let $h_{s,k} = \mathbf{1}\{\kappa_s = k\}$ and $M_k = \sum_s h_{s,k}$. The marginal and conditional entropy of the variables X, Y in the main text are

$$\pi_k = \frac{1}{n}\sum_s r_{s,k}, \qquad \bar{h} = H(Y \mid X) = \frac{1}{n}\sum_s H(r_{s,:}). \tag{36}$$

Writing $d = \mathrm{KL}(\pi \| u_K)$, the information gap decomposes as

$$\varepsilon = \log K - I(X;Y) = d + \bar{h}. \tag{37}$$

Both terms are nonnegative. Pinsker's inequality with natural logarithms gives

$$\max_k \pi_k \leq \frac{1}{K} + \frac{1}{2}\|\pi - u_K\|_1 \leq \frac{1}{K} + \sqrt{\frac{d}{2}}. \tag{38}$$

Next, center the count vector by $Z_k = M_k - n\pi_k$. Independence across tokens and $1 - x \leq -\log x$ imply

$$\begin{aligned} \mathbb{E}[\|Z\|_2^2 \mid R] &= \sum_k \mathrm{Var}(M_k \mid R) = \sum_{s,k} r_{s,k}(1 - r_{s,k}) \\ &\leq -\sum_{s,k} r_{s,k} \log r_{s,k} = n\bar{h}. \end{aligned} \tag{39}$$

Zero entries are interpreted by continuity. No independence between the cluster counts is required: the squared Euclidean norm is a sum of coordinate squares without cross terms. For $\bar{h} > 0$, Markov's inequality yields

$$\Pr\left(\|Z\|_2 > \sqrt{\frac{n\bar{h}}{\delta}} \;\middle|\; R\right) \leq \delta. \tag{40}$$

If $\bar{h} = 0$, every assignment is deterministic and $Z = 0$ almost surely, so the same event bound holds. Combining this event with Eq. (38), and then applying Cauchy–Schwarz, gives

$$\begin{aligned} m(H) = \max_k(n\pi_k + Z_k) &\leq \frac{n}{K} + n\sqrt{\frac{d}{2}} + \sqrt{\frac{n\bar{h}}{\delta}} \\ &\leq \frac{n}{K} + \sqrt{\left(\frac{n^2}{2} + \frac{n}{\delta}\right)(d + \bar{h})} \\ &= \frac{n}{K} + \sqrt{\left(\frac{n^2}{2} + \frac{n}{\delta}\right)(\log K - I(X;Y))}. \end{aligned} \tag{41}$$

Using $m(H) \le n$ proves $m(H) \le M_\delta(I(X;Y))$ with conditional probability at least $1-\delta$. Since $M_\delta(I(X;Y)) \ge n/K \ge 1$, monotonicity of B and Theorem 2.1 give, on the same event,

$$\mathrm{Lip}_F(\mathcal{A}_H) \le B_{\mathrm{ours}}(H) \le B(M_\delta(I(X;Y)))\,. \tag{42}$$

No additional failure probability is needed for the sensitivity statement, because the structural theorem holds for every realized partition. □

At fixed n, K, v, β, δ, increasing $I(X;Y)$ tightens or preserves the certificate. At its maximum $I(X;Y) = \log K$, the assignments are deterministic with a uniform marginal, yielding groups of size n/K whenever divisibility permits. Thus, InfoLoss directly favors the assignment regime associated with the tightest certificate.

D.4 Deterministic argmax grouping at inference

For row-wise argmax $H(R)$ with any fixed tie-breaking rule, a separate deterministic bound is

$$\begin{aligned} m(H(R)) &\le M_{\mathrm{A}}(\varepsilon) := \min\left\{ n,\ n\left[\frac{1}{K} + \sqrt{\frac{\varepsilon}{2}} + \varepsilon\right]\right\}, \\ \mathrm{Lip}_F(\mathcal{A}_{H(R)}) &\le B(M_{\mathrm{A}}(\varepsilon)), \qquad \varepsilon = \log K - I(X;Y). \end{aligned} \tag{43}$$

Proof. Retain $\pi, \bar{h}, \varepsilon$ from Eqs. (36)–(37). Let $h_{s,:}$ be the one-hot row obtained by argmax, and write $a_s = \max_k r_{s,k}$. For each coordinate,

$$|h_{s,k} - r_{s,k}| \le 1 - a_s \le -\log a_s \le H(r_{s,:}). \tag{44}$$

The first inequality holds for the selected coordinate and every other coordinate, including ties. The last follows from $-\log r_{s,k} \ge -\log a_s$ for all positive entries. Thus, for $M_k = \sum_s h_{s,k}$, Eq. (38) and $d, \bar{h} \le \varepsilon$ give

$$\frac{M_k}{n} \le \pi_k + \frac{1}{n}\sum_s |h_{s,k} - r_{s,k}| \le \frac{1}{K} + \sqrt{\frac{\varepsilon}{2}} + \bar{h} \le \frac{1}{K} + \sqrt{\frac{\varepsilon}{2}} + \varepsilon. \tag{45}$$

Taking the maximum and using $m(H(R)) \le n$ proves the size bound. Since $M_{\mathrm{A}}(\varepsilon) \ge n/K \ge 1$, Theorem 2.1 proves the sensitivity bound. No sampling assumption or failure probability is needed. □

D.5 Patching, multiple heads, and simultaneous bounds

At fixed K, v, β, δ and information gap, the dense ceiling $B(n)$ and the information-controlled size bounds are nondecreasing in n. Spatial patching replaces N with $n = \lceil N/L_s \rceil$, and approximately balanced groups reduce the maximum interacting set to about n/K. This gives spatial patching and clustering complementary roles in controlling the derived bound.

The deterministic bound in Appendix D.4 applies separately to every temporal patch. For a simultaneous sampled bound over $T' \ge 1$ patches, put $\varepsilon_* = \max_p \varepsilon_p$. Applying Theorem 2.2 with failure probability δ/T' per patch and taking a union bound replaces n/δ by nT'/δ and the information gap by ε_* in Eq. (16). No independence between temporal patches is required, provided each analyzed patch uses independent token-wise sampling conditional on its pre-draw soft assignments. Temporal patching changes the confidence factor, not the size of a single spatial-attention block.

For layer ℓ, let $\overline{\mathcal{L}}^{(\ell)}_{\mathrm{info}} = -T'^{-1}\sum_p I^{(\ell)}_p$. Then $\sum_p \varepsilon^{(\ell)}_p = T'(\overline{\mathcal{L}}^{(\ell)}_{\mathrm{info}} + \log K_\ell)$, and Eq. (15) sums these layer-wise averages. The training objective therefore reduces the aggregate information gap across layers and temporal patches while remaining invariant to cluster-label permutations, allowing groups to adapt locally over time.

The block argument also extends to multiple heads sharing a partition. If head h has non-residual bound $b_h(q) = v_h[\beta_h + (1+\beta_h)\sqrt{s_{\beta_h}(q)}]$ and the concatenated outputs are projected by W_O, the corresponding residual multi-head bound is

$$\mathrm{Lip}_F(\mathcal{A}^{\mathrm{MH}}_H) \le 1 + \|W_O\|_2 \left(\sum_h b_h(m(H))^2\right)^{1/2}. \tag{46}$$

This follows by summing squared head-output differences before applying the output projection. Normalization, feed-forward layers, regional attention, and the prediction head require their own bounds when analyzing a larger composition.

E Properties of InfoLoss

InfoLoss maximizes mutual information between station-token data and cluster assignments separately within each temporal patch. At patch p, a token is drawn uniformly from the $n = N'$ data items and its cluster is sampled according to the corresponding row of $\mathbf{R}_p$. The empirical conditional distribution is therefore $\Pr(Y_p = k \mid X_p = \mathbf{x}^{\mathrm{clu}}_{p,s}) = R_{p,s,k}$, and the marginal is $\pi_{p,k} = n^{-1} \sum_s R_{p,s,k}$. The following properties concern the soft assignments; hard groups are obtained by sampling or argmax as described in Section 2.3.1.

Cluster balance. The marginal entropy satisfies $H(Y_p) = \log K - \mathrm{KL}(\boldsymbol{\pi}_p \| u_K)$, where u_K is uniform. Maximizing this entropy encourages balanced soft cluster masses and discourages collapse into a few clusters. This equal-mass preference discourages cluster collapse and promotes broad use of the available regional units.

Assignment confidence. The conditional entropy is exactly the average row entropy,

$$H(Y_p \mid X_p) = \frac{1}{n} \sum_s H(\mathbf{R}_{p,s,:}). \tag{47}$$

Minimizing it favors confident, near-deterministic token-to-cluster assignments. Together with evolution-driven centroids, this produces clear and input-adaptive regional memberships.

Joint effect and temporal scope. For $\ell_p = -I(X_p; Y_p)$, the gap to the lower bound $-\log K$ is

$$\varepsilon_p = \ell_p + \log K = H(Y_p \mid X_p) + \mathrm{KL}(\boldsymbol{\pi}_p \| u_K). \tag{48}$$

Both terms are nonnegative, so a small gap jointly promotes confident assignments and balanced cluster usage. Theorem 2.2 translates this gap into a probabilistic upper bound on the largest sampled group and hence into the fixed-partition Lipschitz upper bound of Theorem 2.1. Appendix D.4 provides the deterministic argmax counterpart. A balanced deterministic partition attains zero when divisibility permits. Because the loss is evaluated independently at each temporal patch and is invariant to cluster-label permutations, it supports the locally adaptive regrouping central to STCFormer.

F Experimental Setup Details

F.1 Datasets and Preprocessing

Table 4: Statistics of the processed weather datasets. Each variable defines a forecasting task.

Statistic	French	Hunan	Global
Stations	234	3,570	3,850
Time steps	8,784	4,368	17,544
Start	2016-01-01 00:00	2023-04-01 09:00	2019-01-01 00:00
End	2016-12-31 23:00	2023-09-30 08:00	2020-12-31 23:00
Interval	1 hour	1 hour	1 hour
Variables	Temp., U-wind, V-wind	Temp., U-wind, V-wind	Temp., wind speed

The **French dataset**[1] comprises hourly temperature, u-wind, and v-wind variables collected from 234 meteorological stations, spanning from January 1, 2016, to December 31, 2016. This dataset is derived from the MeteoNet dataset. The **Hunan dataset** contains hourly temperature, u-wind, and v-wind data collected from 3,570 meteorological stations, covering the period from April 1, 2023,

[1] https://github.com/meteofrance/meteonet

to September 30, 2023. This dataset is provided by the Hunan Meteorological Bureau. The **Global dataset** includes hourly temperature and wind speed variables collected from 3,850 meteorological stations worldwide, spanning from January 1, 2019, to December 31, 2020. This dataset was curated by Corrformer Wu et al. (2023). For all datasets, we partition the data into training, validation, and test sets with a ratio of 6:2:2. We apply Z-score normalization using the statistics of the training set.

Table 4 summarizes the processed datasets, comprising eight separate dataset–variable forecasting tasks. Station counts refer to the original stations before spatial patching. The chronological split uses $\lfloor 0.2T \rfloor$ time steps for each of validation and testing, with the remainder used for training, where T is the total sequence length.

F.2 Baselines

In our experiments, we compare STCFormer against 15 baselines in two categories. The six spatio-temporal methods are DCRNN Li et al. (2018), Corrformer Wu et al. (2023), EasyST Tang et al. (2024), CDPNet Xu et al. (2025), STELLA Fu et al. (2025), and S^{2}Transformer Chen et al. (2026). The nine multivariate time series methods are DLinear Zeng et al. (2023), PatchTST Nie et al. (2023), Crossformer Zhang & Yan (2023), TimeXer Wang et al. (2024), CCM Chen et al. (2024), DUET Qiu et al. (2025), TimeFilter Hu et al. (2025), TQNet Lin et al. (2025), and xPatch Stitsyuk & Choi (2025). CCM uses DLinear as its forecasting backbone and learns cluster-specific shared parameters; tables abbreviate this combination as CCM. These baselines cover graph-based spatial modeling, station-specific forecasting, and multivariate temporal modeling.

F.3 Implementation and Hyperparameters

Training and evaluation. All experiments are implemented with PyTorch and BasicTS Shao et al. (2025) on a server equipped with an NVIDIA A100 GPU with 40 GB of memory. We use the past 48 hours to predict the next 24 and 72 hours, with MSE as the training loss. The maximum number of training epochs is 50, with an early-stopping patience of 5. All models use Adam with an initial learning rate of 10^{-4}, a weight decay of 10^{-4}, and a cosine annealing learning rate scheduler. For TimeFilter on Hunan and Global, we use a temporal patch length of 24 and 30 training epochs; the French configuration is unchanged. All reported experimental results, including hyperparameter sweeps, are means of three runs; tables also report standard deviations where available. Efficiency benchmarks use batch size 8 for every method. The main text reports 24-hour MSE. Additional forecasting metrics and 72-hour results are provided in Appendix G, and additional ablation metrics in Appendix H.

Evaluation metrics. MSE and MAE are computed after inverse normalization, in the original data space. For each dataset's target-variable task, SEDI uses one fixed threshold shared by all stations: the 95th percentile of the training-period observations pooled across all stations. The threshold is then held fixed for all methods and evaluated horizons. For each run, we pool all test samples into a single contingency table and compute SEDI once, rather than averaging station-wise scores. Let $h = \mathrm{TP}/(\mathrm{TP} + \mathrm{FN})$ and $f = \mathrm{FP}/(\mathrm{FP} + \mathrm{TN})$ denote the pooled hit and false-alarm rates. Then $\mathrm{SEDI} = [\log f - \log h - \log(1-f) + \log(1-h)]/[\log f + \log h + \log(1-f) + \log(1-h)]$, with larger values indicating better extreme-event discrimination. Reported SEDI values summarize the three run-level scores.

Model configuration. The common settings across datasets are an embedding dimension of 256, an FFN dimension of 512, a dropout rate of 0.1, and four attention heads. The temporal and spatial patch sizes (L_t, L_s) are $(4, 2)$ for French, $(4, 7)$ for Hunan, and $(6, 7)$ for Global. STCFormer uses three layers, a clustering ratio of 0.05, and an InfoLoss weight of $\lambda = 0.1$ on all datasets. InfoLoss is the negative within-patch mutual information in Eq. (15), computed at every layer, averaged over temporal patches, and summed across layers; no additional entropy coefficient is used. The layer counts of the baselines follow their original papers. For models requiring a predefined graph, we construct a 20-nearest-neighbor spatial adjacency matrix using station latitude and longitude.

G Additional Forecasting Results

Tables 5–9 report 24-hour MAE and SEDI and all 72-hour metrics under the main experimental setup. Across the complete 48 comparisons, STCFormer ranks first in 42 and second in five, demonstrating consistently strong performance across metrics and horizons.

Table 5: 24-hour forecasting results (MAE, ↓) with a 48-hour look-back window. Values are mean ± standard deviation over three runs. Bold and underlined means indicate the best and second-best reported values in each column; ties share the same marking.

Method	French			Hunan			Global	
	Temp.	U-Wind	V-Wind	Temp.	U-Wind	V-Wind	Temp.	Wind
Spatio-Temporal Methods								
DCRNN	<u>1.894</u>±0.017	1.791±0.040	2.157±0.025	1.558±0.456	0.654±0.005	0.783±0.009	1.941±0.023	<u>1.282</u>±0.000
Corrformer	2.121±0.018	1.540±0.016	1.794±0.005	1.662±0.126	0.675±0.001	0.756±0.002	1.888±0.004	1.304±0.011
EasyST	2.138±0.100	1.554±0.023	1.673±0.000	1.829±0.230	0.683±0.012	0.762±0.012	1.843±0.003	1.289±0.002
CDPNet	2.083±0.022	1.502±0.004	1.535±0.007	1.696±0.009	0.691±0.000	0.789±0.004	1.843±0.001	1.290±0.002
STELLA	4.099±0.291	2.438±0.166	2.336±0.155	3.674±0.170	1.038±0.022	1.304±0.057	1.966±0.002	1.315±0.000
S^2Transformer	2.141±0.016	1.484±0.030	1.567±0.057	1.955±0.099	0.689±0.001	0.805±0.001	<u>1.802</u>±0.001	1.290±0.000
Multivariate Time Series Methods								
DLinear	1.964±0.003	1.519±0.001	1.575±0.000	1.583±0.002	0.643±0.000	0.723±0.000	2.062±0.000	1.392±0.000
PatchTST	1.971±0.006	1.515±0.003	1.608±0.002	1.577±0.005	0.652±0.001	0.725±0.001	1.918±0.000	1.380±0.001
Crossformer	2.111±0.005	<u>1.408</u>±0.000	1.520±0.004	1.634±0.015	0.659±0.003	0.750±0.003	1.903±0.007	1.299±0.003
TimeXer	1.907±0.025	1.445±0.008	1.551±0.001	1.634±0.026	0.659±0.001	0.730±0.000	1.912±0.003	1.352±0.000
CCM	1.903±0.004	1.410±0.002	1.502±0.001	1.505±0.003	0.643±0.000	0.718±0.001	1.891±0.000	1.330±0.000
DUET	1.971±0.029	1.465±0.004	1.484±0.012	1.697±0.004	0.646±0.006	0.786±0.011	1.872±0.016	1.288±0.006
TimeFilter	1.940±0.012	1.416±0.004	<u>1.483</u>±0.002	1.529±0.018	<u>0.639</u>±0.004	<u>0.712</u>±0.009	1.853±0.025	<u>1.282</u>±0.009
TQNet	1.930±0.001	1.426±0.003	1.505±0.008	<u>1.492</u>±0.008	0.646±0.000	0.719±0.002	1.840±0.001	1.297±0.000
xPatch	1.982±0.003	1.472±0.000	1.547±0.001	1.545±0.003	0.643±0.000	0.720±0.000	1.925±0.001	1.322±0.000
STCFormer	**1.861**±0.015	**1.400**±0.008	**1.466**±0.004	**1.482**±0.026	**0.622**±0.001	**0.700**±0.002	**1.757**±0.000	**1.273**±0.001

Table 6: 24-hour forecasting results (SEDI, ↑) with a 48-hour look-back window. Values are mean ± standard deviation over three runs. Bold and underlined means indicate the best and second-best reported values in each column; ties share the same marking.

Method	French			Hunan			Global	
	Temp.	U-Wind	V-Wind	Temp.	U-Wind	V-Wind	Temp.	Wind
Spatio-Temporal Methods								
DCRNN	0.556±0.061	0.676±0.006	0.607±0.053	0.583±0.091	0.561±0.030	0.613±0.052	**0.920**±0.003	0.648±0.008
Corrformer	0.503±0.012	0.645±0.019	**0.616**±0.015	0.649±0.058	0.532±0.001	0.627±0.001	0.887±0.008	0.595±0.002
EasyST	0.632±0.004	0.598±0.000	0.483±0.003	0.701±0.134	0.541±0.010	0.551±0.007	0.901±0.001	0.631±0.001
CDPNet	0.625±0.002	0.671±0.004	0.607±0.004	0.735±0.021	0.522±0.006	<u>0.629</u>±0.015	0.916±0.001	0.639±0.008
STELLA	0.452±0.048	0.530±0.029	0.400±0.041	0.492±0.089	0.480±0.025	0.533±0.054	0.899±0.003	0.597±0.001
S^2Transformer	0.554±0.015	0.576±0.006	0.493±0.001	0.653±0.069	0.533±0.035	0.592±0.027	0.911±0.001	0.617±0.005
Multivariate Time Series Methods								
DLinear	0.644±0.002	0.607±0.000	0.524±0.000	<u>0.744</u>±0.000	0.502±0.000	0.547±0.000	0.898±0.001	0.602±0.001
PatchTST	0.644±0.004	0.649±0.000	0.587±0.006	0.742±0.012	0.567±0.001	0.598±0.011	0.910±0.000	0.621±0.001
Crossformer	0.587±0.042	<u>0.684</u>±0.013	0.602±0.012	0.741±0.008	0.513±0.005	0.621±0.008	0.908±0.001	0.647±0.009
TimeXer	0.630±0.001	0.645±0.000	0.603±0.016	0.670±0.005	0.572±0.003	<u>0.629</u>±0.002	0.913±0.001	0.646±0.001
CCM	0.643±0.006	0.671±0.005	0.607±0.006	0.707±0.012	0.571±0.002	<u>0.629</u>±0.002	0.916±0.000	0.628±0.001
DUET	0.619±0.005	0.637±0.003	0.587±0.008	0.717±0.042	0.525±0.013	0.600±0.004	0.910±0.002	0.616±0.000
TimeFilter	0.638±0.022	0.670±0.009	0.609±0.007	0.723±0.013	<u>0.573</u>±0.001	<u>0.629</u>±0.005	0.917±0.001	<u>0.650</u>±0.001
TQNet	<u>0.669</u>±0.007	0.671±0.004	0.605±0.003	0.740±0.005	0.571±0.003	0.611±0.003	0.916±0.001	0.643±0.001
xPatch	0.647±0.011	0.658±0.001	0.589±0.001	0.740±0.009	0.567±0.001	0.604±0.001	0.912±0.000	0.638±0.001
STCFormer	**0.672**±0.001	**0.689**±0.007	<u>0.615</u>±0.006	**0.751**±0.009	**0.580**±0.001	**0.639**±0.017	<u>0.918</u>±0.002	**0.653**±0.001

Table 7: 72-hour forecasting results (MAE, ↓) with a 48-hour look-back window. Values are mean ± standard deviation over three runs. Bold and underlined means indicate the best and second-best reported values in each column; ties share the same marking.

Method	French			Hunan			Global	
	Temp.	U-Wind	V-Wind	Temp.	U-Wind	V-Wind	Temp.	Wind
Spatio-Temporal Methods								
DCRNN	3.302±0.016	1.987±0.126	2.103±0.127	2.388±0.034	0.733±0.003	0.891±0.026	2.924±0.340	1.492±0.003
Corrformer	2.711±0.030	2.100±0.034	1.996±0.008	2.813±0.245	0.721±0.002	0.893±0.005	2.784±0.005	1.465±0.002
EasyST	2.878±0.063	1.992±0.016	1.957±0.049	4.431±0.660	0.710±0.002	0.855±0.010	2.502±0.018	1.498±0.000
CDPNet	2.862±0.007	1.973±0.005	1.966±0.010	2.418±0.028	0.723±0.003	0.869±0.001	2.511±0.001	1.499±0.002
STELLA	4.631±0.019	2.754±0.077	2.487±0.089	4.042±0.269	1.015±0.013	1.266±0.009	2.588±0.005	1.474±0.002
S^2Transformer	2.910±0.148	2.062±0.071	1.969±0.016	2.940±0.162	0.765±0.013	0.978±0.015	2.564±0.000	1.465±0.001
Multivariate Time Series Methods								
DLinear	2.598±0.000	1.944±0.003	1.936±0.002	2.383±0.003	0.714±0.000	0.843±0.000	2.644±0.000	1.515±0.000
PatchTST	2.599±0.006	1.975±0.002	1.952±0.001	2.420±0.033	0.716±0.001	0.861±0.001	2.559±0.000	1.495±0.000
Crossformer	2.633±0.002	1.977±0.039	1.956±0.019	2.375±0.046	0.713±0.001	0.860±0.003	2.505±0.011	1.464±0.004
TimeXer	2.597±0.008	1.923±0.007	1.944±0.007	2.420±0.018	0.708±0.002	0.852±0.005	2.531±0.001	1.462±0.002
CCM	2.599±0.003	**1.912**±0.002	1.948±0.005	2.456±0.017	0.718±0.000	0.858±0.002	2.568±0.001	1.500±0.000
DUET	2.646±0.028	1.952±0.001	1.934±0.022	2.406±0.002	0.717±0.006	0.902±0.003	2.502±0.008	1.459±0.000
TimeFilter	2.591±0.013	1.956±0.008	1.953±0.012	2.376±0.004	0.715±0.002	0.862±0.015	2.661±0.011	1.472±0.008
TQNet	2.594±0.001	1.940±0.006	1.934±0.004	2.451±0.008	0.719±0.000	0.854±0.010	2.512±0.002	1.467±0.000
xPatch	2.596±0.002	1.977±0.001	1.952±0.001	2.417±0.011	0.716±0.000	0.862±0.000	2.582±0.001	1.486±0.000
STCFormer	**2.585**±0.006	1.917±0.044	**1.925**±0.015	**2.328**±0.100	**0.707**±0.003	**0.827**±0.007	**2.493**±0.004	**1.436**±0.001

Table 8: 72-hour forecasting results (MSE, ↓) with a 48-hour look-back window. Values are mean ± standard deviation over three runs. Bold and underlined means indicate the best and second-best reported values in each column; ties share the same marking.

Method	French			Hunan			Global	
	Temp.	U-Wind	V-Wind	Temp.	U-Wind	V-Wind	Temp.	Wind
Spatio-Temporal Methods								
DCRNN	19.294±0.154	7.819±0.883	9.671±1.655	11.447±0.063	2.369±0.052	3.567±0.109	17.582±4.554	4.852±0.018
Corrformer	12.053±0.319	9.043±0.302	8.401±0.023	14.779±0.658	2.223±0.027	3.735±0.001	13.125±0.010	4.826±0.006
EasyST	13.579±0.614	7.361±0.078	7.437±0.387	31.283±7.987	2.216±0.021	3.263±0.074	12.895±0.239	4.892±0.017
CDPNet	13.508±0.064	7.838±0.022	8.214±0.105	11.485±0.351	2.371±0.026	3.494±0.001	12.985±0.077	5.011±0.008
STELLA	33.753±0.423	12.604±0.623	10.618±0.737	25.984±3.066	2.537±0.029	3.797±0.019	13.075±0.067	4.797±0.011
S^2Transformer	13.937±1.781	7.855±0.363	7.197±0.091	14.260±1.227	2.220±0.034	3.357±0.125	13.135±0.127	4.754±0.016
Multivariate Time Series Methods								
DLinear	11.287±0.019	7.214±0.012	7.722±0.009	11.458±0.025	2.261±0.000	3.333±0.000	14.201±0.014	5.018±0.007
PatchTST	11.399±0.044	7.793±0.000	8.109±0.039	11.629±0.158	2.331±0.008	3.474±0.001	13.570±0.006	5.004±0.004
Crossformer	11.447±0.030	8.025±0.478	7.969±0.106	12.589±0.247	2.315±0.012	3.463±0.098	13.322±0.119	4.842±0.017
TimeXer	11.383±0.082	7.446±0.044	8.065±0.034	11.702±0.274	2.258±0.007	3.309±0.042	13.390±0.023	4.808±0.005
CCM	11.293±0.022	7.295±0.036	8.023±0.050	12.934±0.129	2.329±0.001	3.384±0.007	13.656±0.017	5.041±0.004
DUET	11.929±0.207	7.236±0.024	7.895±0.162	**11.337**±0.117	2.222±0.020	3.627±0.017	12.633±0.107	4.904±0.069
TimeFilter	11.415±0.186	7.666±0.116	8.086±0.112	11.365±0.093	2.303±0.016	3.390±0.129	13.287±0.043	4.808±0.019
TQNet	11.430±0.011	7.408±0.030	7.980±0.060	11.377±0.062	2.317±0.003	3.336±0.080	13.121±0.014	4.872±0.001
xPatch	11.285±0.003	7.696±0.004	8.094±0.005	11.709±0.140	2.326±0.001	3.469±0.006	13.803±0.007	4.952±0.003
STCFormer	**11.145**±0.015	**7.120**±0.454	**7.043**±0.065	11.432±0.981	**2.208**±0.013	**3.249**±0.051	**12.531**±0.074	**4.750**±0.003

Table 9: 72-hour forecasting results (SEDI, ↑) with a 48-hour look-back window. Values are mean ± standard deviation over three runs. Bold and underlined means indicate the best and second-best reported values in each column; ties share the same marking.

Method	French			Hunan			Global	
	Temp.	U-Wind	V-Wind	Temp.	U-Wind	V-Wind	Temp.	Wind
Spatio-Temporal Methods								
DCRNN	0.394±0.173	0.456±0.004	0.406±0.010	0.548±0.025	0.416±0.006	0.441±0.025	0.867±0.010	0.521±0.003
Corrformer	0.342±0.015	0.472±0.010	0.370±0.014	0.328±0.046	0.481±0.007	0.473±0.016	0.849±0.004	0.450±0.003
EasyST	0.499±0.003	0.463±0.003	0.371±0.010	0.447±0.034	0.424±0.009	0.463±0.005	0.874±0.001	0.494±0.002
CDPNet	0.498±0.002	0.513±0.003	0.395±0.000	0.550±0.006	0.456±0.004	0.460±0.000	0.880±0.005	0.520±0.002
STELLA	0.277±0.014	0.397±0.012	0.284±0.028	0.349±0.029	0.381±0.006	0.402±0.010	0.865±0.004	0.455±0.002
S^2Transformer	0.304±0.084	0.419±0.001	0.353±0.014	0.376±0.046	0.412±0.020	0.449±0.037	0.871±0.013	0.477±0.004
Multivariate Time Series Methods								
DLinear	0.504±0.005	0.462±0.000	0.386±0.000	0.509±0.001	0.400±0.000	0.428±0.000	0.858±0.001	0.480±0.001
PatchTST	0.486±0.011	0.494±0.002	0.406±0.005	0.486±0.010	0.478±0.005	0.463±0.001	0.882±0.000	0.503±0.000
Crossformer	-0.270±0.005	**0.521**±0.009	0.425±0.019	0.504±0.027	0.467±0.004	0.476±0.017	0.878±0.002	0.511±0.006
TimeXer	0.470±0.005	0.516±0.012	0.424±0.005	0.479±0.012	0.484±0.009	**0.481**±0.002	0.886±0.001	0.526±0.001
CCM	0.510±0.001	0.519±0.001	0.434±0.006	0.527±0.003	0.485±0.002	0.468±0.001	0.880±0.001	0.506±0.000
DUET	0.473±0.004	0.458±0.004	0.423±0.002	0.487±0.008	0.444±0.005	0.467±0.010	0.875±0.010	0.488±0.004
TimeFilter	0.515±0.002	0.504±0.006	0.403±0.022	0.538±0.010	0.478±0.000	0.479±0.002	0.846±0.002	0.527±0.002
TQNet	0.518±0.007	0.513±0.002	0.425±0.004	0.513±0.001	0.484±0.001	0.464±0.004	0.886±0.001	0.524±0.001
xPatch	0.514±0.004	0.502±0.001	0.415±0.003	0.494±0.004	0.476±0.000	0.465±0.000	0.883±0.000	0.521±0.001
STCFormer	**0.527**±0.008	0.520±0.009	**0.438**±0.009	**0.581**±0.027	**0.496**±0.007	0.480±0.018	**0.889**±0.001	**0.530**±0.000

H Additional Ablation Results

Tables 10 and 11 report the MAE and SEDI results corresponding to the 24-hour MSE ablation in Table 2. The full model attains the best reported mean on all eight tasks for both metrics, consistent with the MSE comparison. The variants and all other experimental settings follow the main ablation study.

Table 10: 24-hour ablation results (MAE, ↓) with a 48-hour look-back window. Values are mean ± standard deviation over three runs. Bold means indicate the best reported values in each column; ties are both bold.

Variant	French			Hunan			Global	
	Temp.	U-Wind	V-Wind	Temp.	U-Wind	V-Wind	Temp.	Wind
Window-shared	1.885±0.015	1.411±0.008	1.497±0.016	1.522±0.020	0.652±0.008	0.721±0.006	1.766±0.003	1.325±0.010
Full Attention	1.869±0.010	1.412±0.019	1.490±0.032	1.518±0.023	0.649±0.000	0.722±0.001	1.766±0.001	1.298±0.001
Static Clustering	1.886±0.023	1.414±0.013	1.492±0.009	1.508±0.043	0.649±0.002	0.725±0.002	1.764±0.003	1.298±0.001
w/o InfoLoss	1.906±0.046	1.412±0.009	1.479±0.005	1.549±0.009	0.659±0.002	0.719±0.001	1.767±0.000	1.295±0.001
w/o Global Attn.	1.877±0.009	1.410±0.003	1.477±0.011	1.572±0.009	0.653±0.000	0.728±0.002	1.773±0.001	1.294±0.001
w/o Local Attn.	1.887±0.006	1.418±0.019	1.474±0.001	1.491±0.022	0.650±0.000	0.724±0.001	1.783±0.001	1.301±0.002
STCFormer	**1.861**±0.015	**1.400**±0.008	**1.466**±0.004	**1.482**±0.026	**0.622**±0.001	**0.700**±0.002	**1.757**±0.000	**1.273**±0.001

Table 11: 24-hour ablation results (SEDI, ↑) with a 48-hour look-back window. Values are mean ± standard deviation over three runs. Bold means indicate the best reported values in each column; ties are both bold.

Variant	French			Hunan			Global	
	Temp.	U-Wind	V-Wind	Temp.	U-Wind	V-Wind	Temp.	Wind
Window-shared	0.665±0.001	0.664±0.000	0.600±0.018	0.718±0.013	0.544±0.005	0.611±0.006	0.910±0.001	0.638±0.001
Full Attention	0.670±0.004	0.667±0.003	0.611±0.000	0.723±0.015	0.546±0.003	0.628±0.006	0.909±0.000	0.633±0.000
Static Clustering	0.668±0.013	0.656±0.004	0.609±0.031	0.724±0.036	0.546±0.001	0.625±0.009	0.914±0.002	0.649±0.001
w/o InfoLoss	0.670±0.003	0.655±0.002	0.596±0.007	0.713±0.010	0.553±0.007	0.632±0.004	0.911±0.000	0.645±0.000
w/o Global Attn.	0.665±0.000	0.658±0.006	0.607±0.017	0.712±0.017	0.553±0.001	0.627±0.013	0.912±0.001	0.646±0.001
w/o Local Attn.	0.661±0.003	0.661±0.009	0.598±0.011	0.714±0.020	0.557±0.003	0.630±0.006	0.909±0.000	0.649±0.000
STCFormer	**0.672**±0.001	**0.689**±0.007	**0.615**±0.006	**0.751**±0.009	**0.580**±0.001	**0.639**±0.017	**0.918**±0.002	**0.653**±0.001

I Additional Analyses

We extend the comparison to three time series foundation models and three grid-based predictive models. Table 12 summarizes 24-hour MSE on all eight dataset–variable tasks using a 48-hour input window. STCFormer results are those of the main experiment. We report mean and standard deviation, with the best mean in each column in bold.

Table 12: Additional 24-hour forecasting comparisons (MSE, ↓) with a 48-hour input window. Values are mean ± standard deviation over three runs. Bold means indicate the best reported values in each column; ties are both bold.

Method	French			Hunan			Global	
	Temp.	U-Wind	V-Wind	Temp.	U-Wind	V-Wind	Temp.	Wind
Time Series Foundation Models								
Timer	7.304±0.014	4.831±0.008	5.694±0.033	5.041±0.012	1.987±0.008	2.769±0.020	8.002±0.008	4.873±0.015
Chronos-Bolt-Tiny	6.706±0.012	4.538±0.002	5.437±0.002	4.990±0.003	1.924±0.003	2.537±0.000	7.894±0.004	4.691±0.005
Time-MoE	6.427±0.014	4.231±0.017	5.026±0.009	5.138±0.008	1.947±0.011	2.429±0.014	7.267±0.007	3.798±0.006
Grid-Based Predictive Models								
PhyDNet	10.225±0.487	5.224±0.278	6.259±0.106	6.028±0.050	2.315±0.026	3.192±0.054	22.217±0.422	6.830±0.005
PredRNN	8.793±0.409	4.278±0.069	5.276±0.236	5.359±0.057	2.313±0.004	3.276±0.065	20.139±0.169	6.574±0.007
SimVP	9.472±0.217	4.862±0.042	5.560±0.314	7.274±0.595	2.508±0.028	3.553±0.238	20.372±0.077	6.687±0.096
STCFormer	**6.291**±0.090	**4.045**±0.028	**4.883**±0.013	**4.743**±0.149	**1.832**±0.001	**2.402**±0.003	**6.917**±0.002	**3.584**±0.002

I.1 Comparison with Time Series Foundation Models

We evaluate Timer (Liu et al., 2024b), Chronos-Bolt-Tiny (Ansari et al., 2024), and Time-MoE (Shi et al., 2025). These models undergo full-parameter fine-tuning for five epochs on French and three epochs on Hunan and Global. STCFormer obtains lower mean MSE on all eight tasks, with reductions of 1.1%–5.6% relative to the strongest foundation-model baseline on each task. Time-MoE is the strongest of these baselines on six tasks, whereas Chronos-Bolt-Tiny performs best on Hunan temperature and U-wind. These results support the effectiveness of STCFormer under the evaluated fine-tuning budgets.

I.2 Comparison with Grid-Based Predictive Models

We additionally evaluate PhyDNet (Guen & Thome, 2020), PredRNN (Wang et al., 2023), and SimVP (Gao et al., 2022), each trained for 30 epochs. Station observations are interpolated onto a grid, forecasts are generated on that grid, and predictions are mapped back to station locations for evaluation. STCFormer achieves the lowest mean MSE on all eight tasks. On Global temperature and wind, it reduces MSE from the strongest grid baseline's 20.139 and 6.574 to 6.917 and 3.584,

respectively. Although interpolation can introduce station–grid conversion error, the consistent margin across all eight tasks highlights both the necessity and effectiveness of forecasting directly on irregular station networks.

I.3 Additional Hyperparameter Studies

We complement the InfoLoss-weight, cluster-ratio, and embedding-dimension studies in Section 3.4.1 with look-back length and spatio-temporal patch size. We report test MSE for temperature forecasting: the look-back study forecasts 72 hours on French temperature, while the patch-size study forecasts 24 hours. Each plotted value is the mean of three runs; error bars are omitted for compactness.

I.3.1 Look-Back Window Length

Figure 6 compares look-back windows of 24, 48, 96, and 168 hours for 72-hour French temperature forecasting. STCFormer attains MSE values of 11.498, 11.245, 11.318, and 11.501, respectively, with its lowest error at 48 hours. It achieves the lowest reported MSE among the compared methods at 24- and 48-hour look-back lengths; CCM performs best at 96 and 168 hours. The results support the 48-hour reference window, while showing that longer histories do not consistently improve performance. CCM uses DLinear as its forecasting backbone, as in the main comparison.

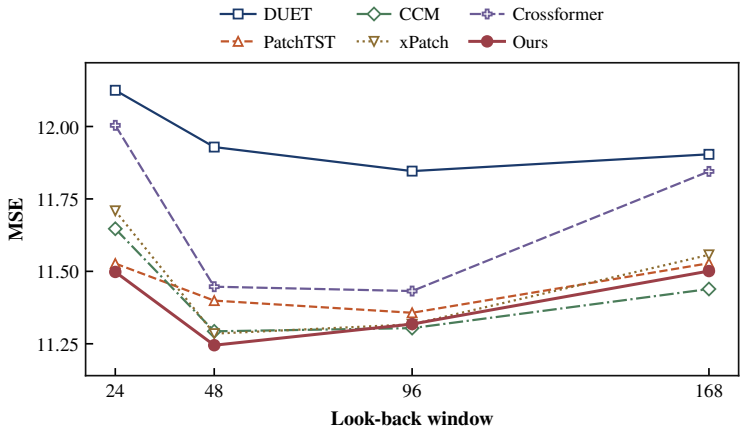
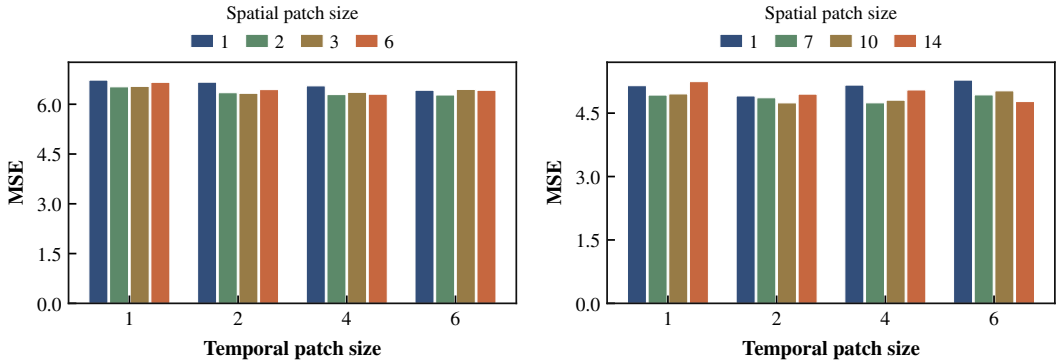


Figure 6: Look-back sensitivity on French temperature with a fixed 72-hour forecasting horizon. The horizontal axis is the input-window length in hours.

I.3.2 Spatio-Temporal Patch Sizes

We jointly vary the temporal patch length $L_t \in \{1, 2, 4, 6\}$ and the spatial patch length L_s, using $L_s \in \{1, 2, 3, 6\}$ on French and $L_s \in \{1, 7, 10, 14\}$ on Hunan. Figure 7 presents the resulting 24-hour MSE. The French sweep favors $(L_t, L_s) = (6, 2)$, while the main experiments use $(4, 2)$. Hunan reaches 4.742 at $(2, 10)$, almost identical to 4.743 at $(4, 7)$. With $L_t = 4$, replacing $L_s = 1$ by 2 on French and 7 on Hunan reduces MSE from 6.553 to 6.291 and from 5.163 to 4.743, respectively. Joint spatial and temporal patching is thus useful in these settings, but the best combination is dataset-dependent and coarser patches do not uniformly improve performance.



(a) French temperature.

(b) Hunan temperature.

Figure 7: Joint sensitivity to temporal and spatial patch sizes (24-hour MSE). Each bar represents one (L_t, L_s) configuration; colors distinguish spatial patch lengths.

I.4 Additional Efficiency Analysis

We complement the Hunan comparison in Section 3.4.2 with French temperature forecasting, using a 48-hour input window, a 24-hour horizon, and batch size 8 for every method. Figure 8 plots mean training time per epoch against test MSE, with bubble area proportional to peak training GPU memory. The time axis is logarithmic to accommodate the wide runtime range.

Among the methods shown, STCFormer achieves the lowest reported MSE (6.291), taking 15.306 ± 0.003 s/epoch and 0.378 GiB. It is $18.1\times$ faster than DCRNN; compared with TimeFilter, it uses 28.3% less training time and 86.8% less memory. PatchTST and TimeXer train faster, whereas DUET uses less memory. These results demonstrate a favorable accuracy–efficiency trade-off across methods with substantially different computational profiles.

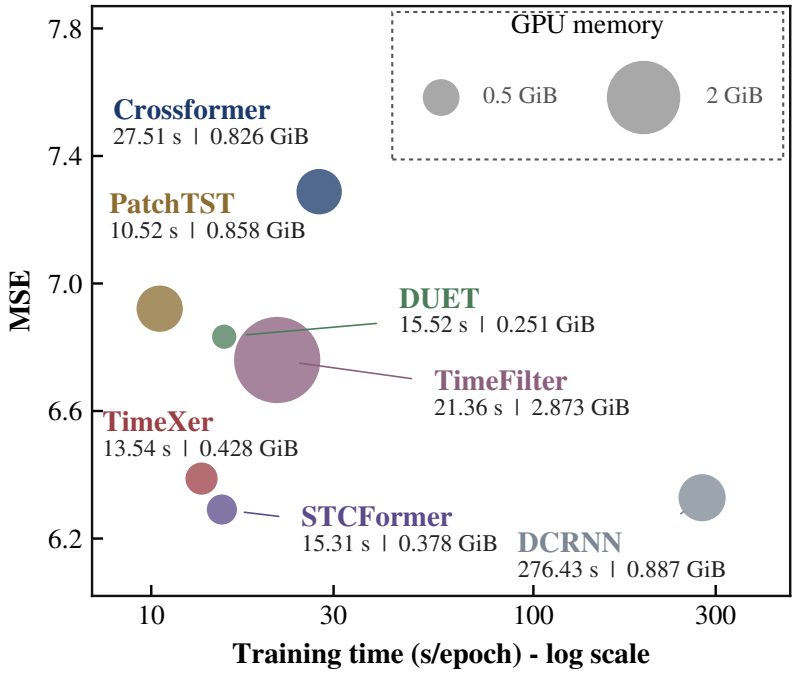


Figure 8: Efficiency on French temperature (48-to-24-hour forecasting, batch size 8). Bubble area indicates peak training GPU memory; labels report time per epoch and memory. The time axis uses a logarithmic scale.

I.4.1 Comparison with Full Attention

We compare CGAB with the Full Attention variant on French and Global temperature forecasting, using a 48-hour input window, a 24-hour horizon, and batch size 8.

Table 13 shows that CGAB lowers MSE on both datasets. Its clustering and regional aggregation increase epoch time by 22.7% on French and 11.4% on Global, while reducing peak memory on the larger Global network by 11.1%, from 10.023 to 8.913 GiB. The modest runtime cost therefore supports more accurate forecasts and, at larger spatial scale, lower memory use, making the additional structured interaction worthwhile.

Table 13: CGAB versus full attention. Time: s/epoch; memory: GiB. All values are three-run means; lower is better. Best values per dataset are bold.

Method	Time	Memory	MSE
French temperature			
Full Attention	**12.478**	**0.315**	6.334
CGAB	15.314	0.378	**6.291**
Global temperature			
Full Attention	**63.194**	10.023	6.976
CGAB	70.427	**8.913**	**6.917**

I.5 Sensitivity to Spatial Patch Construction

We compare four ways to organize stations into spatial patches: random grouping, Hilbert-curve ordering, capacity-constrained geographic clustering, and the serpentine ordering used in the main experiments. This comparison concerns patch construction before embedding, rather than the subsequent dynamic clustering in CGAB.

Table 14 shows lower mean MSE for all three geographic constructions than for random grouping on both datasets. Hilbert ordering performs best on French (6.152), whereas the default serpentine ordering performs best on Hunan (4.743). Relative to random grouping, serpentine ordering reduces mean MSE by 1.4% and 4.8%, respectively. These results validate geographically informed patch construction: every geographic strategy improves over random grouping, and the default serpentine ordering provides a strong, scalable choice across datasets.

Table 14: Spatial patch construction (temperature MSE). Values are mean ± standard deviation over three runs; best means are bold.

Construction	French	Hunan
Random	6.381 ± 0.043	4.982 ± 0.284
Hilbert curve	$\mathbf{6.152} \pm 0.018$	4.808 ± 0.044
Capacity-constrained geographic clustering	6.259 ± 0.140	4.782 ± 0.155
Serpentine (default)	6.291 ± 0.090	$\mathbf{4.743} \pm 0.149$

I.6 Numerical Diagnostics for the Theoretical Analysis

We examine whether the cluster-size dependence in Theorem 2.1 and the assignment behavior encouraged by InfoLoss are visible on French Temperature. Two matched one-layer models differ only in the InfoLoss weight (Full-1L: 0.1; NoInfo-1L: 0). Both use a 48-hour input, 24-hour forecast, 6-hour temporal patches, two-station spatial patches, $n = 117$ valid station tokens, and $K = 10$ clusters. Bound and assignment statistics cover all 1,685 test windows and eight temporal patches per window (13,480 records per model).

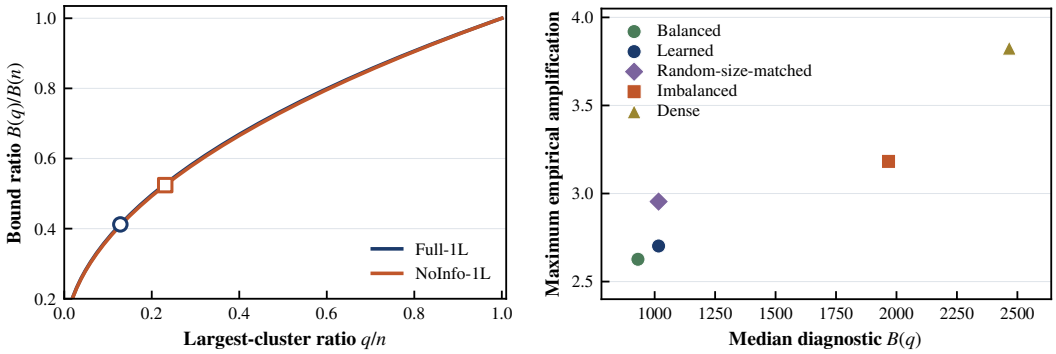
Cluster size and the diagnostic bound. Figure 9a evaluates the monotone $B(q)/B(n)$ dependence on the largest cluster size q; the curves follow the bound formula using constants reconstructed from the reported median and dense diagnostics, and markers show the observed medians. A smaller largest cluster yields a smaller fixed-partition diagnostic, while $q = n$ recovers the dense value. The mean diagnostic ratio is 0.4076 for Full-1L and 0.5337 for NoInfo-1L. Using the maximum observed local-attention input norm instantiates the theoretical dependence on the empirical test domain.

Effect of InfoLoss. Table 15 summarizes the matched ablation. With InfoLoss, the mean station–cluster mutual information is higher, while the largest cluster, cluster-size variation, and empty-cluster rate are lower. The mean local-to-dense diagnostic ratio falls by 0.1261. This is consistent with the theoretical chain from more informative, balanced assignments to smaller clusters and a tighter fixed-partition diagnostic.

Table 15: French Temperature one-layer assignment and bound diagnostics, averaged over 13,480 test-window/patch records per model. Δ is Full-1L minus NoInfo-1L.

Metric	Preferred	Full-1L	NoInfo-1L	Δ
Mutual information I	$\uparrow$	**2.2878**	2.0283	$+0.2594$
Largest-cluster ratio $m(H)/n$	$\downarrow$	**0.1248**	0.2454	-0.1206
Cluster-size CV	$\downarrow$	**0.1513**	0.6951	-0.5438
Empty-cluster rate	$\downarrow$	**0.0000**	0.0225	-0.0225
Bound ratio $B(m(H))/B(n)$	$\downarrow$	**0.4076**	0.5337	-0.1261

Fixed-partition perturbations. For the theorem-facing residual local-attention operator, we freeze each partition before perturbing its input and disable dropout. On the same 64 test windows (eight patches each), we evaluate 20 unit-Frobenius Gaussian directions at five relative perturbation magnitudes. Figure 9b compares only the *largest sampled* amplification for each partition family. The maxima increase from Balanced (2.6266) through Learned (2.7020), size-matched Random (2.9542), and Imbalanced (3.1824) to Dense (3.8256), consistent with the diagnostic ordering at this aggregate level. Balanced gives the smallest amplification and Learned remains close, with both clearly below size-matched Random, Imbalanced, and Dense. The separation between Learned and size-matched Random further shows that the learned partition aligns the restricted interactions with the data.



(a) Cluster size and the fixed-partition diagnostic.

(b) Largest sampled amplification.

Figure 9: Numerical diagnostics on French temperature. (a) Normalized bound versus largest-cluster ratio; open markers denote test-record medians. (b) Maximum sampled fixed-partition amplification versus median $B(q)$ across five partition families.

I.7 Prediction Visualizations

Figures 10–12 compare STCFormer with DLinear, PatchTST, Crossformer, TimeXer, DUET, S^{2}Transformer, and EasyST on French temperature, Hunan temperature, and Global wind speed. Each figure shows the same station and forecast window across methods, using 48 hours of observations to predict the following 72 hours. Gray curves denote the observed history, black curves the future ground truth, and red dashed curves the predictions; the shaded region marks the forecast horizon.

For French temperature, STCFormer closely follows the low-temperature troughs and the declining level of successive peaks. On Hunan temperature, it captures both the cooler regime after the observed drop and the subsequent daily fluctuations, whereas several baselines retain a warm bias. On Global wind speed, STCFormer reproduces the sustained increase and elevated values near the end of the horizon more faithfully than the comparatively flat forecasts of DLinear, PatchTST, and TimeXer. Together, these cases visually confirm STCFormer's ability to track regime changes, periodic variation, and evolving trends across distinct station forecasting settings.

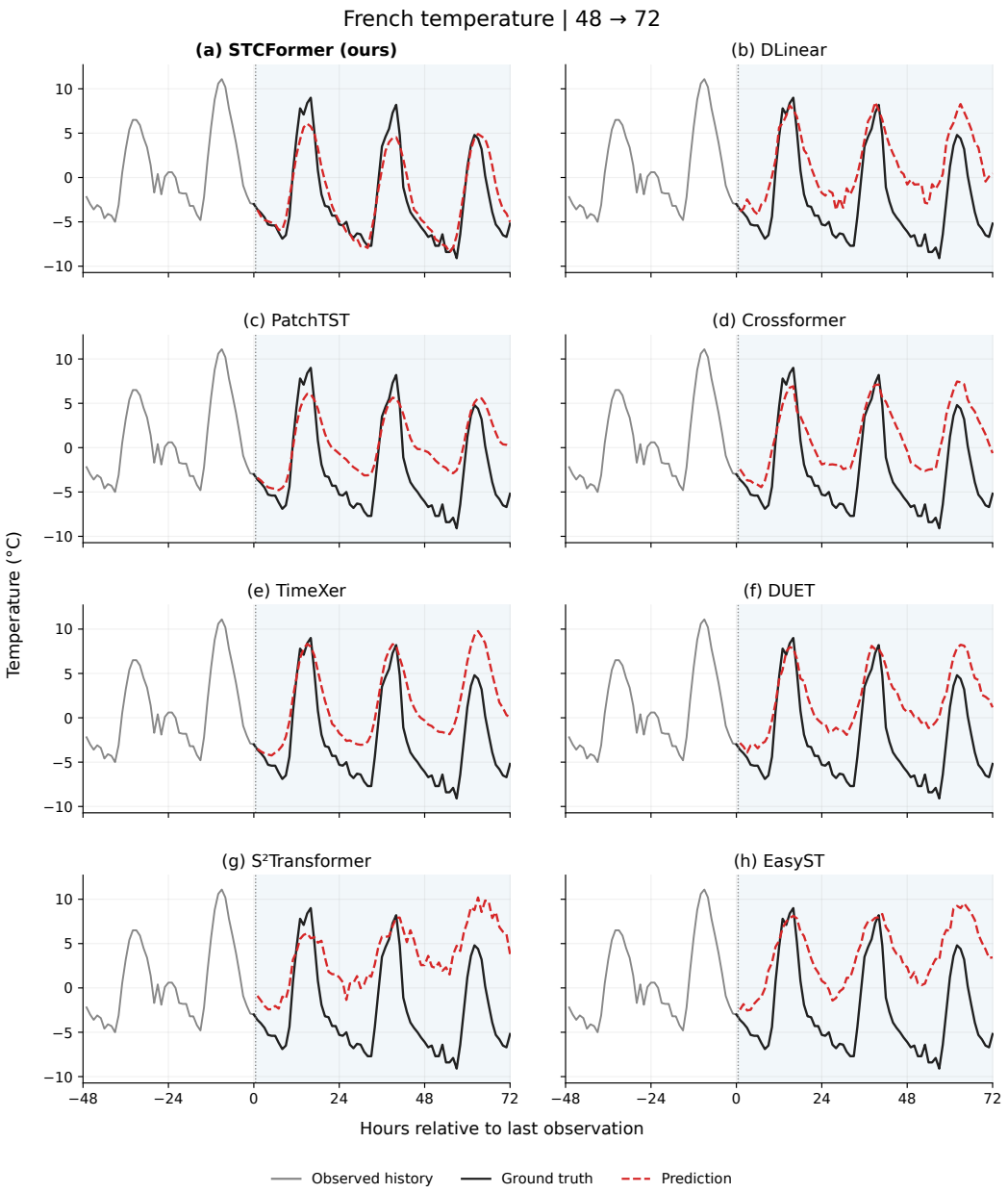


Figure 10: Prediction visualization on French temperature with a 48-hour input and a 72-hour forecast horizon. All methods are shown for the same station and window. Gray: observed history; black: ground truth; red dashed: prediction. The shaded region denotes the forecast horizon.

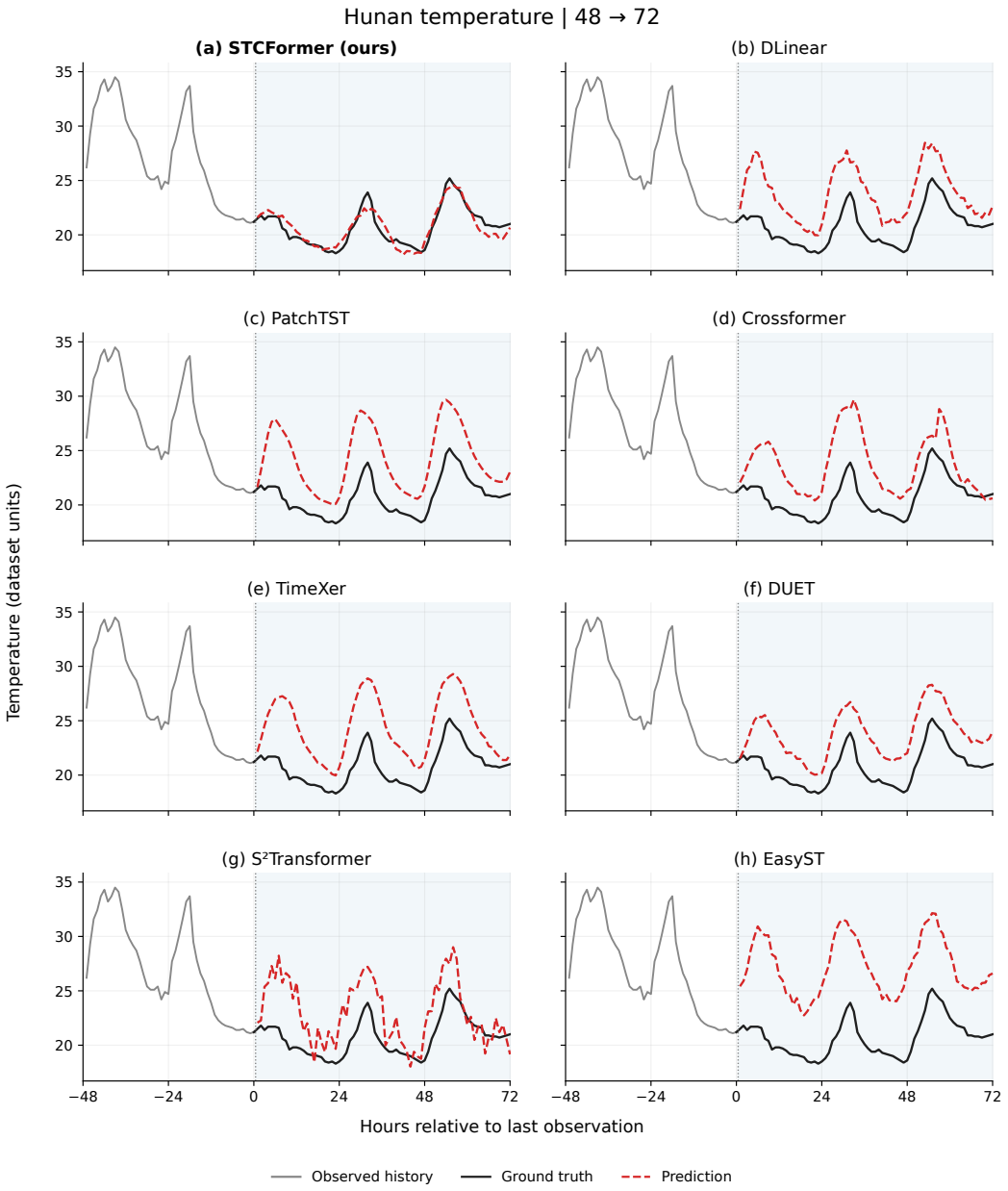


Figure 11: Prediction visualization on Hunan temperature with a 48-hour input and a 72-hour forecast horizon. The plotting conventions follow Figure 10. Values are shown in dataset units.

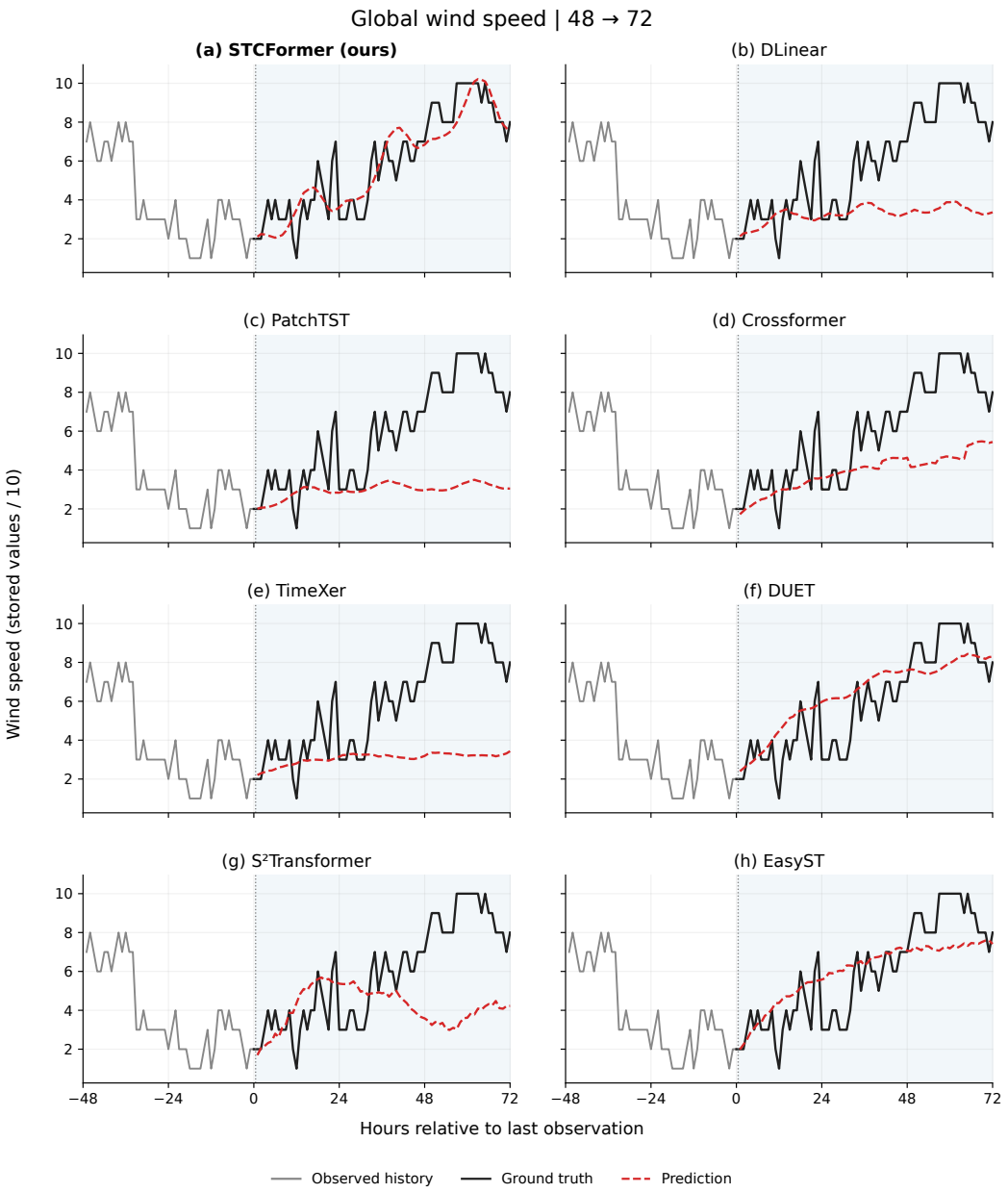


Figure 12: Prediction visualization on Global wind speed with a 48-hour input and a 72-hour forecast horizon. The plotting conventions follow Figure 10. Stored values are divided by 10 for display, consistently for observations and predictions.