Dyna-STTN: Distilling Physical Dynamics into Spatio-Temporal Transformer Network for Weather Forecasting

Jiafeiyang Dong [1] **Yanhan Liu** [2]

[1] *School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore* [2] *Department of Physics, National University of Singapore, Singapore*. Correspondence to: Jiafeiyang Dong JDONG010@e.ntu.edu.sg.

1. Introduction

Global climate change has intensified the volatility of meteorological systems. Traditional Numerical Weather Prediction (NWP) models[1], which possess strong theoretical properties and interpretability for various atmospheric processes, offer physical rigor but suffer from prohibitive computational costs and sensitivity to initial conditions. While offering high predictive accuracy, data-driven models[2, 3, 4, 5, 6, 7, 8, 9, 10] remain opaque "black boxes" that frequently lack interpretability and violate physical consistency, particularly during extreme weather events[11].

To bridge the gap between physical interpretability and prediction accuracy, we propose **Dyna-STTN**, a teacher-student distillation framework that integrates atmospheric physical dynamic principles into deep learning architectures. Specifically, a PDE-based teacher network models the evolution of potential energy fields to reveal underlying atmospheric dynamics. This teacher guides a lightweight Spatio-Temporal Transformer student through a dual-loss objective, forcing the model to adhere to governing physical laws while capturing complex spatio-temporal dependencies.

In summary, the contributions of this paper are threefold:

- We develop **Dyna-STTN**, a novel framework that **distills physical PDE constraints** into a **Spatio-Temporal Transformer** through a **dual-loss objective** architecture to **ensure both accuracy and physical consistency**.
- We perform systematic validation on the WeatherBench dataset, demonstrating a **22.5% maximum reduction** in RMSE compared to the baselines.
- We implement a **modular and scalable** architecture that allows the physics-informed distillation paradigm to be **readily transferred** to other domains, such as traffic flow forecasting.

2. Methodology

2.1 Teacher: Physical Differential Equations Network

The teacher network models atmospheric dynamics [12] via a Meteorological Potential Energy Field (MPEF) $\mathbf{e} \in \mathbb{R}^N$, governed by the physical partial differential equation (PDE):

$$\frac{\partial \mathbf{e}}{\partial t} = -\theta \odot (\Delta \mathbf{e}) \tag{1}$$

where Δ is the graph Laplacian. Following the paradigm of physics-guided networks [13], we generalize this into a continuous-time Neural ODE to capture latent energy discrepancies:

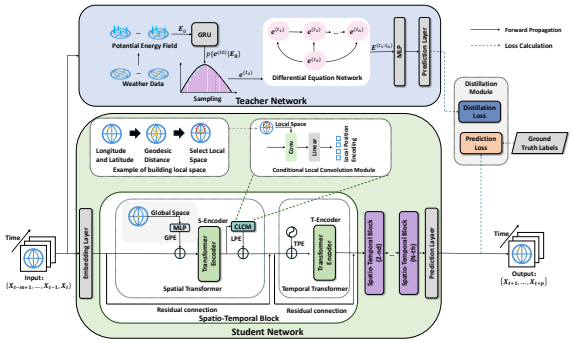


Fig. 1: Overview of Dyna-STTN. The framework integrates a physics-guided teacher network and a spatio-temporal transformer student through a knowledge distillation bridge.

$$\frac{\partial \mathbf{e}(t)}{\partial t} = \Phi(\mathbf{e}(t), t, \theta) = -\theta \odot \mathrm{Tanh}(\Delta \mathbf{e}) \tag{2}$$

where Φ is a residual Graph Convolutional Network [14] mapping the non-linear energy evolution trajectory.

2.2 Student: Spatio-Temporal Transformer Network

Guided by the teacher's physical consistency, the student network extracts multi-scale dependencies. Static geographic context is provided by Global Position Encoding (GPE), mapping coordinates $(P_{i,1}, P_{i,2})$ into latent space. To capture localized spatial correlations, the Conditional Local Convolution Module(CLCM) [15] dynamically generates kernels based on geodesic distances ρ_{ij} and angular factors γ_{ij}:

$$\kappa(z_j, z_i) = \frac{\gamma_{ij}}{2\pi} \cdot \exp\left(-\frac{\rho_{ij}^2}{\tau}\right) \cdot \mathrm{MLP}([z_j, z_i]) \tag{3}$$

where τ is a learnable smoothing parameter. Subsequently, the Temporal Transformer models long-range dependencies across the 12-hour horizon. By integrating sinusoidal position encodings (P_e), the model ensures the chronological consistency of meteorological sequences:

$$\mathbf{Z}^{(l+1)} = \text{T-Encoder}(\mathbf{Z}^{(l')} + P_e) \tag{4}$$

where $\mathbf{Z}^{(l')}$ denotes high-order spatial features. This hierarchical architecture ensures that the student distills physical rigor from the teacher while maintaining inference efficiency.

2.3 Physical Knowledge Distillation

To facilitate efficient knowledge transfer from the teacher to the student, we implement a **dual-loss objective** that reconciles physical alignment with predictive precision. The total loss is minimized as:

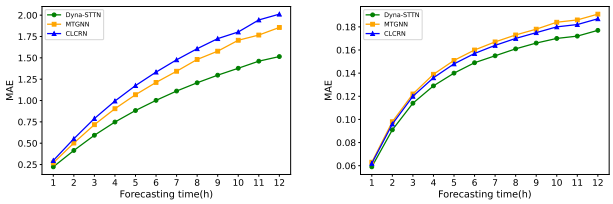
$$\mathcal{L}_{total} = \alpha \mathcal{L}_{task} + (1 - \alpha)\mathcal{L}_{KD} \tag{5}$$

where α is a balancing coefficient. The task loss, defined as $\mathcal{L}_{task} = \frac{1}{T}\sum_{t=1}^{T}|Y_t^s - Y_t^{true}|$, ensures accurate forecasting relative to ground-truth observations. Simultaneously, the distillation loss $\mathcal{L}_{KD} = \frac{1}{T}\sum_{t=1}^{T}|F_t^s - F_t^t|$ measures the Mean Absolute Error (MAE) between the student (F_t^s) and teacher (F_t^t) feature outputs. This mechanism forces the student network to internalize the latent atmospheric dynamics modeled by the teacher, effectively embedding physical priors into the Transformer architecture without increasing inference latency.

3. Experiments

Table 1: Performance of our method and baselines over 12 hours on WeatherBench [16]. Results with underlines are the best achieved by baselines, and bold results are the overall best.

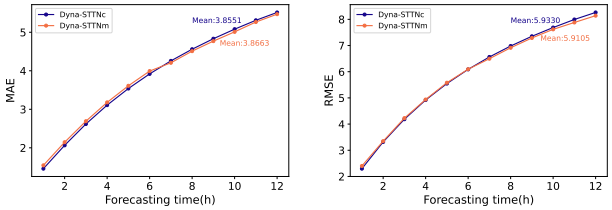
Model	Temperature		Cloud cover		Humidity		Wind	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
ARIMA	4.9043	5.7771	0.3234	0.3767	9.9751	12.2784	3.8187	4.5806
LSTM	2.7116	4.6402	0.2740	0.3791	13.2951	18.5921	4.0157	5.1958
Informer	2.9863	4.6274	0.2255	0.3002	12.7210	13.0772	3.0866	4.2826
FEDformer	1.1093	1.7943	0.1825	0.2766	5.6888	8.1784	1.8702	3.0125
STGCN	4.3525	6.8600	0.2019	0.2954	7.9753	11.1092	3.6467	4.8236
DCRNN	1.3172	2.1784	0.1593	0.2541	5.0461	7.9562	1.4274	2.3334
ASTGCN	1.4896	2.4622	0.1996	0.2957	7.2884	10.4711	2.0889	3.1356
MTGNN	0.7706	1.3246	0.1507	0.2427	4.1283	6.3966	1.1937	2.0061
CLCRN	1.1688	1.8825	0.1480	0.2421	4.5310	7.0781	1.3260	2.1292
Teacher-Net	0.9544	1.6508	0.1566	0.2519	4.8213	7.5449	1.3195	2.1579
Student-Net	0.7152	1.2342	0.1408	0.2322	3.9956	6.1537	1.0626	1.6749
Dyna-STTN	**0.6919**	**1.1826**	**0.1402**	**0.2296**	**3.8351**	**5.9130**	**0.9870**	**1.5540**
Improvement	**10.2%**	**10.7%**	**5.3%**	**5.2%**	**7.1%**	**7.6%**	**17.3%**	**22.5%**



(a) MAE on Wind (b) MAE on Cloud cover

Fig. 2: MAE results per prediction step compared with baselines for two meteorological variables.

Quantitative evaluations on WeatherBench (Table 1) demonstrate that Dyna-STTN consistently achieves the best performance across all indicators, outperforming baselines by 5.2%–22.5% in MAE and RMSE. Notably, the model delivers a substantial 22.5% reduction in wind RMSE. As illustrated in the 12-hour forecasting curves (Fig. 2), Dyna-STTN maintains the lowest error accumulation over time, verifying the effectiveness of the physics-distillation mechanism in preserving long-range predictive stability.



(a) MAE on Humidity (b) RMSE on Humidity

Fig. 3: Impact of different distance functions.

Ablation on distance functions shows that while Cosine distance facilitates faster initial convergence, MAE offers superior training stability and computational efficiency. Given the negligible difference in final predictive performance, MAE is adopted as the default metric.

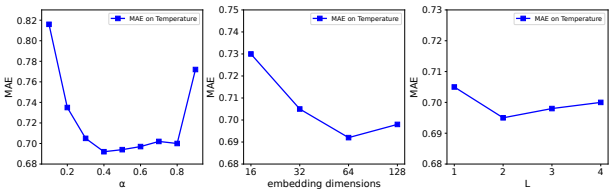


Fig. 4: Performance of different hyper-parameters.

The architecture adopts 64-dimensional embeddings and dual-layer blocks to optimize the trade-off between spatio-temporal expressive capacity and computational efficiency. Furthermore, the distillation weight α is calibrated to ensure effective synergy between physical priors and data-driven learning, maintaining performance stability across a robust range.

4. Conclusion

In this work, we present Dyna-STTN, a teacher-student distillation framework that successfully integrates physical PDE constraints into a Spatio-Temporal Transformer architecture. The proposed model achieves significant performance, demonstrating a maximum error reduction of 22.5% compared to baselines. Through systematic validation, including multi-scale meteorological experiments and rigorous ablation studies, our model still remains robust. Furthermore, the modular and scalable design of the system ensures its flexibility, allowing the framework to be effectively extended or transferred to other domains, such as traffic prediction.

For future work, we aim to replace the conventional MSE loss with a distribution-alignment objective. By minimizing the Wasserstein distance between predicted and label conditional distributions, the model could achieve unbiased training while better preserving the intrinsic geometric structures and temporal autocorrelations inherent in meteorological sequences. Additionally, we plan to incorporate semantic modules into the student architecture to characterize temporal similarities through high-dimensional semantic spatial dependencies.

Acknowledgments

We would like to thank Professor Minglong Lei from Beijing University of Technology for his invaluable guidance and insightful suggestions during the undergraduate thesis project that formed the foundation of this work.

References

[1] Sultan Al-Yahyai, Yassine Charabi, and Adel Gastli. Review of the use of numerical weather prediction (NWP) models for wind energy assessment. *Renewable and Sustainable Energy Reviews*, 14(9):3192–3198, 2010.

[2] Zahra Karevan and Johan A. K. Suykens. Transductive LSTM for time-series prediction: An application to weather forecasting. *Neural Networks*, 125:1–9, 2020.

[3] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. Attention Based Spatial-Temporal Graph Convolutional Networks for traffic flow forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 922–929, 2019.

[4] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-Temporal Graph Convolutional Networks: A deep learning framework for traffic forecasting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3634–3640, 2018.

[5] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations*, 2018.

[6] 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 Conference on Knowledge Discovery and Data Mining*, pages 753–763, 2020.

[7] 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.

[8] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34:22419–22430, 2021.

[9] Hao Lin, Zhangyang Gao, Yutong Xu, Lirong Wu, Li 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.

[10] Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *International Conference on Learning Representations (ICLR)*, 2023.

[11] Martin G Schultz et al. Can deep learning beat numerical weather prediction? *Phil. Trans. R. Soc. A*, 379(2194):20200097, 2021.

[12] Penelope Maher et al. Model hierarchies for understanding atmospheric circulation. *Reviews of Geophysics*, 57(2):250–280, 2019.

[13] Jiahao Ji, Jingyuan Wang, Zhe Jiang, Jiawei Jiang, and Hu Zhang. Stden: Towards physics-guided neural networks for traffic flow prediction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(4):4048—-4056, June 2022.

[14] Thomas N. Kipf and Max Welling. Semi-supervised classification with Graph Convolutional Networks. In *International Conference on Learning Representations (ICLR)*, 2017.

[15] Irina Sandu et al. Impacts of orography on large-scale atmospheric circulation. *npj Climate and Atmos. Sci.*, 2(1):10, 2019.

[16] Stephan Rasp, Peter D. Dueben, Sebastian Scher, Jonathan A. Weyn, Soukayna Mouatadid, and Nils Thuerey. Weatherbench: A benchmark data set for data-driven weather forecasting. *Journal of Advances in Modeling Earth Systems*, 12(11), November 2020.

Appendix A. Problem Formulation

We first define the notations used throughout this paper. Weather forecasting is formulated as a multivariate time-series prediction task within spatial meteorological graphs.

1.1 Meteorological Graph Definition

A spatial meteorological graph is denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{P})$, where $\mathcal{V}$ is a set of N nodes representing meteorological stations. $\mathcal{E}$ is a set of edges denoting the meteorological proximity and spatial correlations among different geographical regions. $\mathcal{P} = \{P_i = (P_{i,1}, P_{i,2}) \in \mathbb{S}^2, i = 1, \ldots, N\}$ is the set of geographical locations (longitude and latitude) for each node, mapped onto a 2D spherical space $\mathbb{S}^2$.

1.2 Weather Data Representation

The weather data collected by these stations is denoted as $\mathbf{X} \in \mathbb{R}^{T \times N \times d}$, where T is the total length of the time series and d is the dimension of node features (e.g., temperature, wind speed, humidity). We

treat the weather data as a dynamic temporal graph; at each time step t, the global weather condition is represented as $\mathbf{X}_t \in \mathbb{R}^{N\times d}$.

1.3 Predictive Objective

The goal of our Dyna-STTN framework is to learn a mapping function $\mathcal{F}$ that utilizes the meteorological network $\mathcal{G}$ and M steps of historical observations to predict the subsequent P steps of future weather conditions:

$$\hat{\mathbf{X}}_{t+1:t+P} = \mathcal{F}\left(\mathcal{G}, \mathbf{X}_{t-M+1:t}; \Theta\right) \tag{A1}$$

where $\mathbf{X}_{t-M+1:t} \in \mathbb{R}^{M\times N\times d}$ represents the input historical sequence, $\hat{\mathbf{X}}_{t+1:t+P} \in \mathbb{R}^{P\times N\times d}$ is the predicted future sequence, and Θ denotes the learnable parameters optimized via physics-guided distillation.

Appendix B. Mathematical Derivations of Teacher Network

2.1 Meteorological Potential Energy Field (MPEF)

We assume that weather fluctuations are driven by a latent potential energy field within the meteorological graph. This field represents the energy distribution across various atmospheric regions, where gradients in potential energy provide the driving forces for atmospheric motion.

For each station $v_i \in \mathcal{V}$, we define the potential energy as $e_i \in \mathbb{R}$. The overall Meteorological Potential Energy Field (MPEF) is expressed as a vector:

$$\mathbf{e} = (e_1, \ldots, e_N)^\top \in \mathbb{R}^N \tag{B.1}$$

2.2 Potential Energy Differential Equation

We introduce the continuity equation to describe the transmission of physical quantities (mass, energy, etc.) across the graph:

$$\frac{\partial u}{\partial t} + \operatorname{div}(\mathbf{q}) = 0 \tag{B.2}$$

where u is the energy density and $\mathbf{q}$ is the energy flux. To adapt this to our framework, we establish two key relationships:

First, we relate potential energy to energy density via a trainable volume parameter θ_i:

$$e_i = \theta_i \cdot u_i \implies \mathbf{e} = \theta \odot \mathbf{u} \tag{B.3}$$

Second, the energy flux between adjacent stations is modeled as the negative gradient of potential energy:

$$q_{ij} = -(\nabla e)_{ij} = -(e_i - e_j) \tag{B.4}$$

By substituting these into the continuity equation and utilizing the graph Laplacian property $\Delta\mathbf{e} = -\operatorname{div}(\nabla\mathbf{e})$, we derive the governing PDE for MPEF evolution:

$$\frac{\partial \mathbf{e}}{\partial t} = -\theta \odot (\Delta\mathbf{e}) \tag{B.5}$$

2.3 The Differential Equation Network

Following the Neural ODE paradigm, we generalize discrete residual updates into a continuous form to model atmospheric dynamics:

$$\frac{\partial \mathbf{e}(t)}{\partial t} = \Phi(\mathbf{e}(t), t, \theta) = -\theta \odot \operatorname{Tanh}(\Delta\mathbf{e}) \tag{B.6}$$

where Φ is implemented as a residual GCN. The use of the Tanh activation ensures stability while Δ captures spatial discrepancies between neighbors.

2.4 Weather Forecasting Pipeline

The forecasting task is achieved through the following sequential pipeline:

1. Initialization: Historical conditions $\mathbf{X}$ are mapped to the energy space via $\mathbf{E}_0 = \mathbf{W}_{CF} \odot \mathbf{X}$. The initial state $\mathbf{e}(t_0)$ is sampled from a Gaussian distribution $\mathcal{N}(\mu_0, \sigma_0)$ generated by a GRU encoder.
2. ODE Solving: An ODE solver computes the evolution result $\mathbf{E}(t_1 : t_Q)$ over Q future time steps:
$$\mathbf{E}(t_1 : t_Q) = \operatorname{ODESolver}(\Phi, \mathbf{e}(t_0), \mathbf{W}, [t_1, \ldots, t_Q]) \tag{B.7}$$
3. Mapping and Loss: An MLP maps the energy states back to predicted conditions $\hat{\mathbf{Y}}_{teacher}$. The teacher is optimized using the Mean Absolute Error (MAE) loss:
$$\mathcal{L}_T = \|\mathbf{Y} - \hat{\mathbf{Y}}_{teacher}\|_1 \tag{B.8}$$

Appendix C. Structural Details of Student Network

The student network, consists of L spatio-temporal blocks designed to capture multi-scale dependencies while adhering to the physical constraints distilled from the teacher.

3.1 Multi-scale Spatial Transformer

The spatial module Φ_s extracts inter-node dependencies by integrating global and local geographical inductive biases into the Transformer architecture.

3.1.1 Spatial Transformer Encoder

The encoder follows a multi-head self-attention (MHSA) mechanism. For an input $\mathbf{Z}^{(l)}$, the attention scores for all nodes are computed via projection matrices $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V$:

$$\operatorname{Att}(\mathbf{Z}^{(l)}) = \operatorname{softmax}\left(\frac{\mathbf{Q}^{(l)}(\mathbf{K}^{(l)})^\top}{\sqrt{d^{(l)}}}\right)\mathbf{V}^{(l)} \tag{C.1}$$

The output $\mathbf{Z}^{(l')}$ is obtained through a residual feed-forward network (FFN) with layer normalization.

3.1.2 Global Position Encoding (GPE)

To incorporate absolute geographic information, we map the longitude and latitude $(P_{i,1}, P_{i,2}) \in \mathbb{S}^2$ of

each station v_i into a latent embedding $\mathbf{p}_i$ using an MLP:

$$\mathbf{p}_i = \mathrm{MLP}([P_{i,1}, P_{i,2}]), \quad \mathbf{Z}^{(l)} = \mathbf{Z}^{(l)} + \mathbf{P}_G \tag{C.2}$$

3.1.3 Local Position Encoding via CLCM

To capture localized meteorological correlations, we introduce Conditional Local Convolution Module(CLCM). The convolution kernel κ is conditioned on neighbor features $\mathbf{z}_j$ and high-level attention representations $\mathbf{z}_i$, weighted by two physical scaling factors:

$$\kappa(\mathbf{z}_j, \mathbf{z}_i) = \underbrace{\frac{\gamma_{ij}}{2\pi}}_{\text{Angle Scale}} \cdot \underbrace{\exp\left(-\frac{\rho_{ij}^2}{\tau}\right)}_{\text{Distance Scale}} \cdot \mathrm{MLP}([\mathbf{z}_j, \mathbf{z}_i]) \tag{C.3}$$

where γ_{ij} is the azimuth angle bisector and ρ_{ij} is the geodesic distance on the Earth's surface. The local encoding $\mathbf{P}_L$ is then added to the attention output: $\mathbf{Z}^{(l')} = \mathbf{Z}^{(l')} + \mathbf{P}_L$.

3.2 Temporal Transformer

The temporal module Φ_t captures long-range dependencies across M time steps in parallel.

3.2.1 Temporal Transformer Encoder

The T-Encoder processes the spatial features $\mathbf{Z}^{(l')}$ using temporal self-attention, enabling the model to distinguish chronological patterns within the 12-hour forecasting window.

3.2.2 Temporal Position Encoding (TPE)

We employ sinusoidal absolute position encodings to preserve the sequential order of meteorological changes:

$$\begin{aligned} PE_{(pos,2k)} &= \sin(pos/10000^{2k/d}), \\ PE_{(pos,2k+1)} &= \cos(pos/10000^{2k/d}) \end{aligned} \tag{C.4}$$

3.3 Forecasting and Loss Function

The final latent representation $\mathbf{Z}^{(L)}$ is mapped back to the meteorological prediction space via dual linear layers:

$$\hat{\mathbf{Y}} = \mathrm{Linear}(\mathrm{Linear}(\mathbf{Z}^{(L)})) \tag{C.5}$$

where $\hat{\mathbf{Y}} \in \mathbb{R}^{P \times N \times d}$ denotes the P-step prediction result.

To optimize the parameters Θ of the student network, we define the training objective as the Mean Absolute Error (MAE) between the ground truth $\mathbf{Y}$ and the predicted weather conditions $\hat{\mathbf{Y}}$:

$$\mathcal{L} = \|\mathbf{Y} - \hat{\mathbf{Y}}\|_1 \tag{C.6}$$

where $\|\cdot\|_1$ denotes the ℓ_1 norm.

Appendix D. Algorithm Pseudocode

Algorithm 1 Teacher Physical Partial Differential Equations Neural Network Prediction Process

Input: Sequence $\mathbf{X}$, horizon P
Output: Future weather states $\{X_{t+1}, \ldots, X_{t+P}\}$

```
E_0 ← RNNEncoder({X_{t-m+1}, ..., X_t})
e^{(t_0)} ← SamplingModule(E_0)
for i = 1 to n do
    e^{(t_i)} ← PDELayer(e^{(t_{i-1})})
end for
E_{t_1:t_n} ← {e^{(t_1)}, ..., e^{(t_n)}}
{X_{t+1}, ..., X_{t+P}} ← Predictor(MLP(E_{t_1:t_n}))
return {X_{t+1}, ..., X_{t+P}}
```

Algorithm 2 Student Spatio-Temporal Transformer Network Prediction Process

Input: Sequence $\mathbf{X}$, horizon P, blocks L
Output: Results $\hat{\mathbf{Y}}$

```
Z^{(0)} ← Embedding(X)
for l = 1 to L do
    Z'^{(l)} ← SpatialTransformer(Z^{(l-1)}) + Z^{(l-1)}
    Z^{(l)} ← TemporalTransformer(Z'^{(l)}) + Z'^{(l)}
end for
Ŷ ← Predictor(Z^{(L)})
return Ŷ
```

Appendix E. Dataset Introduction

We evaluate our framework using the publicly available WeatherBench benchmark, a dataset derived from the ERA5 reanalysis archive and specifically curated for evaluating data-driven weather forecasting models. While the original ERA5 records provide global atmospheric states from 1979 to 2018 at a high-resolution of 0.25°, we utilize a regridded version at 5.625° resolution (comprising 2,048 spherical grid nodes) to balance modeling complexity with computational tractability.

Four core meteorological variables are selected for our experiments: temperature (K), cloud cover (dimensionless, range 0–1), humidity (%), and surface wind components ($m \cdot s^{-1}$). Detailed quantitative statistics, including mean values and standard deviations for each variable, are summarized in Table A1.

Table A1: Quantitative statistics of the WeatherBench dataset.

Metric	Temperature	Cloud cover	Humidity	Wind
# Features	1	1	1	2
# Nodes	2048	2048	2048	2048
Mean	278.90	0.68	78.03	-0.09 / 0.24
Max	323.71	1.00	171.17	32.25 / 32.90
Min	193.03	0.00	-2.50	-37.78 / -31.69
Std. Dev.	21.12	0.36	18.28	5.61 / 4.81

Appendix F. Evaluation Metrics

We evaluate the performance of Dyna-STTN and all baseline methods using two widely adopted metrics: Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). These metrics are defined as follows:

$$\mathrm{MAE} = \frac{1}{T}\sum_{t=1}^{T}|Y_t - \hat{Y}_t| \tag{E.1}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(Y_t - \hat{Y}_t)^2} \tag{E.2}$$

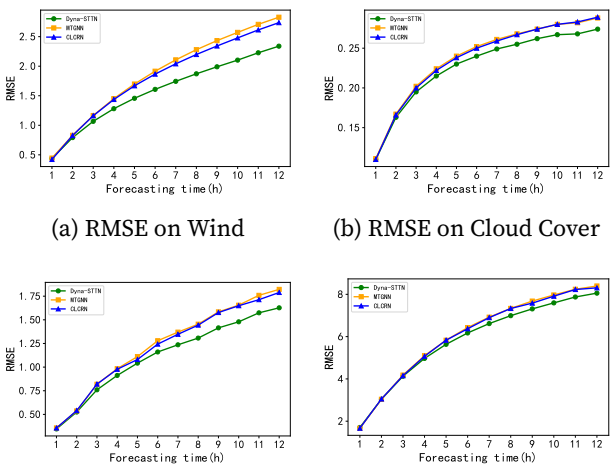
where Y_t and $\hat{Y}_t$ denote the ground truth and the predicted value at time step t, respectively.

While Mean Absolute Percentage Error (MAPE) is frequently employed in traditional spatio-temporal or time-series forecasting tasks, we observe that MAPE values are highly unstable in the context of weather forecasting. Specifically, when the ground truth values are near zero (a common occurrence in meteorological variables such as precipitation or wind speed), MAPE can exhibit extreme volatility, making it difficult to achieve a meaningful comparison.

Furthermore, our preliminary experiments indicated that the MAPE results across all models were highly consistent with the trends observed in MAE and RMSE.

Appendix G. Supplemental RMSE Analysis

Fig. A1 validates Dyna-STTN's robustness across four variables, showing a consistently stable, lower RMSE trajectory than baselines over the 12-hour horizon. The results demonstrate that Dyna-STTN can effectively mitigates error accumulation.



(a) RMSE on Wind (b) RMSE on Cloud Cover

(c) RMSE on Temperature (d) RMSE on Humidity

Fig. A1: RMSE performance comparison over a 12-hour forecasting horizon.