
On the Integration of Spatial-Temporal Knowledge: A Lightweight Approach to Atmospheric Time Series Forecasting

Yisong Fu^{1,2}, Fei Wang^{1,2*}, Zezhi Shao¹, Boyu Diao^{1,2}, Lin Wu^{1,2},
Zhulin An^{1,2}, Chengqing Yu^{1,2}, Yujie Li^{1,2}, Yongjun Xu^{1,2*}

¹State Key Laboratory of AI Safety, Institute of Computing Technology,
Chinese Academy of Sciences ²University of Chinese Academy of Sciences
{fuyisong24s, wangfei, shaozezhi, diaoboyu, wulin}@ict.ac.cn,
{anzhulin, yuchengqing22b, liyujie23s, xyj}@ict.ac.cn

Abstract

Transformers have gained attention in atmospheric time series forecasting (ATSF) for their ability to capture global spatial-temporal correlations. However, their complex architectures lead to excessive parameter counts and extended training times, limiting their scalability to large-scale forecasting. In this paper, we revisit ATSF from a theoretical perspective of atmospheric dynamics and uncover a key insight: spatial-temporal position embedding (STPE) can inherently model spatial-temporal correlations even without attention mechanisms. Its effectiveness arises from the integration of geographical coordinates and temporal features, which are intrinsically linked to atmospheric dynamics. Based on this, we propose **STELLA**, a **S**patial-**T**emporal knowledge **E**mbodied **L**ightweight model **L** for **A**TSF, utilizing only STPE and an MLP architecture in place of Transformer layers. With *10k* parameters and one hour of training, STELLA achieves superior performance on five datasets compared to other advanced methods. The paper emphasizes the effectiveness of spatial-temporal knowledge integration over complex architectures, providing novel insights for ATSF. The code is available at <https://github.com/GestaltCogTeam/STELLA>.

1 Introduction

Atmospheric time series forecasting (ATSF), such as weather forecasting and air quality prediction, is of great significance in a wide variety of domains such as agriculture, energy, and economics. In recent decades, automatic weather stations have grown exponentially, becoming a cornerstone of modern meteorology [34]. These stations are cost-effective for applications [3, 36] and are ideally positioned to provide a large volume of data to advance deep learning (DL) approaches in ATSF [27].

However, the application of DL in ATSF faces two main challenges: (1) the observations of worldwide stations exhibit intricate spatial-temporal correlations [40], necessitating models with advanced mining capabilities to ensure accurate forecasting [52]. (2) The requirement for fine-grained and large-scale forecasting [25, 39] calls for highly efficient and scalable models.

These two challenges are often *trade-offs* in prior studies, as shown in Figure 1. Transformer and its variants, which have gained significant popularity in ATSF, utilize sophisticated architectures to capture global spatial-temporal correlations. For example, AirFormer [16] introduces dartboard attention for air quality prediction, and MRIformer [47] utilizes multi-resolution attention to predict wind

*Corresponding authors.

speed. However, these complex designs come with significant costs, including hundreds of millions of parameters and extended training times, which limit their scalability for large-scale forecasting and hinder their applicability, especially with limited computational resources [6]. Moreover, despite the increased complexity, the performance gains are quite limited and do not justify the trade-off for practical utility. This motivates us to rethink the bottleneck of ATSF.

To this end, we delve deeper into the physical principles of atmospheric dynamics. In the atmospheric system, the evolution of atmospheric variable ν can be described by a partial differential equation (PDE):

$$\frac{\partial \nu}{\partial t} = f(\nu, t, \lambda, \phi, z). \quad (1)$$

We demonstrate it theoretically in §3.5. The equation indicates that ν is directly influenced by time t and geographical coordinates λ, ϕ, z . However, most previous work has overlooked this spatial-temporal knowledge, simplistically treating f as a function of historical values while focusing on fitting it in more complex forms. This misguided direction has led to the key bottleneck of ATSF.

In this paper, we highlight the significance of spatial-temporal knowledge and introduce a spatial-temporal position embedding (STPE) to integrate geographical coordinates and temporal features from Eq.(1). Although position embeddings are widely considered as an adjunct to permutation-invariant attention mechanisms, we demonstrate that **STPE can inherently model spatial-temporal correlations even in the absence of attention mechanisms, offering a 'free lunch' in balancing the performance-efficiency trade-off.**

Consequently, we propose **STELLA**, a **S**patial-**T**emporal knowledge **E**mbodied **L**ightweight model **L** for ATSF. STELLA utilizes STPE and replaces the Transformer layers with a simple MLP. Figure 1 shows STELLA’s lead in both performance and efficiency. With only 10k parameters and one hour of training, STELLA achieves competitive performance against 17 baselines. Furthermore, it is noteworthy that the computational complexity of STELLA grows linearly with the increase of the number of stations N and the parameter count is independent of N . Therefore, STELLA can efficiently scale to the data with a larger N , facilitating large-scale forecasting. STELLA’s leading-edge performance and efficiency challenge the prevailing assumptions that ATSF necessitates complex architecture (e.g., Transformers, STGNNs), offering a balanced solution for ATSF. Our contributions can be summarized as follows.

- We innovatively highlight the significance of STPE in Transformer-based ATSF models. Even without attention mechanisms, it can explicitly capture spatial-temporal correlations by integrating spatial-temporal knowledge into the model. We theoretically prove its effectiveness from the perspective of atmospheric dynamics. Furthermore, STPE can also be applied to other models to improve performance (§4.5).
- We propose STELLA, utilizing the STPE and replacing the Transformer layers with a simple MLP. To the best of our knowledge, it is the first lightweight model designed for ATSF, challenging the prevailing assumption that ATSF necessitates complex architecture.
- STELLA offers a performance-efficiency balanced solution for ATSF. Extensive experiments across five datasets demonstrate the superior performance of STELLA over 17 baselines.

2 Related Work

2.1 DL in Atmospheric Time Series Forecasting

Atmospheric time series forecasting (ATSF), including applications such as weather forecasting and air quality prediction, involves predicting key atmospheric variables (e.g., temperature) collected from

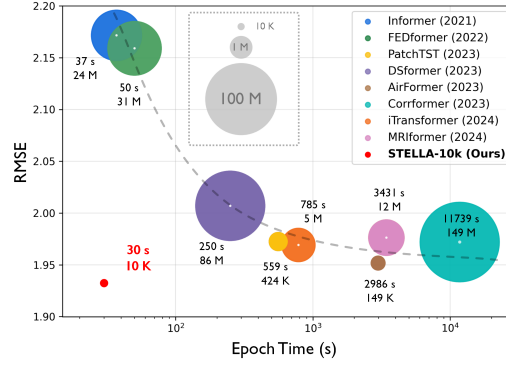


Figure 1: Performance-efficiency comparison on the GlobalWind dataset. A performance-efficiency trade-off can be observed in Transformers, while STELLA leads in both aspects. The area of the plot represents the parameter count of the model.

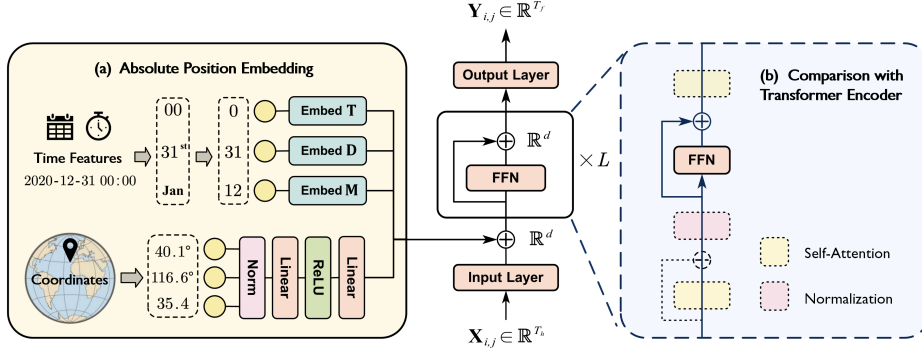


Figure 2: Architecture of STELLA.

weather stations over time. Spatial-temporal graph neural networks (STGNNs) prove effective in modeling spatial-temporal patterns across global stations [17, 22, 32, 1, 41, 28, 44], but most models are limited to short-term forecasting due to their high computational complexity, which restricts their scalability. Recently, Transformer-based models have gained popularity for their ability to capture global spatial-temporal correlations. For example, AirFormer [16] introduces dartboard attention to model spatial correlations in air quality prediction, while Corrformer [40] uses a multi-correlation mechanism as a substitute for attention in weather forecasting. MGSFformer [46] and MRIfomer [47] employ attention to capture multi-resolution correlations through downsampling, aiming to forecast wind speed and air quality. However, the complexity of these models introduces significant computational costs, limiting their practical applicability and scalability for large-scale forecasting.

2.2 Studies of the effectiveness of Transformers

The effectiveness of Transformers has been thoroughly discussed in the fields of computer vision (CV) [48, 19] and natural language processing (NLP) [4]. In time series forecasting (TSF), LSTF-Linear [49] pioneered the exploration and outperformed a variety of Transformer-based methods with a linear model. Shao et al. [30] posited that Transformer-based models face an over-fitting problem on specific datasets. Some recent works further questioned the necessity of attention in Transformers for TSF and replaced the attention with other modules. For example, MTS-Mixers [14] attempt to use random matrices and factorized MLPs instead of attention for information aggregation. MEAformer [10] replaces conventional attention with a linear-complexity mixing module. Additionally, SOFTS [7] employs STAR module as a substitute for attention, which aggregates all series into a global core representation. These studies consider PE as supplementary to attention mechanisms and consequently remove it along with attention, yet none have recognized the importance of PE.

3 Methodology

3.1 Problem Definition

Atmospheric Time Series Forecasting. We consider N stations and each station collects C time series of atmospheric variables (e.g., temperature). Then the observed data at time t can be denoted as $\mathbf{X}_t \in \mathbb{R}^{N \times C}$. The 3D geographical coordinates of stations are organized as a matrix $\Sigma \in \mathbb{R}^{3 \times N}$, which is naturally accessible in station-based forecasting. Given the historical time series of all stations from the past T_h time steps and optional spatial and temporal information, we aim to learn a function $\mathcal{F}_\theta(\cdot)$ to forecast the values of future T_f time steps :

$$\mathbf{Y}_{t:t+T_f} = \mathcal{F}_\theta(\mathbf{X}_{t-T_h:t}; \Sigma, t), \quad (2)$$

where $\mathbf{X}_{t-T_h:t} \in \mathbb{R}^{T_h \times N \times C}$ is the historical data, and $\mathbf{Y}_{t:t+T_f} \in \mathbb{R}^{T_f \times N \times C}$ is the future data.

3.2 Overview of STELLA

As shown in Figure 2, STELLA consists of an STPE module (Figure 2a) and an encoder that retains only the Feed-Forward Network (FFN) and discards all other components (Figure 2b). This

design significantly enhances computational and memory efficiency while maintaining competitive performance relative to more complex architectures. Our choice to pair STPE with a simple MLP architecture was driven by a key motivation: to isolate and demonstrate the standalone power of STPE in spatial-temporal modeling. This minimalistic yet effective design not only reduces computational overhead but also offers clear interpretability regarding the contribution of spatial-temporal knowledge in ATSF.

3.3 Spatial-Temporal Position Embedding

Position embedding encodes the positional information of tokens in a sequence [38] and is widely regarded as an auxiliary component for permutation-invariant attention mechanisms. However, we introduce spatial-temporal position embedding (STPE) to integrate geographical coordinates and temporal features into the model, demonstrating that it can inherently capture spatial-temporal correlations by embedding additional spatial-temporal knowledge. Specifically, STPE consists of two components: spatial embedding and temporal embedding.

Spatial Embedding. The spatial embedding provides the geographical coordinates of stations to the model, which can explicitly model spatial correlations among worldwide stations. Specifically, we encode the geographical coordinates of the station into latent space. First, to account for the differing ranges of coordinate values, we perform normalization on each coordinate independently. Then, we utilize a feed-forward network (FFN). Therefore, the spatial embedding $\mathbf{SE}^i \in \mathbb{R}^d$ can be denoted as:

$$\mathbf{SE}^i = \text{FFN}(\boldsymbol{\Sigma}^i) = \mathbf{W}_2 \text{ReLU}(\mathbf{W}_1 \boldsymbol{\Sigma}^i + b_1) + b_2, \quad (3)$$

where $\boldsymbol{\Sigma}^i \in \mathbb{R}^3$ represents the normalized coordinates of station i .

Temporal Embedding. Temporal embedding provides real-world temporal knowledge to the model. We utilize three learnable embedding matrices $\mathbf{T} \in \mathbb{R}^{24 \times d}$, $\mathbf{D} \in \mathbb{R}^{31 \times d}$ and $\mathbf{M} \in \mathbb{R}^{12 \times d}$ to save the temporal embeddings of all time steps [31]. They represent the patterns of weather in three scales ($\mathbf{T}$ denotes hours in a day, $\mathbf{D}$ denotes days in a month and $\mathbf{M}$ denotes the months in a year), contributing to modeling the multi-scale temporal correlations of atmospheric states. We add them together to obtain temporal embedding $\mathbf{TE}_t$:

$$\mathbf{TE}_t = \mathbf{T}_t + \mathbf{D}_t + \mathbf{M}_t. \quad (4)$$

3.4 MLP Backbone

Input Layer. Let $\mathbf{X}^{i,j} \in \mathbb{R}^{T_h}$ be the historical time series of station i and variable j . $\mathbf{X}^{i,j}$ is mapped by the input embedding layer to $\mathbf{H}^{i,j} \in \mathbb{R}^d$ in latent space, then added to the spatial and temporal embedding to obtain $\mathbf{E}_t^{i,j}$:

$$\begin{aligned} \mathbf{H}^{i,j} &= \text{Linear}(\mathbf{X}^{i,j}), \\ \mathbf{E}_t^{i,j} &= \mathbf{H}^{i,j} + \mathbf{SE}^i + \mathbf{TE}_t. \end{aligned} \quad (5)$$

Encoder. We utilize an L -layer MLP as encoder to learn the representation $\mathbf{Z}^{i,j}$ from the embedded data $\mathbf{E}_t^{i,j}$. Let $(\mathbf{Z}^{i,j})^0 = \mathbf{E}_t^{i,j}$, and the l -th MLP layer with residual connection can be denoted as:

$$(\mathbf{Z}^{i,j})^{l+1} = \text{FFN}^l((\mathbf{Z}^{i,j})^l) + (\mathbf{Z}^{i,j})^l. \quad (6)$$

Output Layer. We employ a linear layer to map the representation $\mathbf{Z} \in \mathbb{R}^{d \times N \times C}$ to the specified dimension, generating the prediction $\hat{\mathbf{Y}} \in \mathbb{R}^{T_f \times N \times C}$.

3.5 Theoretical Analysis

In this part, we provide a theoretical analysis of STELLA, focusing on its effectiveness and efficiency.

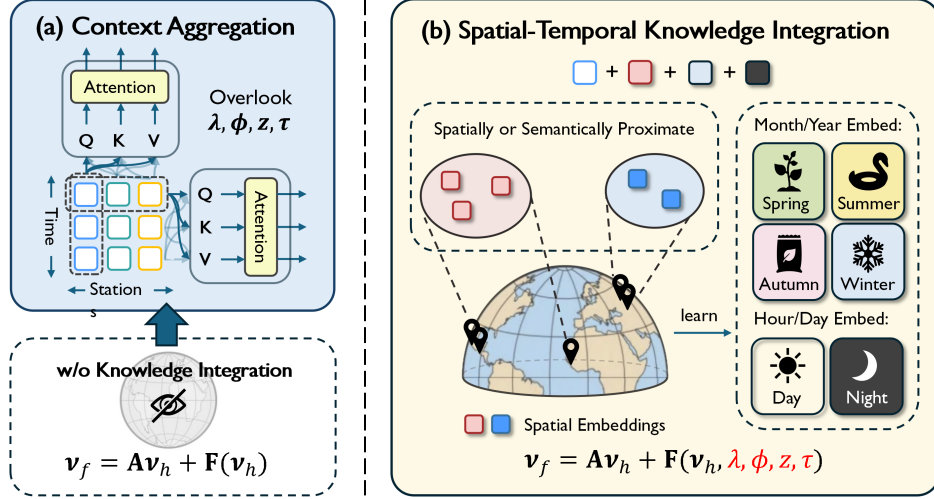


Figure 3: The key distinction that enhances the effectiveness of STELLA compared to prior approaches. (a) Most prior methods primarily rely on the context aggregation of historical observations (vanilla attention as an example). (b) In contrast, STELLA makes predictions guided by spatial and temporal knowledge. For clarity, history data is denoted as $\mathbf{v}_h$, future data as $\mathbf{v}_f$, and the day-in-month embedding is omitted.

Effectiveness of STELLA. The effectiveness of STELLA lies in the fact that STPE integrates geographical coordinates and real-world temporal knowledge into the model, which are intrinsically linked to atmospheric dynamics. In the following, we theoretically demonstrate this relationship.

Theorem 1. Let $\{\lambda, \phi, z\}$ be the longitude, latitude, and altitude of a weather station and ν be a meteorological variable collected by the station, then the time evolution of ν is a function of ν , time t and coordinate λ, ϕ, z :

$$\frac{\partial \nu}{\partial t} = f(\nu, \lambda, \phi, z, t). \quad (7)$$

Proof. We provide proof with zonal wind speed as an example. See Appendix A.1 for the full proof. According to the fundamental equations of atmospheric dynamics, the wind speed $\mathbf{V}$ satisfies:

$$\frac{d\mathbf{V}}{dt} = -\frac{1}{\rho} \nabla p - 2\boldsymbol{\Omega} \times \mathbf{V} + \mathbf{g} + \mathbf{F}, \quad (8)$$

where p is the pressure, ρ is the air density, and other terms are constants. We can transform the equation into spherical coordinates and apply the thin-layer approximation. The zonal wind speed u can be expressed as:

$$\frac{du}{dt} = -\frac{1}{\rho a \cos \phi} \frac{\partial p}{\partial \lambda} + f v + \frac{uv \tan \phi}{a} + F_\lambda, \quad (9)$$

where $\frac{d}{dt} = \frac{\partial}{\partial t} + u \frac{\partial}{\partial \lambda} + v \frac{\partial}{\partial \phi} + w \frac{\partial}{\partial z}$, a is the Earth's radius. It is possible to render the left side of the equation spatially independent by rearranging terms:

$$\frac{\partial u}{\partial t} = -\left(u \frac{\partial u}{\partial \lambda} + v \frac{\partial u}{\partial \phi} + w \frac{\partial u}{\partial z}\right) - \frac{1}{\rho a \cos \phi} \frac{\partial p}{\partial \lambda} + f v + \frac{uv \tan \phi}{a} + F_\lambda. \quad (10)$$

Therefore, we have

$$\frac{\partial u}{\partial t} = f(u, \lambda, \phi, z, t). \quad (11)$$

□

Considering using historical data spanning T_h steps to predict future T_f steps, we can derive the following corollary:

Corollary 2.

$$\boldsymbol{\nu}_{\tau+1:\tau+T_f} = \mathbf{A}\boldsymbol{\nu}_{\tau-T_h+1:\tau} + \mathbf{F}(\boldsymbol{\nu}_{\tau-T_h+1:\tau}, \lambda, \phi, z, \tau), \quad (12)$$

where $\boldsymbol{\nu}_{\tau-T_h+1:\tau}$ is the historical data, $\boldsymbol{\nu}_{\tau+1:\tau+T_f}$ is the future data, and $\|\mathbf{A}\|_\infty = 1$.

The detailed proof is provided in Appendix A.2. According to Eq.(12), we can employ a neural network $\mathcal{F}_\theta$ to approximate $\mathbf{F}(\boldsymbol{\nu}, \lambda, \phi, z, \tau)$. However, most previous models, including Transformers and STGNNs, have overlooked the critical spatial-temporal factors λ, ϕ, z, τ , instead treating $\mathbf{F}$ simplistically as a function of historical values, as shown in Figure 3 (a). These studies introduce increasingly complex context aggregation methods in an attempt to better fit historical data. However, blindly guessing spatial-temporal correlations by solely aggregating historical context may lead to overfitting, which becomes the key bottleneck for these models.

In comparison, STELLA can explicitly model $\mathbf{F}$ with λ, ϕ, z, τ introduced by STPE, as shown in Figure 3 (b). STELLA aims to learn similar STPEs for spatially or semantically proximate stations and time steps, thereby capturing spatial-temporal correlations. Specifically, the spatial embedding learns the climatic features of different locations, while the temporal embedding captures the periodicity and seasonality of atmospheric states. We demonstrate this through visualization in §4.6. Therefore, STPE serves as an effective method for modeling spatial-temporal correlations.

Efficiency of STELLA. We theoretically analyze the efficiency of STELLA from the perspectives of parameter volume and computational complexity.

Theorem 3. *The total number of parameters required for STELLA is*

$$\underbrace{2(d+1)dL + (T_h + T_f + 2)d}_{MLP} + \underbrace{(d+72)d}_{STPE}. \quad (13)$$

The proof is provided in Appendix A.3. According to **Theorem 3**, the parameter count of STELLA is independent of the number of stations N . Therefore, a key advantage of STELLA is that its deployment overhead remains static regardless of how many stations it serves, making it particularly suitable for resource-constrained environments. In addition, the computational complexity of STELLA grows linearly with N . In contrast, context aggregation causes quadratic complexity and $\mathcal{O}(N)$ parameters, which are unaffordable with large-scale stations. Therefore, STELLA efficiently models spatial correlations with N -independent parameters and linear complexity, ensuring optimal scalability for large-scale data.

4 Experiments

4.1 Experimental Setup

Datasets. We conduct experiments on five real-world datasets: (1) **GlobalWind and GlobalTemp** [40] comprise the hourly averaged wind speed and temperature of 3,850 stations on a global scale, spanning two years. Following the prior work [40], we use the past 48 hours to predict the next 24 hours for short-term weather forecasting. (2) **ChinaWind and ChinaTemp** comprise the daily averaged wind speed and temperature of 396 stations in China, spanning 10 years. We use the past 60 days to predict the next 30 days, addressing long-term weather forecasting. (3) **China-PM2.5** comprises the hourly averaged wind speed of 1,316 stations in China, spanning five years. We use the past 72 hours to predict the next 72 hours for mid-term air quality prediction. See Appendix C.1 for more details of the datasets.

Baselines. We compare our STELLA with the following five categories of baselines. (1) *Classic methods*: HI [5], ARIMA [33]. (2) *STGNNs*: AGCRN [1], MTGNN [41], GTS [28]. (3) *Transformer-based TSF methods*: Informer [53], FEDformer [54], DSformer [45], PatchTST [23], iTransformer [20], DUET [26]. (3) *lightweight TSF methods*: N-BEATS [24], DLinear [49], FITS [43]. (4) *ATSF specialized Transformers*: AirFormer [16], Corrformer [40], MRIfomer [47]. See Appendix C.2 for a detailed introduction to the baselines.

Table 1: Weather forecasting results on 5 datasets. The best results are in **bold** and the second best results are underlined. Dashes denote the out-of-memory (OOM) errors.

Methods	GlobalWind		GlobalTemp		ChinaWind		ChinaTemp		ChinaPM2.5	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
Classic Methods										
HI	2.697	1.831	3.859	2.575	9.851	6.751	7.630	5.834	37.11	20.29
ARIMA	2.116	1.539	4.575	3.267	7.947	5.795	5.396	4.026	38.66	20.40
STGNNs										
AGCRN	-	-	-	-	7.458	5.061	4.336	2.999	-	-
MTGNN	-	-	-	-	7.294	5.055	4.221	3.168	28.21	16.12
GTS	-	-	-	-	7.312	4.997	4.298	3.082	-	-
Transformer-Based TSF Methods										
Informer	2.172	1.496	5.770	4.415	7.832	5.279	4.477	3.045	28.85	16.04
FEDformer	2.159	1.471	3.324	2.405	7.334	5.051	4.665	3.455	28.75	16.14
DSformer	2.007	1.347	3.089	2.057	7.311	5.000	4.919	3.605	27.68	14.94
PatchTST	1.973	1.332	3.130	2.062	7.295	5.033	4.909	3.600	26.87	14.37
iTransformer	1.969	1.314	2.950	1.883	7.248	4.995	4.292	3.193	26.82	14.40
DUET	1.946	1.318	3.072	2.042	7.278	5.029	4.480	3.048	26.61	14.41
MLP-Based TSF Methods										
N-BEATS	2.031	1.390	3.034	2.117	7.297	4.998	4.791	3.486	26.48	14.44
DLinear	2.005	1.350	3.149	2.072	7.309	5.031	4.990	3.659	27.60	14.76
FITS	2.021	1.354	3.150	2.072	7.284	5.039	5.283	3.823	27.73	14.96
ATSF Specialized Methods										
AirFormer	1.952	1.314	5.594	4.127	7.909	5.377	4.772	3.561	29.63	15.55
Corrformer	1.972	1.304	2.777	1.888	7.224	4.950	4.728	3.377	28.07	15.28
MRIfomer	1.976	1.318	3.085	1.999	7.264	4.993	5.132	3.678	26.99	14.39
STELLA-10k	<u>1.933</u>	<u>1.294</u>	3.041	2.009	<u>7.118</u>	<u>4.876</u>	<u>4.167</u>	3.003	<u>26.32</u>	<u>14.08</u>
	± 0.001	± 0.002	± 0.002	± 0.002	± 0.003	± 0.001	± 0.002	± 0.003	± 0.003	± 0.002
STELLA _{opt}	1.919	1.284	2.724	1.858	7.104	4.869	4.112	2.975	26.15	13.85
	± 0.002	± 0.002	± 0.003	± 0.002	± 0.010	± 0.001	± 0.003	± 0.005	± 0.004	± 0.001

Implementation Details. We develop STELLA-10k which has approximately 10k parameters. The number of MLP layers is 2, and the hidden dimension is 32. To explore the full performance of the model, we also conducted hyperparameter research on different datasets to find the optimal configuration, denoted as STELLA_{opt}. Detailed configurations are provided in Appendix C.4. We adopt the Adam optimizer [11] to train our model and the learning rate is set to 5e-4. We trained all baselines with MAE (Mean Absolute Error) loss [30, 15], and the results of all baselines are obtained using the best hyperparameters through hyperparameter search. We evaluate the performance of all baselines using two commonly used metrics: MAE and RMSE (Root Mean Square Error). All models are implemented with PyTorch 2.3.1 and trained on an NVIDIA GeForce RTX 4090 24GB GPU and an Intel Xeon Gold 6330 CPU.

4.2 Main Results

Table 1 presents the results of the performance comparison between STELLA and other baselines on all datasets. The results of STELLA are averaged over 5 runs, with the standard deviation included. It is evident that the performance of lightweight TSF methods, such as DLinear and N-BEATS, is not satisfactory. This indicates that lightweight TSF models, which fail to capture spatial-temporal correlations, are inadequate for large-scale prediction tasks. STGNNs suffer from OOM errors due to their high computational complexity, while the performance is also suboptimal. Similarly, despite the complex architectures of Transformer-based models, most of them exhibit limited performance.

In contrast, STELLA-10k achieves competitive performance with a simple MLP architecture and only 10k parameters, while STELLA_{opt} consistently outperforms all other baselines on five datasets and three ATSF tasks. This suggests that integrating spatial-temporal information can significantly enhance model performance, proving to be more effective than the complex architectures of Transformer-based models. In addition, we provide a comparative analysis between STELLA and numerical weather prediction (NWP) methods in Appendix D.1.

4.3 Efficiency Analysis

Figure 1 illustrates the performance-efficiency comparison with Transformers. Here we further compare STELLA and other baselines, evaluating parameter counts, epoch time, and GPU memory usage [18]. Experiments are conducted on the most challenging GlobalWind dataset. As shown in Table 2, STELLA surpasses other DL methods in terms of all three efficiency metrics. When compared to the ATSF specialized methods, STELLA demonstrates an order-of-magnitude improvement across all three efficiency metrics, being about $10\times$ to $10,000\times$ smaller, $100\times$ to $300\times$ faster, and $50\times$ memory-efficient, respectively. Additionally, due to its compact parameter size and simple computations, STELLA can be efficiently trained in a CPU environment. It requires only 141 seconds to train STELLA-10k for an epoch, making it well-suited for environments with limited computational resources. See Appendix D.2 for detailed efficiency experiments under limited resources.

Table 2: Efficiency metrics of STELLA and other Transformer-based methods on GlobalWind.

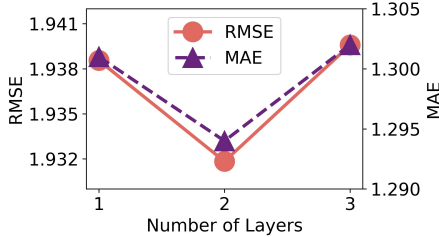
METHODS	PARAMS	EPOCH TIME	MAX MEM.
Informer	23.94M	37s	1.39GB
FEDformer	31.07M	50s	1.63GB
DSformer	85.99M	250s	13.6GB
PatchTST	424.1K	559s	19.22GB
iTransformer	4.55M	785s	16.61GB
AirFormer	148.7K	2986s	14.01GB
Corrformer	148.7M	11739s	18.41GB
MRFormer	11.66M	3431s	12.69GB
STELLA-10k	9.98K	30s (141s CPU)	792MB

4.4 Ablation Study

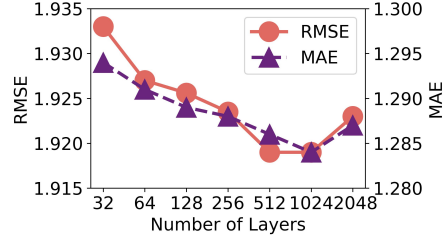
Effects of STPE. STPE is the key component of STELLA. To study its effects, we first conduct experiments on models with the spatial embedding and temporal embedding removed separately. Table 3 reveals that removing either embedding component leads to a decrease in MSE. This indicates that both spatial and temporal embeddings contribute positively to model performance. Next, we compare relative position embedding (RPE) with STPE. Specifically, RPE embeds the indices of stations instead of the absolute geographical coordinates. Since we project the temporal dimension into the hidden space, temporal RPE is unnecessary. The results are presented in Table 3. Although RPE introduces Nd parameters, significantly increasing the model size, its performance still falls short of that of STPE, further validating its effectiveness of STPE.

Table 3: Ablation results of STELLA on five datasets.

Methods	GlobalWind		GlobalTemp		ChinaWind		ChinaTemp		ChinaPM2.5	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
w/o Spatial Embedding	1.962	1.329	2.821	1.921	7.199	4.953	4.798	3.506	26.32	13.95
w/o Temporal Embedding	1.963	1.327	2.812	1.918	7.209	4.952	4.357	3.163	26.63	14.08
RPE	1.960	1.341	2.900	1.984	7.197	4.925	4.555	3.340	26.50	13.96
STELLA	1.919	1.284	2.724	1.858	7.104	4.869	4.112	2.975	26.15	13.85



(a)



(b)

Figure 4: Results of hyperparameters analysis on GlobalWind dataset. (a) Effects of the number of layers ($d = 64$). (b) Effects of the hidden dimension ($L = 2$).

Hyperparameter Study. We investigate the effects of two important hyperparameters: the number of layers L in the MLP and the hidden dimension d . As illustrated in Figure 4 (a), STELLA achieves

Table 4: Improvements obtained by the adoption of STPE.

Datasets		GlobalWind		GlobalTemp		ChinaWind		ChinaTemp	
Metric		RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
PatchTST	Original	1.973	1.332	3.130	2.062	7.295	5.033	4.909	3.600
	+STPE	1.947	1.307	3.077	1.994	7.196	4.978	4.490	3.303
DSformer	Original	2.007	1.347	3.089	2.062	7.311	5.000	4.919	3.605
	+ STPE	1.982	1.325	3.044	2.020	7.243	4.950	4.762	3.416
iTransformer	Original	1.969	1.314	2.950	1.883	7.248	4.995	4.292	3.193
	+STPE	1.917	1.279	2.855	1.804	7.104	4.914	4.188	3.600

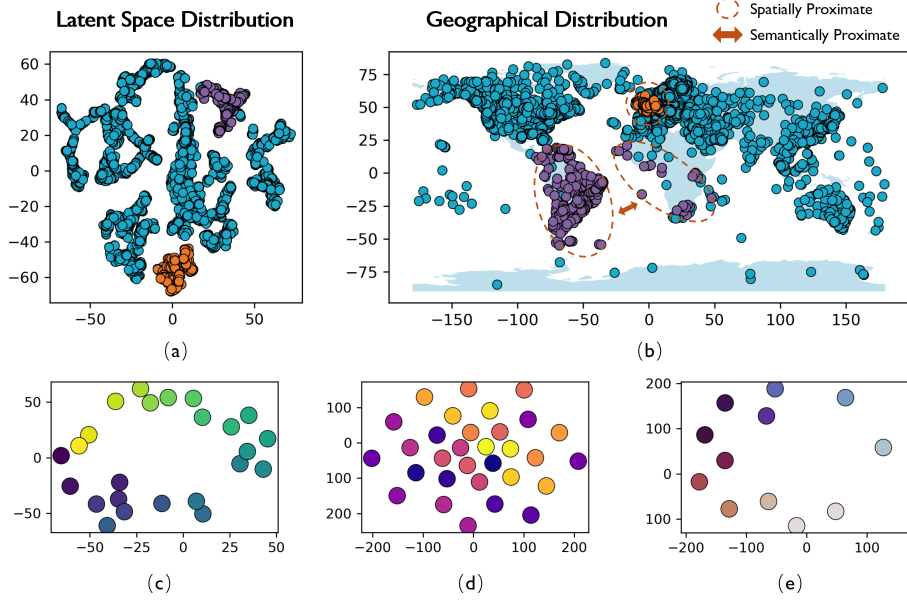


Figure 5: Visualization of STPE. (a) Spatial embedding in the 2D latent space. (b) Geographical distribution of stations. The model learns spatial and semantic similarity through spatial embedding. (c) Hour-in-day embedding T . (d) Day-in-month embedding D . (e) Month-in-year embedding M .

the best performance when $L = 2$, whereas an increase in L beyond 2 results in over-fitting and a consequent decline in model performance. Figure 4 (b) shows that the metrics decrease as the hidden dimension increases and begin to converge when d exceeds 1024. However, STELLA-10k ($d = 32$) already outperforms other Transformer-based models. This further substantiates that STPE is more effective than the complex architectures of Transformer-based models.

4.5 Generalization of Spatial-Temporal Position Embedding

In this section, we further evaluate the effects of STPE by applying it to Transformer-based models, with the results reported in Table 4. Only Transformers that independently embed each channel are compatible with STPE (Informer, FEDformer, etc., are incompatible), as our approach generates spatial embeddings for each station individually. The result shows that STPE can significantly enhance the performance of Transformers, enabling them to achieve satisfactory results. In particular, iTransformer achieves state-of-the-art performance on GlobalWind after the application of STPE.

4.6 Visualization of PE

In this section, we visualize the STPE to further study its effectiveness. Due to the high dimensionality of the embeddings, we employ t-SNE [37] to visualize them on 2D planes.

Visualization of Spatial Embedding. Figure 5 (a) indicates that spatial embeddings tend to cluster. The model attempts to learn similar embeddings for spatially or semantically proximate stations, resulting in a clustered structure. To demonstrate this, we have marked two clusters with orange and purple and examined the geographical distribution of the stations within each cluster. As shown in Figure 5 (b), the orange cluster is densely distributed in Europe in the geographical space, indicating that the model has learned spatial similarity. Meanwhile, stations in the purple cluster are distributed across South America and Africa, with all distribution areas characterized by tropical or subtropical climates, suggesting that the model has captured semantic similarity.

Visualization of Temporal Embedding. Figure 5 (c-e) shows the temporal embeddings with colors representing the temporal order. The hour-in-day and month-in-year embeddings form ring-like structures in temporal order, revealing the distinct daily and annual periodicities of weather, which is consistent with humans’ common understanding.

4.7 Case Study

To comprehensively illustrate STELLA’s capability to capture the complex spatial-temporal correlations among large-scale stations, we present a visualization of the forecasting results of the Global-Wind dataset from a multi-station perspective. Additional illustrative show-cases are available in Appendix F. Kriging [13] is employed to interpolate discrete points into a continuous surface, enhancing the visual clarity of spatial variations. As shown in Figure 6, the predicted results (right column) are closely aligned with the ground-truth values (left column) across all displayed time steps. This high level of consistency confirms that STELLA effectively captures spatial-temporal patterns in global weather data and delivers accurate predictions.

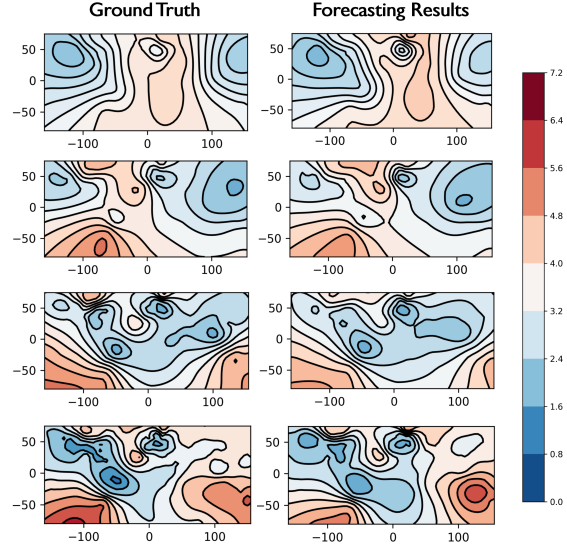


Figure 6: Forecasting results of averaged wind speed with a 6-hour interval and 5° (i.e., 64×32) resolution.

5 Conclusion

This work innovatively highlights the significance of STPE in Transformers for ATSF. Even without attention mechanisms, STPE explicitly captures spatial-temporal correlations by integrating geographical coordinates and temporal features, which are inherently linked to atmospheric dynamics. We then present STELLA, a lightweight and effective model for ATSF. We leverage STPE and replace Transformer layers with a simple MLP. STELLA can achieve satisfactory performance across five weather datasets. The paper posits that the incorporation of spatial-temporal knowledge is more effective than intricate model architectures, illuminating novel insights for ATSF.

Limitations and Future Work. STELLA is limited by its channel-independent modeling and the restricted expressive power of MLPs, and it may produce overly smoothed forecasts for extreme events, which is a common issue with MSE/MAE-trained models. Future work will address these challenges by incorporating covariates and physical constraints. See Appendix G for more details.

Acknowledgement

This work is supported by the NSFC under Grant Nos.62372430 and 62502505, the Youth Innovation Promotion Association CAS No.2023112, the Postdoctoral Fellowship Program of CPSF under Grant Number GZC20251078, the China Postdoctoral Science Foundation No.2025M77154 and HUA Innovation fundings. We sincerely thank all the anonymous reviewers who generously contributed their time and efforts.

References

- [1] Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in neural information processing systems*, 33:17804–17815, 2020.
- [2] Peter Bauer, Alan Thorpe, and Gilbert Brunet. The quiet revolution of numerical weather prediction. *Nature*, 525(7567):47–55, 2015.
- [3] Gabriel F.L.R. Bernardes, Rogério Ishibashi, André A.S. Ivo, Valério Rosset, and Bruno Y.L. Kimura. Prototyping low-cost automatic weather stations for natural disaster monitoring. *Digital Communications and Networks*, 9(4):941–956, 2023.
- [4] Yuchen Bian, Jiaji Huang, Xingyu Cai, Jiahong Yuan, and Kenneth Church. On attention redundancy: A comprehensive study. In *Proceedings of the 2021 conference of the north american chapter of the association for computational linguistics: human language technologies*, pages 930–945, 2021.
- [5] Yue Cui, Jiandong Xie, and Kai Zheng. Historical inertia: A neglected but powerful baseline for long sequence time-series forecasting. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pages 2965–2969, 2021.
- [6] Jinliang Deng, Xuan Song, Ivor W Tsang, and Hui Xiong. Parsimony or capability? decomposition delivers both in long-term time series forecasting. *arXiv preprint arXiv:2401.11929*, 2024.
- [7] Lu Han, Xu-Yang Chen, Han-Jia Ye, and De-Chuan Zhan. Softs: Efficient multivariate time series forecasting with series-core fusion. *Advances in Neural Information Processing Systems*, 37:64145–64175, 2024.
- [8] Kethmi Hirushini Hettige, Jiahao Ji, Shili Xiang, Cheng Long, Gao Cong, and Jingyuan Wang. Airphynet: Harnessing physics-guided neural networks for air quality prediction. In *The Twelfth International Conference on Learning Representations*, 2024.
- [9] Jincan Huang, Yongjun Xu, Qi Wang, Qi Cheems Wang, Xingxing Liang, Fei Wang, Zhao Zhang, Wei Wei, Boxuan Zhang, Libo Huang, et al. Foundation models and intelligent decision-making: Progress, challenges, and perspectives. *The Innovation*, 2025.
- [10] Siyuan Huang, Yepeng Liu, Haoyi Cui, Fan Zhang, Jinjiang Li, Xiaofeng Zhang, Mingli Zhang, and Caiming Zhang. Meaformer: An all-mlp transformer with temporal external attention for long-term time series forecasting. *Information Sciences*, 669:120605, 2024.
- [11] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [12] Wenyuan Li, Zili Liu, Keyan Chen, Hao Chen, Shunlin Liang, Zhengxia Zou, and Zhenwei Shi. Deep-physinet: Bridging deep learning and atmospheric physics for accurate and continuous weather modeling, 2024.
- [13] Yujie Li, Zezhi Shao, Chengqing Yu, Tangwen Qian, Zhao Zhang, Yifan Du, Shaoming He, Fei Wang, and Yongjun Xu. Sta-gann: A valid and generalizable spatio-temporal kriging approach. *arXiv preprint arXiv:2508.16161*, 2025.
- [14] Zhe Li, Zhongwen Rao, Lujia Pan, and Zenglin Xu. Mts-mixers: Multivariate time series forecasting via factorized temporal and channel mixing. *arXiv preprint arXiv:2302.04501*, 2023.
- [15] Yubo Liang, Zezhi Shao, Fei Wang, Zhao Zhang, Tao Sun, and Yongjun Xu. Basics: An open source fair multivariate time series prediction benchmark. In *International symposium on benchmarking, measuring and optimization*, pages 87–101. Springer, 2022.
- [16] Yuxuan Liang, Yutong Xia, Songyu Ke, Yiwei Wang, Qingsong Wen, Junbo Zhang, Yu Zheng, and Roger Zimmermann. Airformer: Predicting nationwide air quality in china with transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 14329–14337, 2023.
- [17] Haitao Lin, Zhangyang Gao, Yongjie Xu, Lirong Wu, Ling Li, and Stan Z Li. Conditional local convolution for spatio-temporal meteorological forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 7470–7478, 2022.
- [18] Shengsheng Lin, Weiwei Lin, Wentai Wu, Haojun Chen, and Junjie Yang. Sparsetsf: Modeling long-term time series forecasting with* 1k* parameters. In *Forty-first International Conference on Machine Learning*, 2024.

- [19] Sihao Lin, Pumeng Lyu, Dongrui Liu, Tao Tang, Xiaodan Liang, Andy Song, and Xiaojun Chang. Mlp can be a good transformer learner. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19489–19498, 2024.
- [20] 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*, 2024.
- [21] Gurii Marchuk. *Numerical methods in weather prediction*. Elsevier, 2012.
- [22] Qingjian Ni, Yuhui Wang, and Yifei Fang. Ge-stdgn: a novel spatio-temporal weather prediction model based on graph evolution. *Applied Intelligence*, 52(7):7638–7652, 2022.
- [23] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*, 2023.
- [24] Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting. In *International Conference on Learning Representations*, 2020.
- [25] Tim Palmer. Climate forecasting: Build high-resolution global climate models. *Nature*, 515(7527):338–339, 2014.
- [26] 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*, pages 1185–1196, 2025.
- [27] Martin G Schultz, Clara Betancourt, Bing Gong, Felix Kleinert, Michael Langguth, Lukas Hubert Leufen, Amirpasha Mozaffari, and Scarlet Stadler. Can deep learning beat numerical weather prediction? *Philosophical Transactions of the Royal Society A*, 379(2194):20200097, 2021.
- [28] Chao Shang, Jie Chen, and Jinbo Bi. Discrete graph structure learning for forecasting multiple time series. In *International Conference on Learning Representations*, 2021.
- [29] Zezhi Shao, Yujie Li, Fei Wang, Chengqing Yu, Yisong Fu, Tangwen Qian, Bin Xu, Boyu Diao, Yongjun Xu, and Xueqi Cheng. Blast: Balanced sampling time series corpus for universal forecasting models. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '25, page 2502–2513, New York, NY, USA, 2025. Association for Computing Machinery.
- [30] Zezhi Shao, Fei Wang, Yongjun Xu, Wei Wei, Chengqing Yu, Zhao Zhang, Di Yao, Tao Sun, Guangyin Jin, Xin Cao, et al. Exploring progress in multivariate time series forecasting: Comprehensive benchmarking and heterogeneity analysis. *IEEE Transactions on Knowledge and Data Engineering*, 2024.
- [31] Zezhi Shao, Zhao Zhang, Fei Wang, Wei Wei, and Yongjun Xu. Spatial-temporal identity: A simple yet effective baseline for multivariate time series forecasting. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 4454–4458, 2022.
- [32] Zezhi Shao, Zhao Zhang, Fei Wang, and Yongjun Xu. Pre-training enhanced spatial-temporal graph neural network for multivariate time series forecasting. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pages 1567–1577, 2022.
- [33] Robert H Shumway, David S Stoffer, and David S Stoffer. *Time series analysis and its applications*, volume 3. Springer, 2000.
- [34] Dipak V Sose and Ajj D Sayyad. Weather monitoring station: a review. *Int. Journal of Engineering Research and Application*, 6(6):55–60, 2016.
- [35] Yanru Sun, Zongxia Xie, Dongyue Chen, Emadeldeen Eldele, and Qinghua Hu. Hierarchical classification auxiliary network for time series forecasting. *arXiv preprint arXiv:2405.18975*, 2024.
- [36] Sonam Tenzin, Satetha Siyang, Theerapat Pobkrut, and Teerakiat Kerdcharoen. Low cost weather station for climate-smart agriculture. In *2017 9th international conference on knowledge and smart technology (KST)*, pages 172–177. IEEE, 2017.
- [37] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

- [38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [39] Fei Wang, Di Yao, Yong Li, Tao Sun, and Zhao Zhang. Ai-enhanced spatial-temporal data-mining technology: New chance for next-generation urban computing. *The Innovation*, 4(2), 2023.
- [40] Haixu Wu, Hang Zhou, Mingsheng Long, and Jianmin Wang. Interpretable weather forecasting for worldwide stations with a unified deep model. *Nature Machine Intelligence*, 5(6):602–611, 2023.
- [41] 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 *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 753–763, 2020.
- [42] Yongjun Xu, Xin Liu, Xin Cao, Changping Huang, Enke Liu, Sen Qian, Xingchen Liu, Yanjun Wu, Fengliang Dong, Cheng-Wei Qiu, et al. Artificial intelligence: A powerful paradigm for scientific research. *The Innovation*, 2(4), 2021.
- [43] Zhijian Xu, Ailing Zeng, and Qiang Xu. Fits: Modeling time series with 10k parameters. In *The Twelfth International Conference on Learning Representations*, 2024.
- [44] Chengqing Yu, Fei Wang, Zezhi Shao, Tangwen Qian, Zhao Zhang, Wei Wei, Zhulin An, Qi Wang, and Yongjun Xu. Ginar+: A robust end-to-end framework for multivariate time series forecasting with missing values. *IEEE Transactions on Knowledge and Data Engineering*, 2025.
- [45] Chengqing Yu, Fei Wang, Zezhi Shao, Tao Sun, Lin Wu, and Yongjun Xu. Dsformer: A double sampling transformer for multivariate time series long-term prediction. In *Proceedings of the 32nd ACM international conference on information and knowledge management*, pages 3062–3072, 2023.
- [46] Chengqing Yu, Fei Wang, Yilun Wang, Zezhi Shao, Tao Sun, Di Yao, and Yongjun Xu. Mgsformer: A multi-granularity spatiotemporal fusion transformer for air quality prediction. *Information Fusion*, page 102607, 2024.
- [47] Chengqing Yu, Guangxi Yan, Chengming Yu, Xinwei Liu, and Xiwei Mi. Mriformer: A multi-resolution interactive transformer for wind speed multi-step prediction. *Information Sciences*, 661:120150, 2024.
- [48] Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10819–10829, 2022.
- [49] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 11121–11128, 2023.
- [50] Chi Zhang, Huaping Zhong, Kuan Zhang, Chengliang Chai, Rui Wang, Xinlin Zhuang, Tianyi Bai, Qiu Jiantao, Lei Cao, Ju Fan, et al. Harnessing diversity for important data selection in pretraining large language models. In *The Thirteenth International Conference on Learning Representations*, 2025.
- [51] Shihao Zhang, Linlin Yang, Michael Bi Mi, Xiaoxu Zheng, and Angela Yao. Improving deep regression with ordinal entropy. In *The Eleventh International Conference on Learning Representations*, 2023.
- [52] Tianjie Zhao, Sheng Wang, Chaojun Ouyang, Min Chen, Chenying Liu, Jin Zhang, Long Yu, Fei Wang, Yong Xie, Jun Li, Fang Wang, Sabine Grunwald, Bryan M. Wong, Fan Zhang, Zhen Qian, Yongjun Xu, Chengqing Yu, Wei Han, Tao Sun, Zezhi Shao, Tangwen Qian, et al. Artificial intelligence for geoscience: Progress, challenges, and perspectives. *The Innovation*, 5:100691, 2024.
- [53] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11106–11115, 2021.
- [54] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*, pages 27268–27286. PMLR, 2022.

A Theoretical Proofs

A.1 Full Proof of Theorem 1

Proof. The fundamental equations of atmospheric motion [21] are

$$\begin{cases} \frac{d\mathbf{V}}{dt} = -\frac{1}{\rho}\nabla p - 2\boldsymbol{\Omega} \times \mathbf{V} + \mathbf{g} + \mathbf{F}, \\ \frac{d\rho}{dt} + \rho\nabla \cdot \mathbf{V} = 0, \\ C_p \frac{dT}{dt} - \frac{1}{\rho} \frac{dp}{dt} = Q, \\ \frac{\partial q}{\partial t} + \mathbf{V} \cdot \nabla q = S_q, \\ p = \rho RT, \end{cases} \quad (14)$$

where $\mathbf{V}$ is the velocity, ρ is the air density, p is the pressure, T is the temperature, q is the specific humidity, and others are physical constants.

Expand the equations into scalar form and transform them into spherical coordinates, yielding the following:

$$\begin{cases} \frac{du}{dt} = -\frac{1}{\rho} \frac{\partial p}{r \cos \phi \partial \lambda} + fv + \frac{uv \tan \phi}{r} + F_\lambda, \\ \frac{dv}{dt} = -\frac{1}{\rho} \frac{\partial p}{r \partial \phi} - fu - \frac{u^2 \tan \phi}{r} + F_\phi, \\ \frac{dw}{dt} = -\frac{1}{\rho} \frac{\partial p}{\partial r} - g + F_r, \\ \frac{d\rho}{dt} = -\rho \left(\frac{1}{r \cos \phi} \frac{\partial u}{\partial \lambda} + \frac{\partial v}{r \partial \phi} + \frac{\partial w}{\partial r} + \frac{2w}{r} - \frac{v}{r} \tan \phi \right), \\ \frac{dT}{dt} = \frac{Q}{C_p} + \frac{1}{\rho C_p} \frac{dp}{dt}, \\ \frac{dq}{dt} = S_q, \\ p = \rho RT, \end{cases} \quad (15)$$

where u is zonal velocity, v is meridional velocity, and w is vertical velocity. The expansion of $\frac{d}{dt}$ is

$$\frac{d}{dt} = \frac{\partial}{\partial t} + u \frac{\partial}{r \cos \phi \partial \lambda} + v \frac{\partial}{r \partial \phi} + w \frac{\partial}{\partial r}. \quad (16)$$

Radial distance r can be further denoted as $r = a + z$, where a is Earth's radius and z is altitude. Since a is a constant and $a \gg z$, we have $\frac{\partial}{\partial r} = \frac{\partial}{\partial z}$ and we can approximate r with a . Then we render the left side of the equation spatial independent by rearranging terms:

$$\begin{cases}
\frac{\partial u}{\partial t} = -\frac{u}{a \cos \phi} \frac{\partial u}{\partial \lambda} - \frac{v}{a} \frac{\partial u}{\partial \phi} - w \frac{\partial u}{\partial z} - \frac{1}{\rho} \frac{\partial p}{a \cos \phi \partial \lambda} + f v + \frac{u v \tan \phi}{a} + F_\lambda, \\
\frac{\partial v}{\partial t} = -\frac{u}{a \cos \phi} \frac{\partial v}{\partial \lambda} - \frac{v}{a} \frac{\partial v}{\partial \phi} - w \frac{\partial v}{\partial z} - \frac{1}{\rho} \frac{\partial p}{a \partial \phi} - f u - \frac{u^2 \tan \phi}{a} + F_\phi, \\
\frac{\partial w}{\partial t} = -\frac{u}{a \cos \phi} \frac{\partial w}{\partial \lambda} - \frac{v}{a} \frac{\partial w}{\partial \phi} - w \frac{\partial w}{\partial z} - \frac{1}{\rho} \frac{\partial p}{\partial z} - g + F_r, \\
\frac{\partial \rho}{\partial t} = -\frac{u}{a \cos \phi} \frac{\partial \rho}{\partial \lambda} - \frac{v}{a} \frac{\partial \rho}{\partial \phi} - w \frac{\partial \rho}{\partial z} - \rho \left(\frac{1}{r \cos \phi} \frac{\partial u}{\partial \lambda} + \frac{\partial v}{r \partial \phi} + \frac{\partial w}{\partial r} + \frac{2w}{r} - \frac{v}{r} \tan \phi \right), \\
\frac{\partial T}{\partial t} = -\frac{u}{a \cos \phi} \frac{\partial T}{\partial \lambda} - \frac{v}{a} \frac{\partial T}{\partial \phi} - w \frac{\partial T}{\partial z} + \frac{1}{\rho C_p} \left(\frac{\partial p}{\partial t} + \frac{u}{a \cos \phi} \frac{\partial p}{\partial \lambda} + \frac{v}{a} \frac{\partial p}{\partial \phi} + w \frac{\partial p}{\partial z} \right) + \frac{Q}{C_p}, \\
\frac{\partial q}{\partial t} = -\frac{u}{a \cos \phi} \frac{\partial q}{\partial \lambda} - \frac{v}{a} \frac{\partial q}{\partial \phi} - w \frac{\partial q}{\partial z} + S_q, \\
p = \rho R T.
\end{cases} \tag{17}$$

Therefore, for each atmospheric variable ν , we have

$$\frac{\partial \nu}{\partial t} = f(\nu, t, \lambda, \phi, z). \tag{18}$$

□

A.2 Proof of Corollary 2

Proof. According to **Theorem 1**, we can integrate both sides of the equation with respect to t , from time step τ_i to step τ_j :

$$\nu_{\tau_j} = \nu_{\tau_i} + \int_{\tau_i \Delta t}^{\tau_j \Delta t} f(\nu(\lambda, \phi, z, t), \lambda, \phi, z, t) dt, \tag{19}$$

where Δt is the interval between time steps.

Before proceeding with the mathematical proof, we first illustrate with Figure 7. A directed edge from τ_i to τ_j in the figure denotes the evolution constrained by Eq.(19). Figure 7 (a) shows the mechanism that the state of the atmosphere evolves step by step, which we need to adopt an autoregressive neural network (e.g. RNN) to approximate it. Through a simple topological transformation, we can obtain the mechanism shown in (b), where the unobserved states are calculated by every historical observation. Therefore, we can adopt a neural network to predict all unobserved states in parallel and it constitutes a more robust approach as it fully leverages historical observations.

We provide a detailed mathematical proof in the following. For brevity, $\int_{\tau_i \Delta t}^{\tau_j \Delta t} F(\lambda, \phi, z, t) dt$ is denoted as I_i^j . For the unobserved data at step $\tau + k$ ($k = 1, 2, \dots, T_f$), it can be represented by:

$$\begin{cases}
\nu_{\tau+k} = \nu_\tau + I_\tau^{\tau+k}, \\
\nu_{\tau+k} = \nu_{\tau-1} + I_{\tau-1}^{\tau+k}, \\
\vdots \\
\nu_{\tau+k} = \nu_{\tau-T_h+1} + I_{\tau-T_h+1}^{\tau+k}.
\end{cases} \tag{20}$$

We can take its linear combination as follows:

$$\begin{aligned}
\nu_{\tau+k} &= \alpha_{k,1} (\nu_\tau + I_\tau^{\tau+k}) + \alpha_{k,2} (\nu_{\tau-1} + I_{\tau-1}^{\tau+k}) + \dots + \alpha_{k,T_h} (\nu_{\tau-T_h+1} + I_{\tau-T_h+1}^{\tau+k}) \\
&= \alpha_k (\nu_{\tau-T_h+1:\tau} + \mathbf{I}_{\tau-T_h+1:\tau}^{\tau+k}),
\end{aligned} \tag{21}$$

where $\mathbf{I}_{\tau-T_h+1:\tau}^{\tau+k}$ is $(I_{\tau-T_h+1}^{\tau+k}, I_{\tau-T_h}^{\tau+k}, \dots, I_\tau^{\tau+k})^\top$ and $\alpha_k \in \mathbb{R}^{T_h}$ satisfies $\|\alpha_k\| = 1$.

By repeating this procedure from $\nu_{\tau+1}$ to $\nu_{\tau+T_f}$, we have

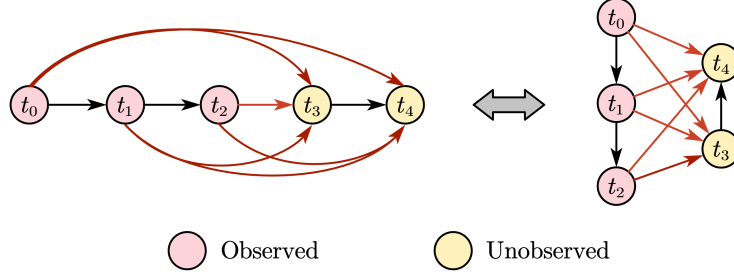


Figure 7: Mechanism of the evolution of the atmosphere state. (a) The atmosphere state evolves step by step through the black edges. (b) Predict the unobserved states in parallel through the red edges. (a) and (b) are topologically equivalent.

$$\begin{cases} \nu_{\tau+1} = \alpha_1 (\nu_{\tau-T_h+1:\tau} + \mathbf{I}_{\tau-T_h+1:\tau}^{\tau+1}), \\ \nu_{\tau+2} = \alpha_2 (\nu_{\tau-T_h+1:\tau} + \mathbf{I}_{\tau-T_h+1:\tau}^{\tau+2}), \\ \dots \\ \nu_{\tau+T_f} = \alpha_{T_f} (\nu_{\tau-T_h+1:\tau} + \mathbf{I}_{\tau-T_h+1:\tau}^{\tau+T_f}). \end{cases} \quad (22)$$

Therefore, we have

$$\nu_{\tau+1:\tau+T_f} = \mathbf{A} \nu_{\tau-T_h+1:\tau} + \mathbf{F}(\nu_{\tau-T_h+1:\tau}, \lambda, \phi, z, \tau), \quad (23)$$

where $\mathbf{A} = (\alpha_1, \alpha_2, \dots, \alpha_{T_f})^\top$ satisfies $\|\mathbf{A}\|_\infty = 1$ and $\mathbf{F}$ is the combination of $\mathbf{I}$.

□

A.3 Proof of Theorem 3

Proof. The data embedding layer maps the input data into the latent space with dimension d , thereby introducing $(T_h + 1)d$ parameters. Analogously, the regression layer introduces $(T_f + 1)d$ parameters. The parameter count of a L -layer MLP with residual connection is $2Ld(d + 1)$.

For the STPE module, the spatial embedding costs $(3 + 1)d + d(d + 1)$ parameters, and the temporal embedding costs $(24 + 31 + 12)d$ parameters.

Therefore, the total number of parameters required for STELLA is $2(d + 1)dL + (T_h + T_f + 2)d + (d + 72)d$. □

B Overall Workflow of STELLA

The overall workflow of STELLA is provided in Algorithm 1.

C Experimental Details

C.1 Dataset Description

To evaluate the comprehensive performance of the proposed model, we conduct experiments on five ATSF datasets with different temporal resolutions and spatial coverages including:

- **GlobalWind and GlobalTemp** [40] are collected from the National Centers for Environmental Information (NCEI)². These datasets contain the hourly averaged wind speed and temperature of 3,850 stations around the world, spanning two years from 1 January 2019 to

²<https://www.ncei.noaa.gov/data/global-hourly/access>

Algorithm 1 Overall workflow of STELLA.

```
1: INPUT: historical data  $\mathbf{X} \in \mathbb{R}^{T_h \times N \times C}$ , geographical coordinates  $\Sigma \in \mathbb{R}^{N \times 3}$ , the first time step  $t$ 
2: OUTPUT: forecasting result  $\mathbf{Y} \in \mathbb{R}^{T_f \times N \times C}$ 
3:  $\mathbf{X} = \mathbf{X}.\text{transpose}(1, -1)$  /*  $\mathbf{X} \in \mathbb{R}^{C \times N \times T_h}$  */
4:  $\mathbf{H} = \text{Linear}(\mathbf{X})$  /* Input layer,  $\mathbf{H} \in \mathbb{R}^{C \times N \times d}$  */
5:  $\mathbf{SE} = \text{FFN}(\Sigma)$  /*  $\mathbf{SE} \in \mathbb{R}^{N \times d}$  */
6:  $\mathbf{SE} = \mathbf{SE}.\text{repeat}(C, 1, 1)$  /*  $\mathbf{S} \in \mathbb{R}^{C \times N \times d}$  */
7:  $\text{hour}, \text{day}, \text{mon} = \text{time\_feature}(t)$  /* Obtain hour, month and day from  $t$  */
8:  $\mathbf{T} = \mathbf{T}[\text{hour}].\text{repeat}(C, N, 1)$ 
9:  $\mathbf{D} = \mathbf{D}[\text{day}].\text{repeat}(C, N, 1)$ 
10:  $\mathbf{M} = \mathbf{M}[\text{mon}].\text{repeat}(C, N, 1)$ 
11:  $\mathbf{Z}_0 = \mathbf{H} + \mathbf{SE} + \mathbf{T} + \mathbf{D} + \mathbf{M}$  /*  $\mathbf{Z}_0 \in \mathbb{R}^{C \times N \times d}$  */
    /* MLP encoder */
12: for  $l$  in  $\{0, 1, \dots, L-1\}$  do
13:    $\mathbf{Z}_{l+1} = \text{FFN}_l(\mathbf{Z}_l) + \mathbf{Z}_l$ 
14: end for
15:  $\mathbf{Y} = \text{Linear}(\mathbf{Z}_L)$  /* Regression layer,  $\mathbf{Y} \in \mathbb{R}^{C \times N \times T_f}$  */
16:  $\mathbf{Y} = \mathbf{Y}.\text{transpose}(1, -1)$ 
17: return  $\mathbf{Y}$ 
```

31 December 2020. Please note that these datasets are rescaled (multiplied ten times) from the raw datasets.

- **ChinaWind and ChinaTemp** are also collected from NCEI³. These datasets contain the daily averaged wind speed and temperature of 396 stations in China (382 stations for **Temp_CN** due to missing values), spanning 10 years from 1 January 2013 to 31 December 2022.
- **ChinaPM2.5** is collected from CNEMC⁴. It contains the hourly averaged wind speed of 1,316 stations in China, spanning 4 years from 1 January 2020 to 31 December 2024. The original dataset only provides the latitudes and longitudes of stations and we obtain the elevations of stations through Open-Elevation⁵.

The statistics of the datasets are shown in Table 5 and the station distributions are shown in Figure 8.

Table 5: Statistics of datasets.

DATASET	COVERAGE	STATION NUM	SAMPLE RATE	TIME SPAN	LENGTH
GlobalWind	Global	3,850	1 hour	2 years	17,544
GlobalTemp	Global	3,850	1 hour	2 years	17,544
ChinaWind	National	396	1 day	10 years	3,652
ChinaTemp	National	382	1 day	10 years	3,652
ChinaPM2.5	National	1,316	1 hour	5 years	43,539

C.2 Introduction to Baselines

- **HI** [5], short for historical inertia, is a simple baseline that adopts the most recent historical data as the prediction results.
- **ARIMA** [33], short for autoregressive integrated moving average, is a statistical forecasting method that uses the combination of historical values to predict future values.
- **AGCRN** [1] is a STGNN that integrates adaptive graph convolution and recurrent networks to dynamically capture spatiotemporal dependencies in multivariate time series.

³<https://www.ncei.noaa.gov/data/global-summary-of-the-day/access>

⁴<https://air.cnemc.cn>

⁵<https://open-elevation.com>

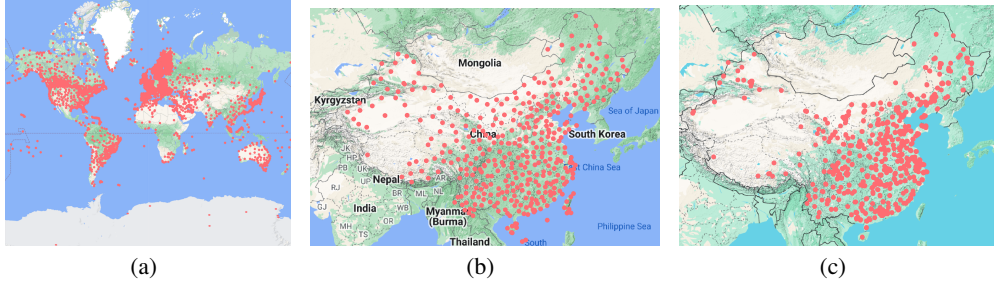


Figure 8: Distributions of the stations. (a) GlobalWind and GlobalTemp. (b) ChinaWind and ChinaTemp. (c) ChinaPM2.5.

- **MTGNN** [41] is a GNN-based model for multivariate time series forecasting. It can automatically learn the hidden spatial dependencies among variables.
- **GTS** [28] is a STGNN that learns the structure simultaneously with the GNN when the graph is unknown.
- **Informer** [53] is a Transformer for time series forecasting (TSF) with a sparse self-attention mechanism.
- **FEDformer** [54] is a frequency-enhanced Transformer combined with seasonal-trend decomposition to capture the overall trend of time series.
- **DSformer** [45] utilizes double sampling blocks to model both local and global information.
- **PatchTST** [23] divides the input time series into patches, which serve as input tokens of Transformer.
- **iTransformer** [20] is a Transformer for TSF that simply applies the attention and Feed-Forward Network (FFN) on the inverted dimensions, thereby enhancing the Transformer’s capability to capture multivariate correlations.
- **DUET** [26] is a dual clustering enhanced model for TSF that clusters temporal and channel dimensions to learn the complex correlations of time series.
- **DLinear** [49] is a lightweight baseline for TSF, which consists of a linear model and a time series decomposition module.
- **N-BEATS** [24] utilizes backward and forward residual links and a very deep stack of fully-connected layers.
- **FITS** [43] is a lightweight baseline for TSF that employs a complex-valued linear layer to learn amplitude scaling and phase shifts, enabling interpolation in the complex frequency domain.
- **AirFormer** [16] employs a dartboard-like mapping and local windows to restrict attention to focusing solely on local information.
- **Corrformer** [40] utilizes a decomposition framework and replaces attention mechanisms with a more efficient multi-correlation mechanism.
- **MRformer** [47] employs a hierarchical tree structure, stacking attention layers to capture correlations from multi-resolution data obtained by downsampling.

C.3 Evaluation Metrics

The evaluation metrics we used in the paper are defined as follows.

Mean Absolute Error (MAE)

$$\text{MAE}(\hat{\mathbf{Y}}, \mathbf{Y}) = \frac{1}{N \cdot C \cdot T_f} \sum_{i=1}^N \sum_{j=1}^C \sum_{k=1}^{T_f} |\hat{\mathbf{Y}}_t^{i,j} - \mathbf{Y}_t^{i,j}|. \quad (24)$$

Root Mean Square Error (RMSE)

$$\text{RMSE}(\hat{\mathbf{Y}}, \mathbf{Y}) = \sqrt{\frac{1}{N \cdot C \cdot T_f} \sum_{i=1}^N \sum_{j=1}^C \sum_{k=1}^{T_f} (\hat{\mathbf{Y}}_t^{i,j} - \mathbf{Y}_t^{i,j})^2}. \quad (25)$$

C.4 Optimal Settings

For reproducibility purposes, we provide the optimal hyperparameters of STELLA on the five datasets, as illustrated in Figure 6. All experiments were performed on an NVIDIA GeForce RTX 4090 24GB GPU.

Table 6: The optimal settings of STELLA.

Settings		GlobalWind	GlobalTemp	ChinaWind	ChinaTemp	ChinaPM2.5
Network Architecture	Hidden Dimension	1024	2048	64	32	256
	Layers of MLP	2	2	2	3	2
	Activation Function	ReLU	ReLU	ReLU	ReLU	ReLU
	Dropout	0.2	0.2	0.2	0.2	0.2
Optimization	Batch Size	32	32	32	32	32
	Epoch	100	100	100	100	100
	Optimizer	Adam	Adam	Adam	Adam	Adam
	Learning Rate	5e-4	5e-4	5e-4	5e-4	5e-4
	Weight Decay	5e-4	5e-4	5e-4	5e-4	5e-4
	LR Scheduler	MultistepLR	MultistepLR	MultistepLR	MultistepLR	MultistepLR
	- Milestone	[1,50]	[1,50]	[1,50]	[50]	[1,25,50]
	- γ	0.5	0.5	0.5	0.5	0.5

D Additional Experimental Results

D.1 Comparison with Numerical Methods

In this section, we compare our model with the numerical weather prediction (NWP) methods in short-term global weather forecasting tasks. Conventional NWP methods use PDEs to describe the atmospheric state transitions across grid points and solve them through numerical simulations. Currently, the ERA5 from the European Centre for Medium-Range Weather Forecasts (ECMWF) and the Global Forecast System (GFS) from NOAA are the most advanced global forecasting models. ERA5 provides gridded global forecasts at a 0.5° resolution while GFS at a 0.25° resolution.

To make the comparison practical, we utilize bilinear interpolation with height correction to obtain the results for scattered stations, which is aligned with the convention in weather forecasting [2, 40]. The results are shown in Table 7.

Both ERA5 and GFS fail to provide accurate predictions, which indicates that grid-based NWP methods are inadequate for fine-grained station-based predictions. In contrast, STELLA can accurately forecast the global weather for worldwide stations, significantly outperforming the numerical methods.

Table 7: Forecasting results from NWP methods and our model on global weather datasets.

Methods	GlobalWind		GlobalTemp	
	MSE	MAE	MSE	MAE
ERA5 (0.5°)	2.606	1.847	5.298	3.270
GFS (0.25°)	3.161	2.340	3.864	2.287
STELLA	1.919	1.284	2.724	1.858

D.2 Efficiency Analysis under Limited Computational Resources

In many application scenarios, training a DL model from scratch is necessary, even when computational resources are limited. However, complex model architectures come with significant costs, including hundreds of millions of parameters and extended training times, which hinder their applicability. In the era of large models [9, 29, 50, 42], this phenomenon is particularly evident. STELLA comes with an efficient solution for this. In this section, we conduct a further efficiency analysis under limited computational resources. To simulate scenarios with limited computational resources, we utilize an Intel Xeon Gold 6330 CPU to train and test models. The top five performing models on the GlobalWind dataset are selected for to experiment. Table 8 presents the training time and inference time of different models on CPU.

The following observations can be made: (1) Except for STELLA, the other top five models are all based on the Transformer architecture. (2) The training times for these models on a CPU are impractical. For instance, training Corformer for 50 epochs would take approximately 10 months. In contrast, STELLA can be efficiently trained in a CPU environment, requiring only about 2 hours to train STELLA-10k for 50 epochs. Furthermore, due to its compact parameter size and straightforward computations, STELLA is highly suitable for deployment on edge devices for inference tasks.

Table 8: The training time and inference time of the top five models on CPU.

METHODS	PERFORMANCE RANKING	TRAINING TIME / EPOCH	INFERENCE TIME / SAMPLE
PatchTST	5	2.25h	0.10s
Corrformer	4	141h	3.90s
iTransformer	3	17.5h	0.70s
AirFormer	2	56.1h	1.77s
STELLA-10k	1	141s	2ms

E Discussion

E.1 Comparison between STELLA and Other Lightweight Methods

In the main text, we provide a comprehensive comparison between STELLA and Transformer-based models. Here, we present a further discussion about the distinctions between STELLA and other lightweight models in terms of both performance and efficiency. We conduct additional experiments on the GlobalWind dataset and compare STELLA to three lightweight TSF models, N-BEATS [24], DLinear [49], FITS [43]. Table 9 presents the performance-efficiency comparison results, from which we derive the following conclusions:

Table 9: The training time and inference time of the top five models on CPU.

METHODS	PERFORMANCE RANKING	PARAMS	EPOCH TIME	MAX MEM.
N-BEATS	10	121.78k	37s	2.00GB
DLinear	7	2.35k	27s	1.10GB
FITS	9	1.80k	29s	1.27GB
STELLA-10k	1	9.98k	30s	792MB

- **Performance.** In terms of performance, STELLA achieves the top ranking among the 17 baselines (15 of which are DL-based), demonstrating a substantial lead over other lightweight models. This superiority stems from the fact that other lightweight models fail in modeling spatial correlations. Taking DLinear as an example, its use of a single linear layer to predict data across all sites is inherently unsuitable for multi-station forecasting scenarios.
- **Efficiency.** We analyze efficiency using three metrics: parameter counts, training time per epoch, and maximum GPU memory usage. FITS and DLinear outperform STELLA in terms of parameter counts. However, STELLA achieves comparable training speed to other lightweight models, while its GPU memory usage is even *lower*. This is because other

lightweight models incorporate additional operations to balance performance. For instance, DLinear introduces convolutional modules, while FITS employs complex frequency-domain interpolation. Despite having linear layers as their backbone, these models are less efficient than anticipated. Overall, STELLA maintains a high level of efficiency without compromising performance.

E.2 Comparison between STELLA and PINNs

While PINNs (physics-informed neural networks) often incorporate physical constraints or regularization, our approach is different in mechanism and purpose. Instead of using physical equations as loss terms or auxiliary supervision in ATSF tasks [12, 8], we draw theoretical motivation from the governing PDEs of atmospheric dynamics to construct input representations (STPE) that inject geographical and temporal priors directly into the model. To the best of our knowledge, this is the first work to prove, both theoretically and empirically, that such a representation alone, even coupled with a simple MLP, can surpass many state-of-the-art ATSF models in both accuracy and efficiency.

E.3 Can STPE be Applied to Linear Model?

As shown in §4.5, STPE can be applied to Transformer-based models to enhance their performance, which naturally raises a question: *Can STPE also be applied to more lightweight methods (e.g., a linear model)?*

To address this, we conducted extensive experiments and found that the improvement was relatively marginal, especially in global weather forecasting tasks (GlobalWind and GlobalTemp datasets), whereas more substantial gains were observed in national-scale tasks (ChinaWind and ChinaTemp datasets), as shown in Table 10. According to **Theorem 1**, the evolution of atmospheric states is nonlinear, thus, the fitting capability is the core limitation for linear models.

Table 10: Improvements obtained by the adoption of STPE.

Datasets		GlobalWind		GlobalTemp		ChinaWind		ChinaTemp	
Metric		RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
DLinear	Original	2.005	1.350	3.149	2.072	7.309	5.031	4.990	3.659
	+STPE	2.002	1.348	3.138	2.078	7.250	5.002	4.881	3.596

F Case Study

In the main text, we present multi-station collaborative prediction results to provide an intuitive understanding of STELLA’s ability to capture spatial correlations and perform collaborative predictions. Here, to enable a clear comparison among different models, we provide supplementary prediction cases from individual stations. We select three representative datasets for each ATSF task, and the results are given by the following advanced models: STELLA, iTransformer [20], Corrformer [40], AirFormer [16], PatchTST [23], DLinear [49]. Figure 9-11 indicates that STELLA provides the most accurate prediction and demonstrates superior performance among the models.

G Limitations and Future Work

Multivariate Correlations. Atmospheric variables are tightly coupled (Eq.14); however, in our modeling, we decoupled these variables, overlooking the multivariate correlations and training separate models for each atmospheric variable. This was primarily driven by performance considerations and was consistent with previous work [40]. Accounting for multivariate correlations and jointly forecasting multiple atmospheric variables might introduce learning challenges, potentially leading to degraded prediction performance. Since the number of atmospheric variables to be predicted is usually limited, independently forecasting each variable is a cost-effective approach relative to performance improvement, especially when using the lightweight STELLA model. Nonetheless, incorporating other meteorological variables as covariates may enhance prediction performance, which we leave as future work.

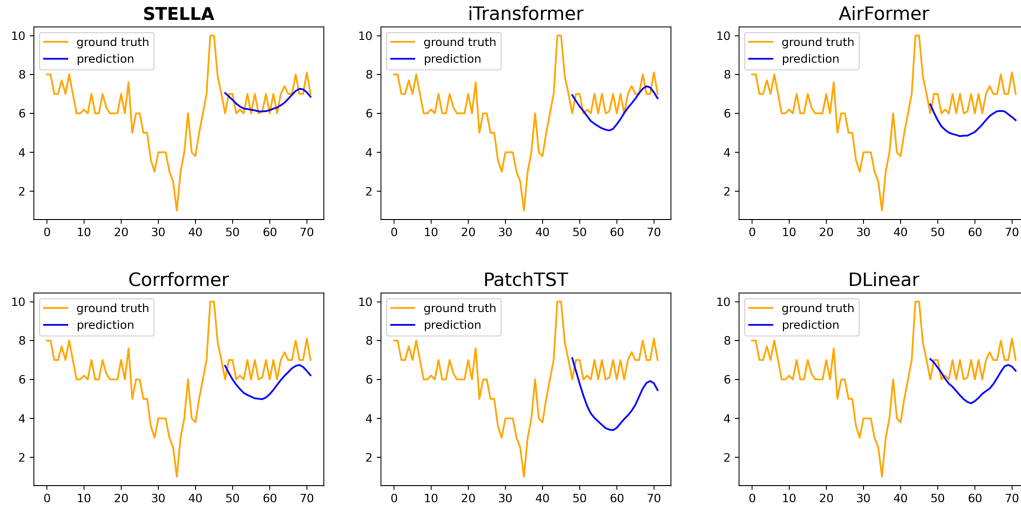


Figure 9: Visualization of the prediction results on the GlobalWind dataset.

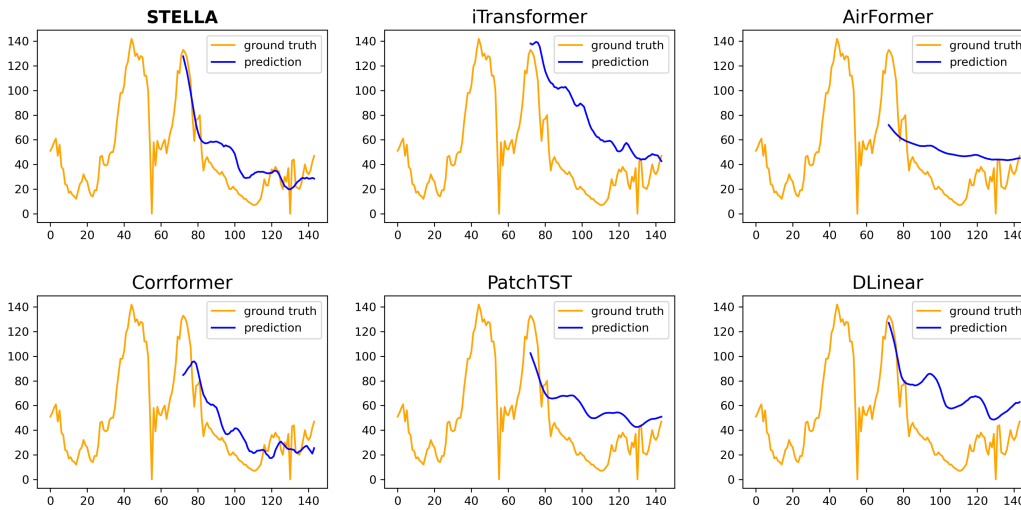


Figure 10: Visualization of the prediction results on the ChinaPM2.5 dataset.

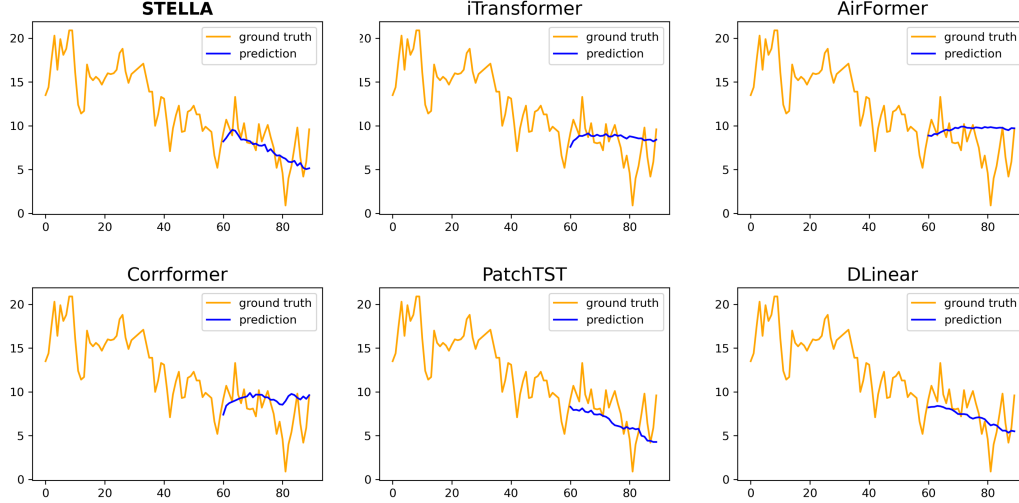


Figure 11: Visualization of the prediction results on the ChinaTemp dataset.

Extreme Weather Forecasting. When forecasting a target time series with violent fluctuations, the model may provide over-smooth predictions, leading to an inability to accurately forecast extreme weather events in practical applications. This is a common issue for DL models due to the use of MSE/MAE loss. A possible explanation is that MSE loss compresses the feature representations into a constrained space, limiting the model’s ability to capture high-entropy features, especially those with significant variability [51]. Incorporating cross-entropy as a classification loss into the loss function may help address this issue [35]. Additionally, although we considered the physical principles of atmospheric dynamics, we did not directly incorporate physical constraints into the model’s predictions. Doing so may help mitigate the issue of over-smoothing and also improve the interpretability of the predictions. We leave this as future work.

Fitting Capacity of MLP Backbone. We recognize that MLPs, due to their shallow and simple architecture, may face limitations in fitting capacity. We believe this constitutes the primary bottleneck limiting STELLA’s further performance improvement. The motivation of this paper is to demonstrate that even minimalist architectures, when integrated with spatial-temporal knowledge, can achieve SOTA performance while offering superior efficiency. As shown in Table 4, introducing STPE into iTransformer leads to a new SOTA on GlobalWind and other datasets, demonstrating that STPE is not only effective on MLPs, but also generalizable to more expressive architectures. We leave the exploration of alternative architectures for future research.

H Broader Impact

This paper proposes STELLA, a lightweight approach for atmospheric time series forecasting. STELLA demonstrates high training efficiency, which helps reduce energy consumption and benefits domains such as agriculture and the economy. However, the deep learning-based approach may lack interpretability in forecasting results, and regional biases in training data may disadvantage certain populations.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: The abstract and introduction clearly state the main contributions and scope of the paper.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: The paper discusses the limitations in Appendix.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[Yes\]](#)

Justification: The paper provides the assumptions and proofs in the Appendix.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [\[Yes\]](#)

Justification: The paper provides detailed descriptions of the datasets, experimental setups, implementation details, and hyperparameters, which are sufficient to reproduce the main results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The paper provides open access to the data and code in the supplemental material with sufficient instructions to reproduce the main experimental results.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The paper thoroughly describes the training and test details.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: The paper provides the standard deviations of the main experimental results and a clear explanation of the statistical methods used to compute metrics, ensuring the robustness of the reported findings.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: The paper provides sufficient information on the computer resources.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: The research aligns with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: The paper discusses the broader impacts in the appendix.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper credits the owners of any used assets.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [\[Yes\]](#)

Justification: The paper introduces new assets that are well documented alongside the assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [\[NA\]](#)

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [\[NA\]](#)

Justification: The research does not involve human subjects, so IRB approvals or equivalent review are not required.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.