NEURAL INFORMATION PROCESSING SYSTEMS
NeurIPS 2024

Generalizing Weather Forecast to Fine-grained Temporal Scales via Physics-AI Hybrid Modeling

Wanghan Xu, Fenghua Ling, Wenlong Zhang, Tao Han, Hao Chen, Wanli Ouyang, Lei Bai

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

上海人工智能实验室
Shanghai Artificial Intelligence Laboratory

Background

Weather forecasts are important and varied.

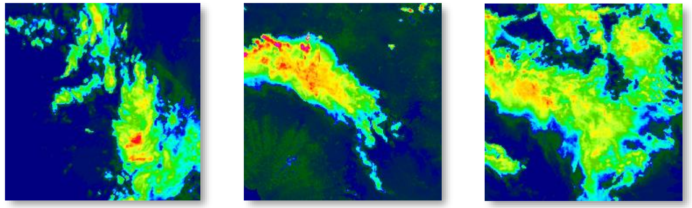
Precipitation Nowcasting

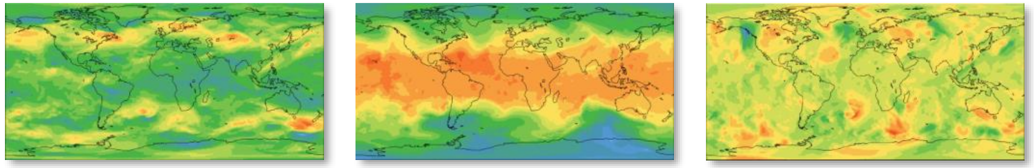
Medium-Range Forecasts

time

30min 60min 90min 120min …

1day 3day 5day 7day …





Gap
Local vs. Global
Minute-level vs. Day-level

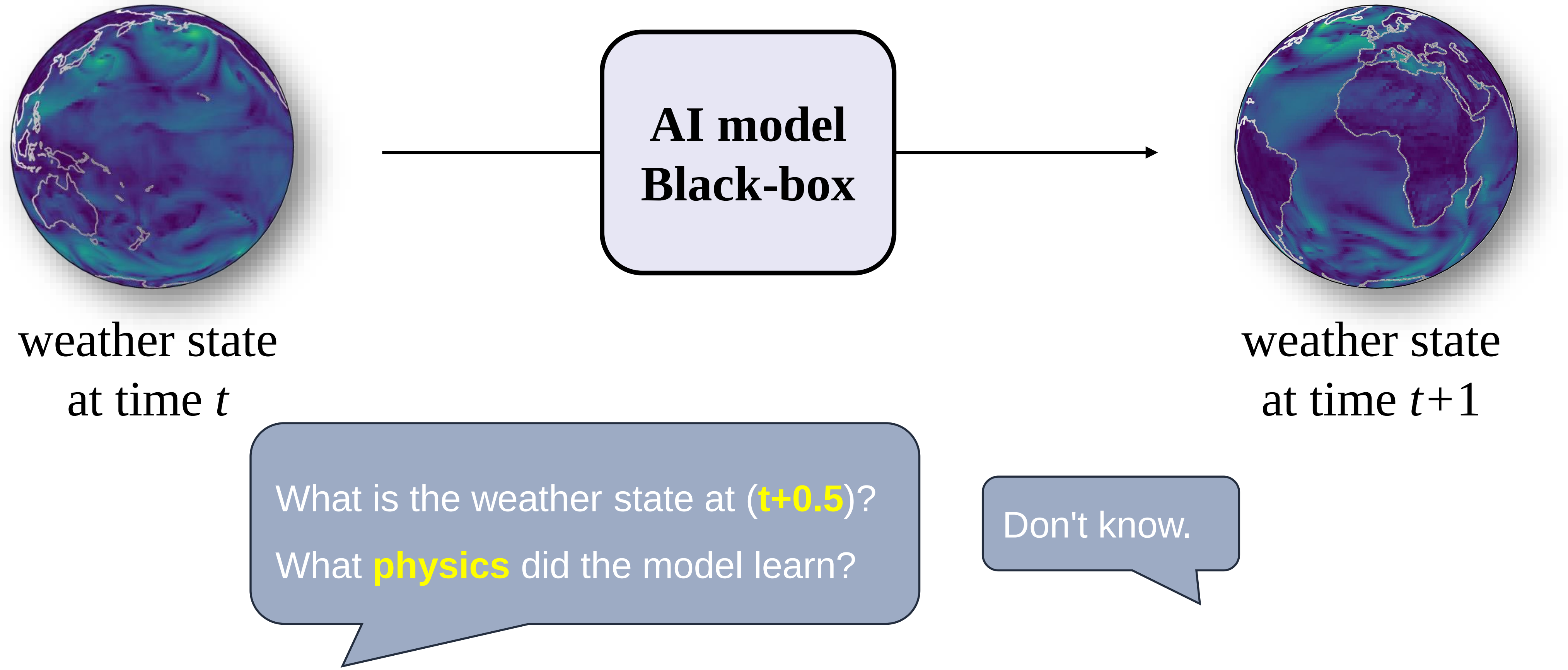
Short-term planning

Should I bring an umbrella when going out?

Long-term strategic

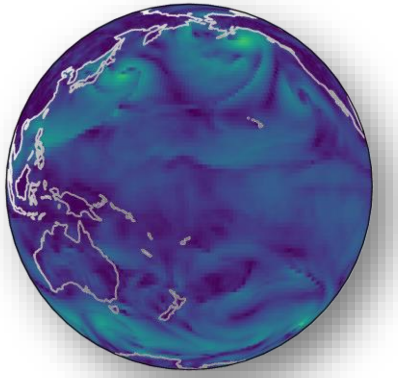
How to plan maritime trade routes?

Challenge and Motivation



Challenge: Existing black-box AI models are unable to generalize at finer temporal scales beyond the inherent time resolution of the training datasets due to the absence of fine-grained physics modeling.

Challenge and Motivation



weather state
at time t

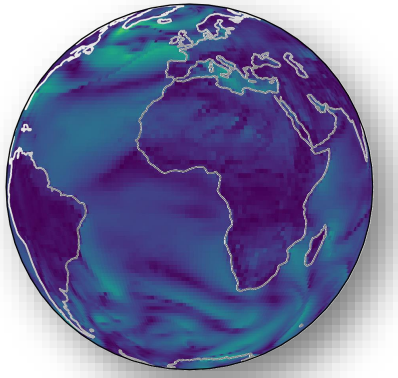
AI model

Physical Priors

$$\frac{\mathrm{d}\mathbf{V}}{\mathrm{d}t} + f\mathbf{k} \times \mathbf{V} = -g\nabla_p z + \mathbf{F}_h$$

$$\frac{\partial \phi}{\partial p} = -\frac{1}{\rho}$$

$$\ldots$$



weather state
at time t+1

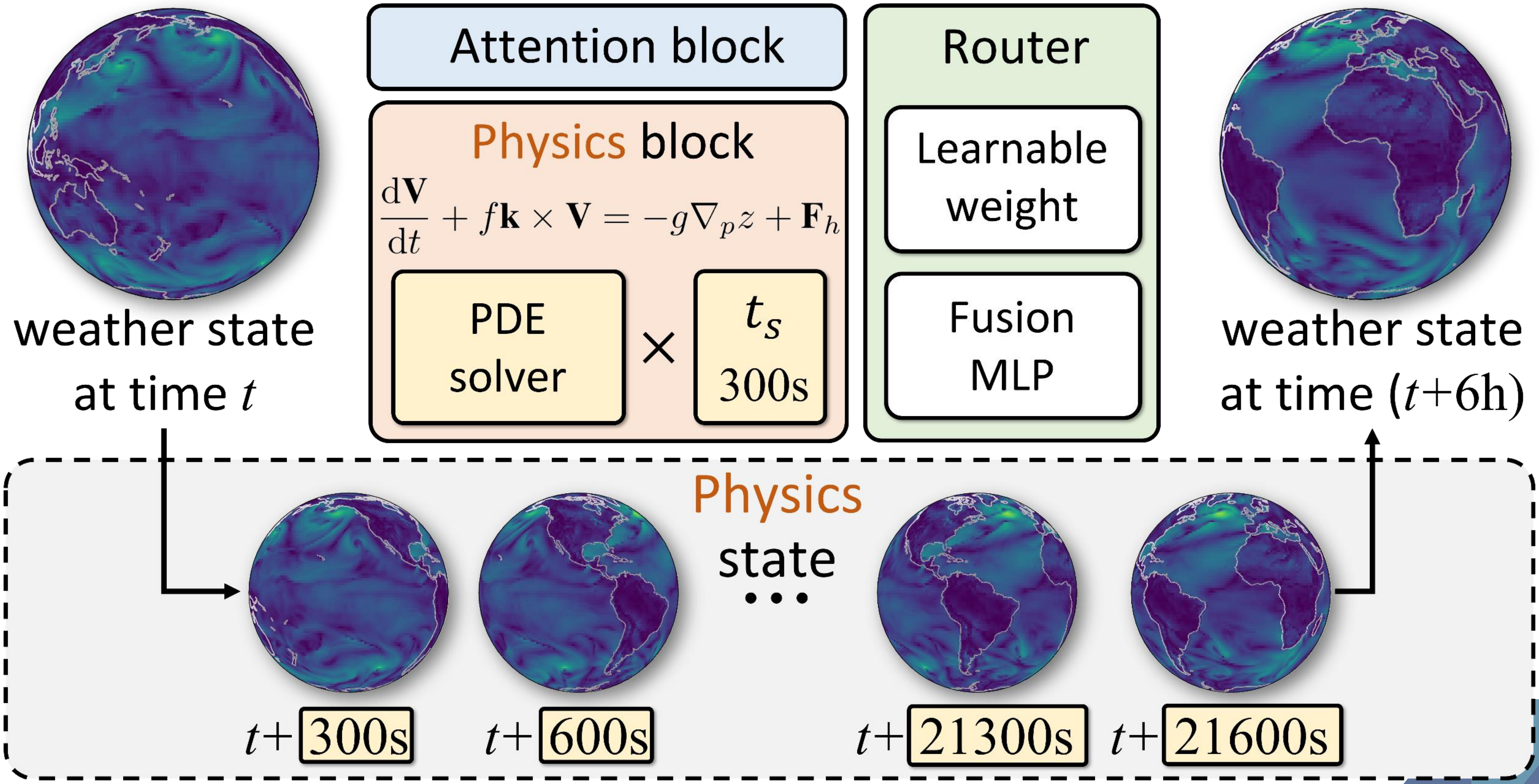
What is the weather state at (**t+0.5**)?

$$\begin{cases} \Delta u = \mathrm{PDESolver}_u(u, v) \\ \Delta v = \mathrm{PDESolver}_v(u, v) \end{cases} \rightarrow \begin{cases} u_{\mathrm{t+0.5}} = u_{\mathrm{t}} + \Delta u \times 0.5 \\ v_{\mathrm{t+0.5}} = v_{\mathrm{t}} + \Delta v \times 0.5 \end{cases}$$

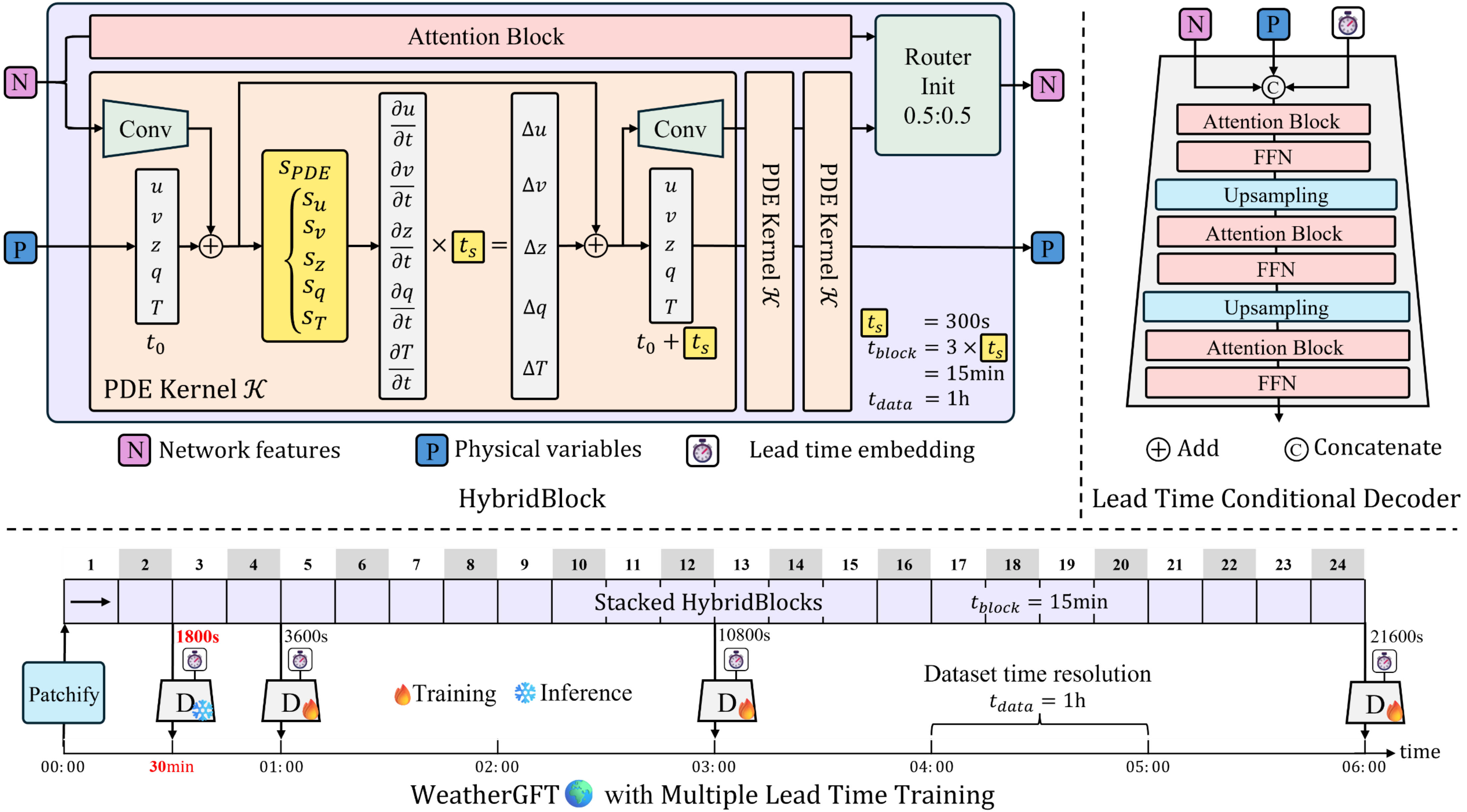
What **physics** did the model learn?

Momentum Theorem
Conservation of Energy
…

Method Overview



Pipeline


Attention Block
Router Init 0.5:0.5
N
P
Conv
Conv
u v z q T
t_0
s_{PDE}
s_u s_v s_z s_q s_T
$\frac{\partial u}{\partial t}$ $\frac{\partial v}{\partial t}$ $\frac{\partial z}{\partial t}$ $\frac{\partial q}{\partial t}$ $\frac{\partial T}{\partial t}$ $\times\, t_s =$
Δu Δv Δz Δq ΔT
$t_0 + t_s$
PDE Kernel $\mathcal{K}$
PDE Kernel $\mathcal{K}$
PDE Kernel $\mathcal{K}$
$t_s = 300$s
$t_{block} = 3 \times t_s = 15$min
$t_{data} = 1$h
N Network features
P Physical variables
Lead time embedding
HybridBlock
Attention Block
FFN
Upsampling
Attention Block
FFN
Upsampling
Attention Block
FFN
Add
Concatenate
Lead Time Conditional Decoder
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
Stacked HybridBlocks
$t_{block} = 15$min
Patchify
1800s
3600s
10800s
21600s
D
Training
Inference
Dataset time resolution
$t_{data} = 1$h
00:00
30min
01:00
02:00
03:00
04:00
05:00
06:00
time
WeatherGFT with Multiple Lead Time Training

Pipeline

Specifically, to enable our model to generalize at a finer-grained temporal resolution, we employ PDEs to model the evolution at a finer time scale:

$$\text{PDE Kernel }\ \mathcal{K}\left(\mathcal{X}\right) = S_{PDE}(\mathcal{X})t_s + \mathcal{X}$$

$$\mathcal{X}_{t_s} = \mathcal{K}(\mathcal{X}_0),\ \text{where}\ t_s = \frac{1}{m}t_{data},\ m \in \mathbb{Z}^+$$

For example,for temperature *T*, its derivative with respect to time is shown below:

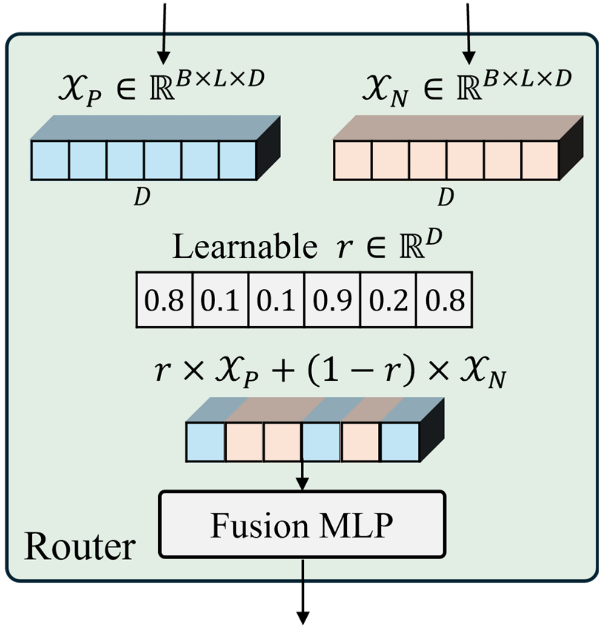
$$\frac{\partial T}{\partial t} = \frac{-L\frac{\partial z}{\partial p}w - \frac{\partial z}{\partial p}w}{c_p} - u\frac{\partial T}{\partial x} - v\frac{\partial T}{\partial y} - w\frac{\partial T}{\partial p},\ \text{where}\ w = -\int\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)\mathrm{d}p$$

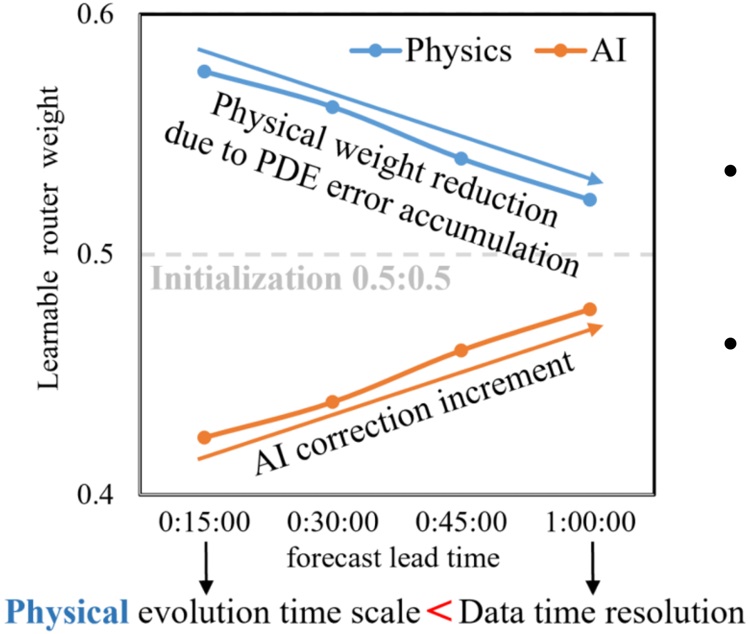
Calculating S_{PDE} requires the use of differential and integral operations. We designed a fast implementation of differentiation and integration through convolution and matrix multiplication respectively.

$$\begin{cases} \frac{\mathrm{d}\mathcal{X}}{\mathrm{d}x} = \frac{1}{12}Conv\left(\mathcal{X}, K_x\right) \\ \int \mathcal{X}\mathrm{d}x = \mathcal{X}M_x \end{cases}, K_x = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & -8 & 0 & 8 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, M_x = \begin{bmatrix} 1 & 1 & \cdots & 1 & 1 \\ 0 & 1 & \cdots & 1 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 1 \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix} \in \mathbb{R}^{W \times W}$$

Pipeline

· Adaptive Router





- The physics module performs the main fine-grained simulation.
- AI module performs dynamic bias correction to eliminate the cumulative error.

· Lead Time Conditional Decoder

$$t_{emb} = \sin(\pi \cdot t \cdot W) \oplus \cos(\pi \cdot t \cdot W) \oplus t, \text{ where } t \text{ is lead time}$$

Experiment

· **Try to answer the following questions:**

(1) How does the model perform on the medium-range forecasting task?

(2) How does the model perform on the ***generalized 30-minute nowcasting*** task?

(3) As a hybrid expert model of AI and physics, what roles do they each play?

(4) How do PDE kernel and multi-lead time training contribute to the overall performance?

· Dataset

Dataset	Train	Test	Time resolution
WeatherBench	✓	✓	1-hour
NASA	×	✓	30-minute

Name	Description	Levels
u10	x-direction wind at 10m height	Single
v10	y-direction wind at 10m height	Single
t2m	Temperature at 2m height	Single
tp	Hourly precipitation	Single
z	Geopotential	13
q	Specific humidity	13
u	x-direction wind	13
v	y-direction wind	13
T	Temperature	13

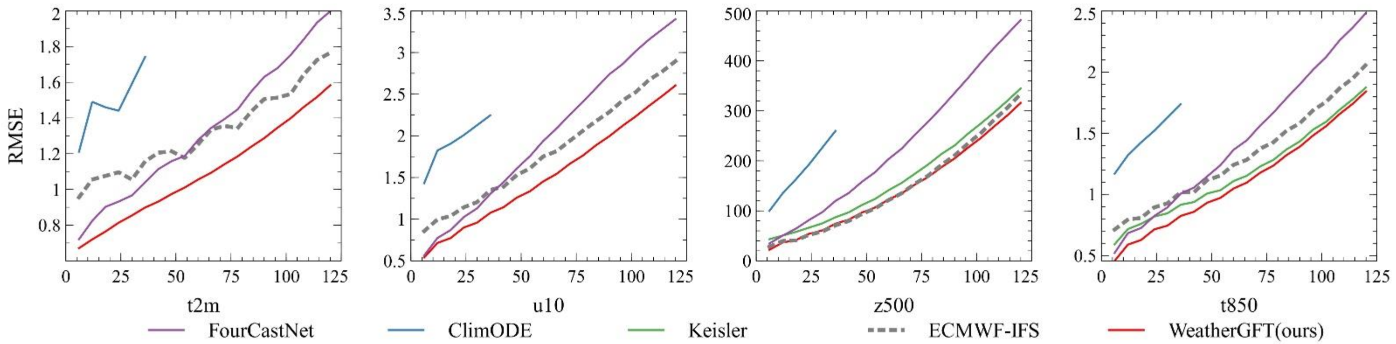
The 13 levels are 50, 100, 150, 200, 250, 300, 400, 500, 600, 700, 850, 925, 1000 hPa.

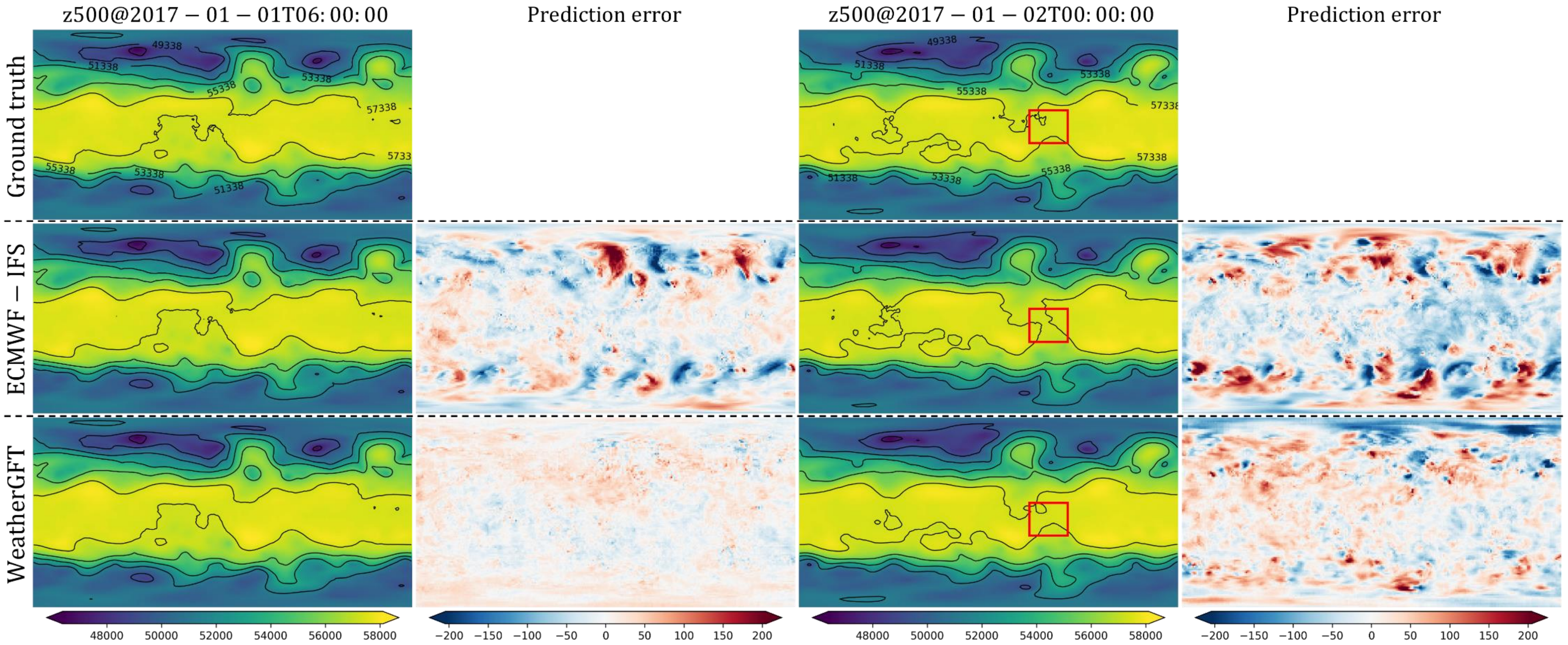
· Hyperparameter

Hyperparameter	Value
Max epoch	50
Batch size	4x8 (GPUs)
Learning rate	5e-4
Learning rate schedule	Cosine
Patch size	4x4
Embedding dimension	1024
MLP ratio	4
Activation function	GLUE
Input (0-hour)	[4, 69, 128, 256]
Output (1, 3, 6-hour)	[4, 3, 69, 128, 256]

Experiment

- **Skillful Medium-Range Forecasts by WeatherGFT**





Experiment

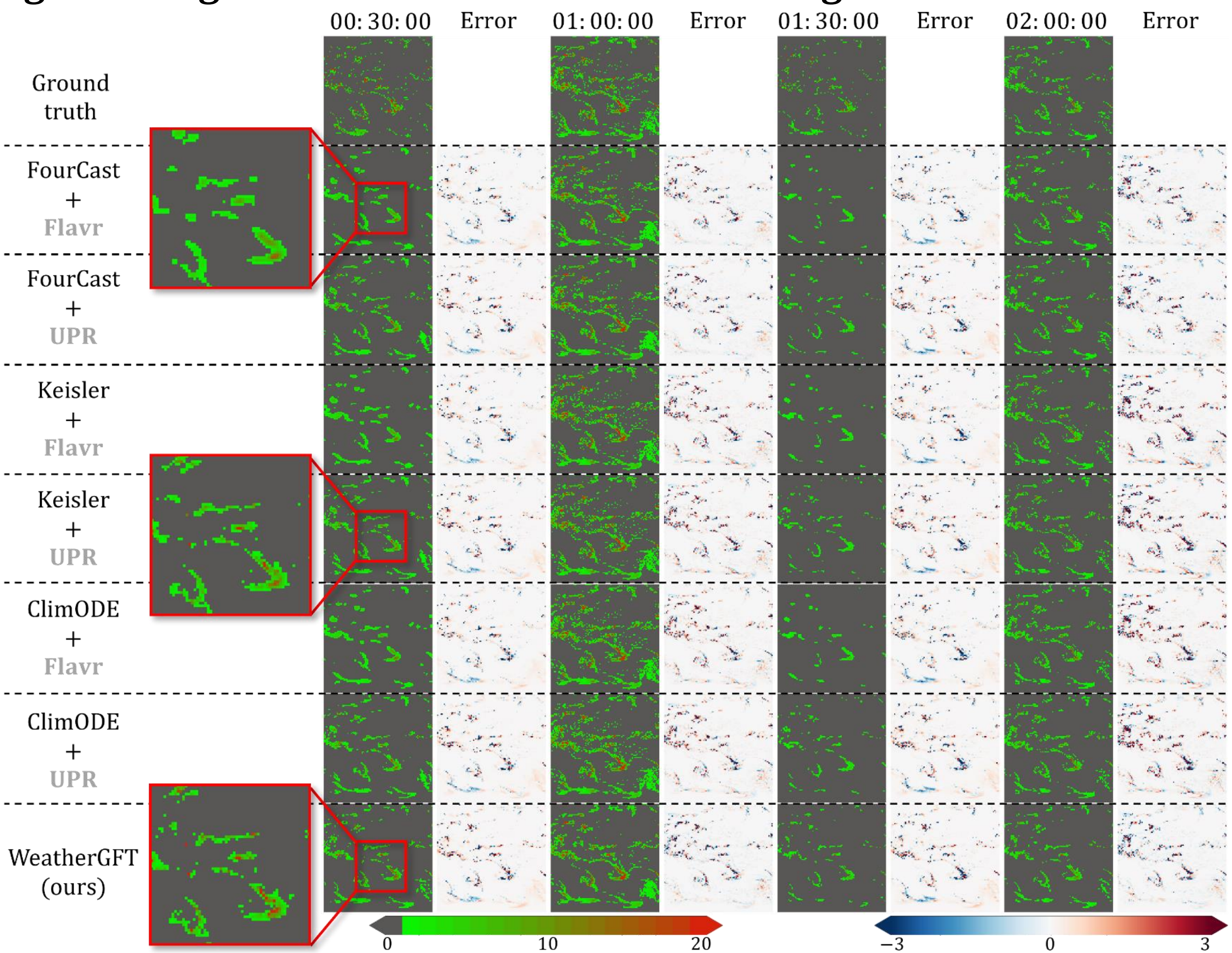
- **Generalizing to Fine-grained Time Scale for Nowcasting**

	30-min			60-min			90-min			120-min		
	CSI↑ @0.5	CSI↑ @1.5	RMSE↓ tp1h	CSI↑ @0.5	CSI↑ @1.5	RMSE↓ tp1h	CSI↑ @0.5	CSI↑ @1.5	RMSE↓ tp1h	CSI↑ @0.5	CSI↑ @1.5	RMSE↓ tp1h
FourCast+Flavr	0.26	0.09	0.67	0.61	0.49	0.24	0.25	0.09	**0.65**	0.37	0.26	0.46
FourCast+UPR	0.20	0.10	0.76	0.61	0.49	0.24	0.11	0.05	1.49	0.37	0.26	0.46
Keisler+Flavr	0.25	0.09	**0.66**	0.59	0.48	0.23	0.25	0.08	0.66	0.41	0.29	0.35
Keisler+UPR	0.26	0.13	0.69	0.59	0.48	0.23	0.26	0.13	0.68	0.41	0.29	0.35
ClimODE+Flavr	0.26	0.09	0.67	0.62	**0.51**	0.22	0.25	0.09	0.66	0.47	0.34	0.32
ClimODE+UPR	0.25	0.12	0.67	0.62	0.49	0.21	0.25	0.11	0.66	0.46	0.32	0.31
WeatherGFT(ours)	**0.28**	**0.17**	0.72	**0.62**	0.50	**0.21**	**0.28**	**0.16**	0.71	**0.54**	**0.40**	**0.27**

60-min and 120-min are trained lead times, while 30-min and 90-min are generalized lead times. Gray represents the results obtained through the frame interpolation model, purple indicates the results obtained through our unified model without interpolating. For precipitation nowcasting, CSI (Critical Success Index) is the most important metric.

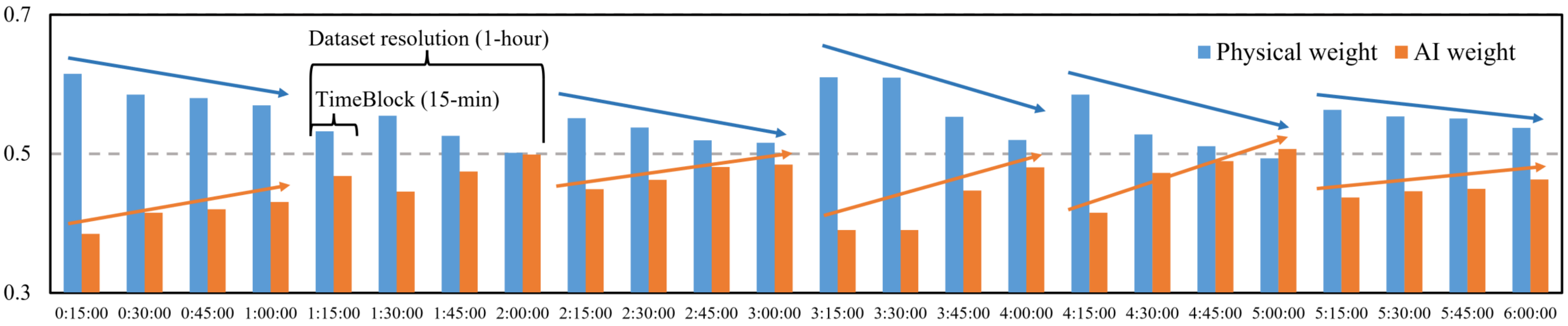
Experiment

- **Generalizing to Fine-grained Time Scale for Nowcasting**



Experiment

- **Forecasts can Benefit from Physics and AI via WeatherGFT**



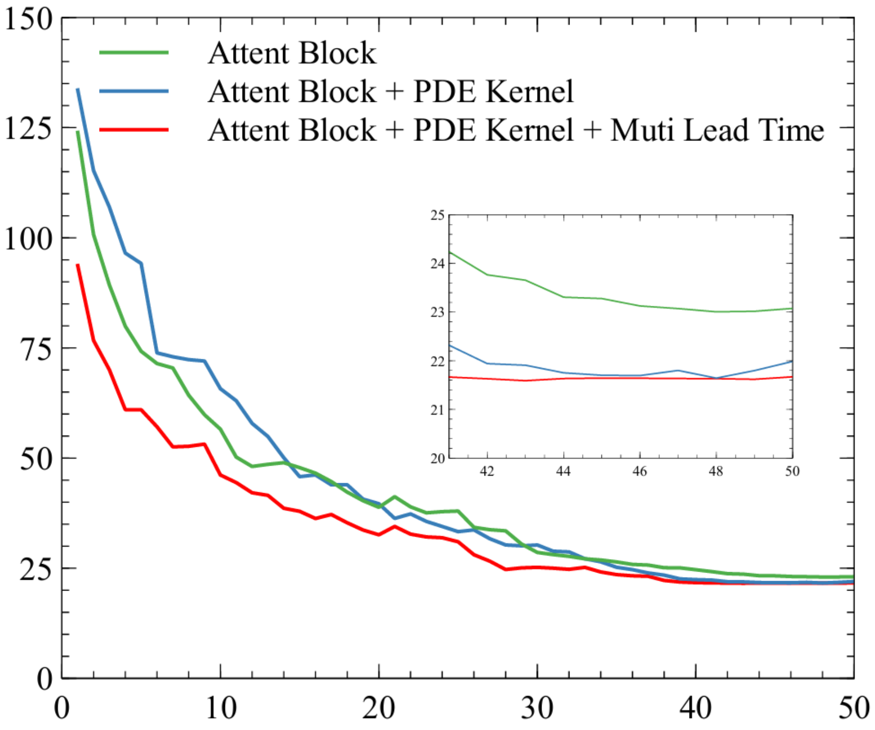
a) The physical weight of the vast majority of HybridBlocks is significantly higher than the weight of AI, which shows that in the process of simulating time evolution, the PDE kernel plays a more important role, while the Attention Block only plays a supportive correction role.
b) The physical weight gradually decreases while the weight of AI increases throughout each hour (dataset time resolution). This aligns with our underlying motivation, which acknowledges that errors may accumulate over time in the physics-based evolution.

Consequently, a greater emphasis on AI corrections becomes necessary to compensate for these accumulated errors.

Experiment

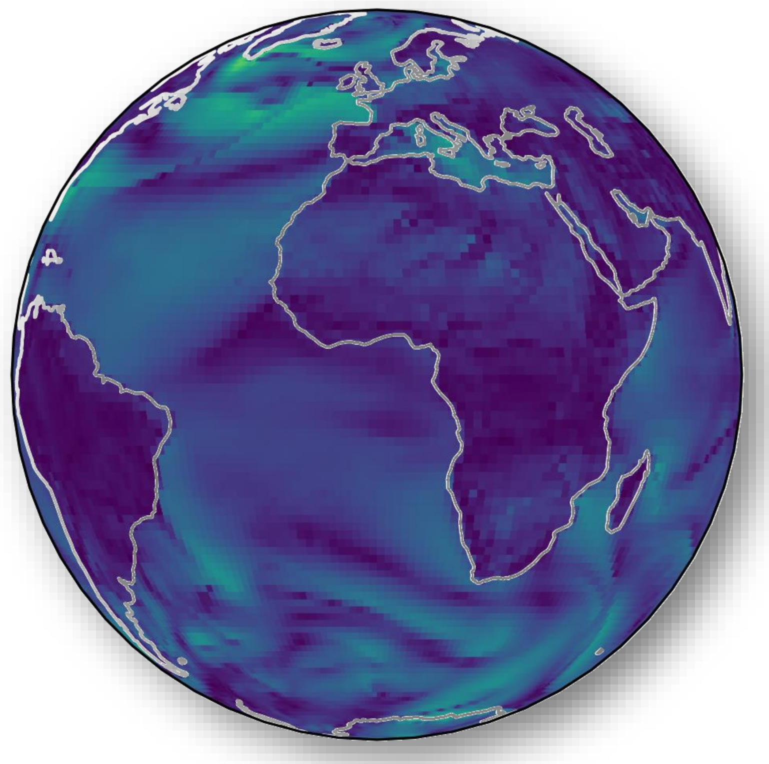
- **Ablation Studies**

	30-min nowcast	RMSE@1-h t2m↓	RMSE@1-h z500↓	RMSE@6-h t2m↓	RMSE@6-h z500↓	RMSE@3-d t2m↓	RMSE@3-d z500↓
Attent Block	×	0.52	18.76	0.73	24.21	1.23	157.9
+ PDE Kernel	✓	0.57	20.43	0.70	**21.78**	1.22	153.8
+ Muti Time	✓	**0.49**	**16.66**	**0.67**	21.80	**1.14**	**152.4**



Multiple lead time training accelerates con vergence and improves the accuracy of model prediction, as shown in the figure. We hypothesize that this phenomenon can be attributed to the loss backward from different lead times, which alleviates the issue of vanishing gradients, allowing the parameters of different layers to quickly warm up and improve the expression of the model.

Conclusion



Most existing data-driven weather forecast methods which operated as black-box models via purely performing data mapping are unable to generalize at finer temporal scale beyond the inherent time resolution of the training datasets due to the absence of the fine-grained physics modeling. This paper proposes a physics-AI hybrid model to solve this problem. Through the exquisitely designed PDE kernel, each block in the networks can simulate the evolution of physical variables at finer-gained time step, while AI plays the role of adaptive correction, which makes our model capable of generalizing predictions to a finer time scale beyond dataset. By employing our proposed multi-lead time training strategy, our model trained on an hourly dataset exhibits remarkable ability of generalized 30-minute forecasts, achieving SOTA performance in both medium-range forecast and precipitation nowcast.

Thanks.

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

上海人工智能实验室
Shanghai Artificial Intelligence Laboratory