ATMOSWEAVE: LEARNING ATMOSPHERIC DYNAMICS FOR GLOBAL AND REGIONAL WEATHER FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Data-driven weather forecasting learns atmospheric evolution directly from large-scale historical data and has achieved strong forecasting performance. Despite this progress, existing machine learning models still struggle to capture local interactions across spatial hierarchies, preserve directional variation, and efficiently model long-range dependencies. We introduce **AtmosWeave**, a graph-based weather forecasting framework that combines multiscale mesh propagation with efficient grouped linear attention. Specifically, Scale-Decoupled Directional Mesh Propagation separates within-level normalization from cross-level fusion and incorporates directional deviations into local propagation, while Geography- and State-Aware Grouped Linear Attention enables selective long-range interactions through source grouping and target-dependent routing. For high-resolution regional forecasting, we further introduce CrossWarp to couple global and regional representations through learned feature warping and selective global information injection, enabling the regional model to better leverage global context. Experiments demonstrate that AtmosWeave achieves strong global forecasting performance while effectively leveraging global information for high-resolution regional forecasting. Code is included with the submission and will be released publicly upon acceptance.

1 INTRODUCTION

Data-driven weather models have achieved strong forecasting performance by directly learning atmospheric evolution from large-scale historical data. Convolutional neural networks (CNNs) and neural operators model local or spectral structure on regular grids (Weyn et al., 2020; Pathak et al., 2022); graph-based models exchange information over spherical or multiresolution meshes (Keisler, 2022; Lam et al., 2023; Oskarsson et al., 2024; Gao et al., 2025; Price et al., 2025); and transformer-based models use attention to connect geographically distant locations (Bi et al., 2023; Chen et al., 2023b; 2026). Together, these advances have enabled data-driven models to produce skillful global medium-range forecasts across a broad set of surface and pressure-level variables.

Despite this progress, key challenges remain in modeling complex atmospheric dynamics. Graph neural networks (GNNs), exemplified by GraphCast, are well suited to global forecasting because multiresolution graph structures naturally represent the spherical geometry of the atmosphere (Lam et al., 2023). However, direct message aggregation on a multimesh can make each level's contribution dependent on its edge count, while isotropic reduction can obscure directional variation. Efficient long-range interaction also remains challenging: repeated local propagation connects distant regions only indirectly, whereas introducing full self-attention would incur quadratic computational complexity (Katharopoulos et al., 2020). These limitations motivate local and global models that preserve spatial-level, directional, and source-specific information for selective use by the receiving state.

Beyond these general challenges, high-resolution regional forecasting raises a further question: how best to exploit global background information. A bounded regional domain lacks the upstream and large-scale context available to a global model. Boundary forcing injects this context only at the lateral boundaries (Xu et al., 2025), and coarse-to-fine transfer typically starts from geographically aligned interpolation (Jackaman & Sutton, 2025). Nested global–regional graphs enable more direct cross-resolution exchange (Gao et al., 2025), but their connectivity is fixed. In each case, geography prescribes the global–regional correspondence rather than the evolving flow, even though the relevant

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

upstream region shifts with the flow. This motivates an adaptive coupling that retains geographic alignment as a reference while letting the regional state select the most relevant global features.

To address these challenges, we introduce ATMOSWEAVE, a graph-based framework for global and regional weather forecasting. Its global processor combines Scale-Decoupled Directional Mesh Propagation (SD-DMP) for level-decoupled, direction-aware local modeling with Geography- and State-Aware Grouped Linear Attention (GS-GLA) for efficient and selective long-range interaction. CrossWarp extends the framework to high-resolution regional forecasting by learning bounded correspondence between global and regional features and injecting global information through confidence-aware residual updates at multiple representation depths.

Our main contributions are as follows:

- We propose **AtmosWeave**, a model for global and regional weather forecasting that combines **SD-DMP** for scale-decoupled and direction-aware local propagation with **GS-GLA** for efficient and selective long-range interaction. Together, these components improve the modeling of multiscale and long-range atmospheric dynamics.
- We introduce **CrossWarp** to couple global and high-resolution regional forecasting through learned feature warping and selective global information injection, enabling the regional model to better leverage global context.
- Extensive experiments on global and high-resolution regional weather forecasting demonstrate strong performance in both settings, validating the effectiveness of the proposed framework.

2 RELATED WORK

2.1 DATA-DRIVEN WEATHER FORECASTING

Data-driven weather forecasting has advanced rapidly with recent progress in deep learning, while subsequent methods have evolved along several architectural paradigms (Dueben & Bauer, 2018; Scher, 2018; Rasp et al., 2020; Rasp & Thuerey, 2021). CNN-based models include cubed-sphere global forecasting (Weyn et al., 2020), while neural operators extend global spectral mixing from the Fourier neural operator to FourCastNet and spherical Fourier neural operators (Li et al., 2021; Pathak et al., 2022; Kurth et al., 2023; Bonev et al., 2023). Transformer-based systems build on self-attention (Vaswani et al., 2017) and include Pangu-Weather, FuXi, ClimaX, EMFormer, and Aurora (Bi et al., 2023; Chen et al., 2023b; Nguyen et al., 2023; Chen et al., 2026; Bodnar et al., 2025). Graph-based weather models draw on graph convolution and relational message passing (Kipf & Welling, 2017; Battaglia et al., 2018), extending mesh-based simulation (Pfaff et al., 2021) and early global graph forecasting (Keisler, 2022) to GraphCast, Graph-EFM, GenCast, OneForecast, and AIFS (Lam et al., 2023; Oskarsson et al., 2024; Price et al., 2025; Gao et al., 2025; Lang et al., 2024; Allen et al., 2025). Graph-based weather models represent spherical multiscale structure well, but conventional message passing may mix mesh levels, lose directional variation, and rely on repeated propagation for long-range interaction. AtmosWeave addresses these issues by normalizing messages within each mesh level before cross-level fusion, preserving directional variation during local propagation, and enabling state- and geography-aware long-range routing with linear complexity.

2.2 GLOBAL-TO-REGIONAL COUPLING

High-resolution regional forecasting resolves fine-scale atmospheric structures within a limited domain (Huang et al., 2026). Because regional evolution is influenced by upstream flow and planetary-scale circulation outside that domain, the regional model can benefit from global background information. Boundary forcing supplies global conditions at the domain edges, but relies on the regional dynamics to propagate them into the interior (Xu et al., 2025). OneForecast instead uses a neural nested grid to create direct connections between global and regional representations (Gao et al., 2025). However, fixed cross-resolution connections treat geographic alignment as feature correspondence and cannot adapt when relevant global patterns shift relative to regional features. CrossWarp preserves geographic alignment as a reference while transporting global features through bounded content-dependent correspondence and confidence-aware residual conditioning.

3 THEORETICAL BACKGROUND

3.1 PROBLEM FORMULATION

We represent the atmospheric state at time t as $X_t \in \mathbb{R}^{C\times H\times W}$ on a $H \times W$ latitude–longitude grid, with C the channel number. Given X_t, a deterministic predictor Φ_θ, parameterized by trainable parameters θ, estimates the state at the next forecast step. We generate a T-step forecast recursively:

$$\widehat{X}_{t+1} = \Phi_\theta(X_t), \qquad \widehat{X}_{t+\tau} = \Phi_\theta\Big(\widehat{X}_{t+\tau-1}\Big), \quad \tau = 2, \dots, T, \tag{1}$$

where $\widehat{X}_{t+\tau}$ is the prediction at lead step τ. AtmosWeave implements Φ_θ with a graph-based encoder–processor–decoder architecture.

3.2 GRAPH NEURAL NETWORKS

Graph-based weather forecasting models represent the atmosphere on a spherical multimesh and exchange information between locations via message passing (Lam et al., 2023). Since AtmosWeave builds on this paradigm, we first review conventional multimesh message passing. At stage $\ell \in \{0, \dots, L-1\}$ of an L-stage processor, let $z_i^\ell \in \mathbb{R}^{D_v}$ denote the feature of node i and $e_{ji}^\ell \in \mathbb{R}^{D_e}$ denote the feature of the directed edge from node j to node i. Here, D_v and D_e are the node- and edge-feature dimensions, respectively. Message passing updates the edge feature as:

$$e'_{ji} = e_{ji}^\ell + f_e\big([e_{ji}^\ell, z_j^\ell, z_i^\ell]\big)\,, \tag{2}$$

which represents the message from node j to i. Here, f_e is a learned network and $[\cdot]$ means feature concatenation. To examine how the receiver combines messages across multi-levels, we group the multimesh edges into S level-specific sets, with $\mathcal{E}_s$ denoting the directed edges at level s. For node i, let $\mathcal{N}_s(i) = \{j : (j, i) \in \mathcal{E}_s\}$ denote its incoming neighbors at level s, and $d_i^s = |\mathcal{N}_s(i)|$ the number of incoming edges; $\mathcal{S}_i = \{s : d_i^s > 0\}$ denotes levels connected to node i and $d_i = \sum_{s\in\mathcal{S}_i} d_i^s$ is its total number of incoming edges. At level s, the level-wise mean-aggregated message received by node i is $m_i^s = (d_i^s)^{-1} \sum_{j\in\mathcal{N}_s(i)} e'_{ji}$. Then averaging over all levels yields the aggregated message,

$$m_i = \frac{1}{d_i} \sum_{s\in\mathcal{S}_i} d_i^s m_i^s = \sum_{s\in\mathcal{S}_i} \frac{d_i^s}{d_i} m_i^s, \tag{3}$$

where d_i^s/d_i is a weight that sets how much level-s messages contribute to m_i and is determined solely by the topology of mesh.

$$z_i^{\ell+1} = z_i^\ell + f_v\big([m_i; z_i^\ell]\big)\,. \tag{4}$$

Notably, Eq. 3 reveals two limitations of mean aggregation: 1) a level's contribution is determined by its edge count rather than the content of its messages; 2) pooling all incoming messages into a single mean does not explicitly preserve their directional organization.

3.3 EFFICIENT GLOBAL INTERACTION

Accurate global weather prediction benefits from effective long-range interaction, which allows local atmospheric states to incorporate information from distant but dynamically connected regions. Self-attention models such dependencies by allowing each target node to retrieve information directly from all source nodes. Let $q_i, k_i \in \mathbb{R}^{d_k}$ and $v_i \in \mathbb{R}^{d_v}$ denote the query, key, and value of node i, where d_k and d_v are their respective feature dimensions. Standard self-attention assigns

$$a_{ij} = \frac{\exp\big(q_i^\top k_j/\sqrt{d_k}\big)}{\sum_{r=1}^N \exp\big(q_i^\top k_r/\sqrt{d_k}\big)}, \qquad o_i = \sum_{j=1}^N a_{ij} v_j, \tag{5}$$

where a_{ij} is the attention weight from source node j to target node i, and $o_i \in \mathbb{R}^{d_v}$ is the resulting target representation. Computing the N^2 pairwise scores incurs quadratic complexity.

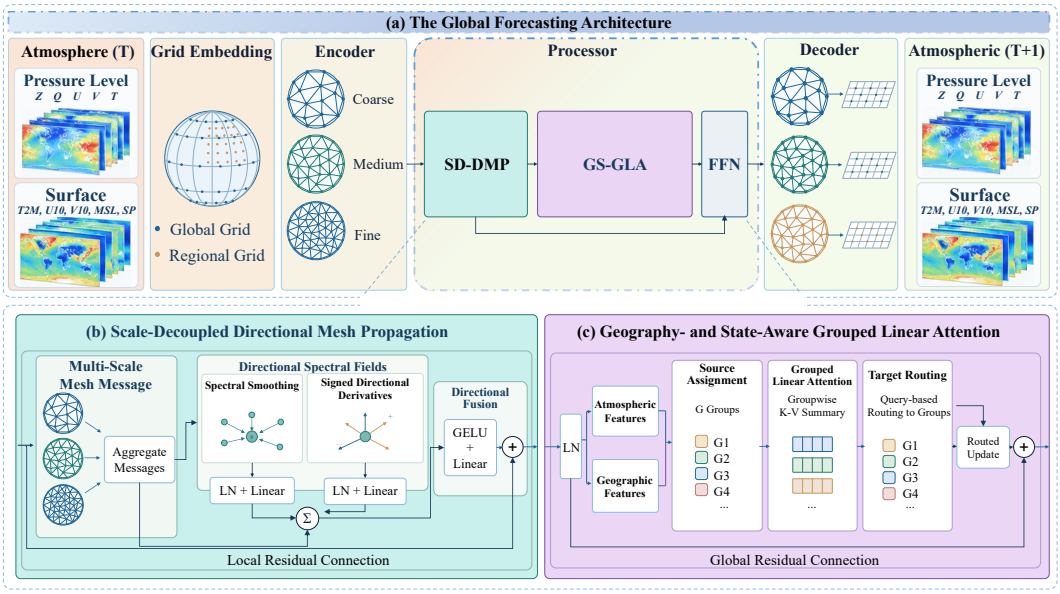

Figure 1: Architecture of AtmosWeave. (a) Global weather forecasting architecture includes the input atmospheric variables, grid embedding, grid-to-mesh encoder, processor, mesh-to-grid decoder, and output atmospheric variables; (b) the SD-DMP module for scale-decoupled and direction-aware local propagation; and (c) the GS-GLA module for state- and geography-aware grouped global interaction.

Kernel linear attention reduces this cost by applying a feature map $\varphi : \mathbb{R}^{d_k} \to \mathbb{R}^{d_\varphi}$ and rearranging the computation:

$$A = \sum_{j=1}^{N} \varphi(k_j) v_j^\top, \qquad b = \sum_{j=1}^{N} \varphi(k_j), \qquad o_i = \frac{\varphi(q_i)^\top A}{\max(\varphi(q_i)^\top b, \epsilon)}. \tag{6}$$

Here, $A \in \mathbb{R}^{d_\varphi \times d_v}$ and $b \in \mathbb{R}^{d_\varphi}$ are shared source statistics, while $\epsilon > 0$ provides numerical stability. This factorization reduces the complexity from quadratic to linear in N for fixed feature dimensions (Katharopoulos et al., 2020). However, compressing all source nodes into a single set of shared statistics may limit its ability to represent heterogeneous and spatially varying dependencies in complex atmospheric dynamics.

4 Methodology

Here we present the methodological contributions of AtmosWeave. We first outline the forecasting pipeline, then describe SD-DMP for local mesh propagation, GS-GLA for long-range interaction, and finally describe CrossWarp for global-to-regional coupling.

4.1 Overview

Figure 1 summarizes the global forecasting pipeline. Grid embedding maps X_t to grid features, which the encoder transfers to a multiscale mesh. Each processor stage applies SD-DMP, GS-GLA, and a feed-forward network (FFN); the decoder and prediction head map the result back to the grid to produce $\widehat{X}_{t+1}$. Architectural and processor details appear in Appendices D.1 and D.2.

4.2 Scale-Decoupled Directional Mesh Propagation

Conventional message aggregation (Eq. 3) suffers two drawbacks: 1) it mixes information from different mesh levels without distinguishing their contributions; and 2) obscures directional variation. Here, we propose SD-DMP to address these two issues.

Regarding the issue of distinguishing contribution, instead of using a fixed weight d_i^s/d_i for level s that solely depends on the topology of mesh, here, we adopt a learnable weight π_{is} to parameterize

the contributions of s-level messages to the aggregated message received by node i, accordingly reformulating Eq. 3 to the following:

$$m_i = \sum_{s \in \mathcal{S}_i} \bar{\pi}_{is} m_i^s. \tag{7}$$

To resolve the directional variation issue, SD-DMP uses the first K nontrivial eigenvectors of the normalized mesh Laplacian as graph coordinates, where K denotes the number of spectral directions. Let $\phi_k(i)$ denote the value of the kth eigenvector at node i.

For an incoming edge $j \to i$, the spectral-coordinate difference is $\phi_k(j) - \phi_k(i)$. Its magnitude defines unsigned weights for the edge-message contrast $c_i^k \in \mathbb{R}^{D_e}$, while the signed difference defines derivative weights for the node-state variation $r_i^k \in \mathbb{R}^{D_v}$. A learned low-rank fusion operator F_{dir} combines these signals into a directional correction $\delta_i \in \mathbb{R}^{D_e}$:

$$\delta_i = F_{dir}\big(\{c_i^k, r_i^k\}_{k=1}^K\big)\,. \tag{8}$$

The node update adds this correction to the level-decoupled message before applying the residual update:

$$u_i^\ell = z_i^\ell + f_v\big([m_i + \delta_i\,;\, z_i^\ell]\big)\,. \tag{9}$$

Here, $f_v : \mathbb{R}^{D_e+D_v} \to \mathbb{R}^{D_v}$ is a learned node-update network, and $u_i^\ell \in \mathbb{R}^{D_v}$ denotes the node representation produced by SD-DMP at stage ℓ. Although the graph coordinates are fixed, the extracted directional signals depend on the current edge features and node states, and therefore vary across input samples and processor stages. See Appendix D.2.1 for details.

4.3 Geography- and State-Aware Grouped Linear Attention

To complement local propagation without incurring the quadratic cost of standard self-attention, GS-GLA builds on kernel linear attention for efficient long-range interaction. Rather than compressing all sources into a single shared summary, it constructs G state- and geography-aware group summaries, one per learned group, and lets each target node route over them independently. Given the normalized local representation $\tilde{u}_i^\ell = \mathrm{LN}(u_i^\ell)$, where LN denotes layer normalization, we compute a node projection together with the query/key/value for each attention head $h \in \{1, \ldots, H_a\}$:

$$p_i = W_H \tilde{u}_i^\ell, \quad (q_i^h, k_i^h, v_i^h) = \big(\sigma(W_Q^h \tilde{u}_i^\ell), \sigma(W_K^h \tilde{u}_i^\ell), W_V^h \tilde{u}_i^\ell\big). \tag{10}$$

Here, H_a is the number of attention heads, d_h is the per-head dimension, W_H, W_Q^h, W_K^h, and W_V^h are learned projection matrices, $q_i^h, k_i^h, v_i^h \in \mathbb{R}^{d_h}$, $p_i \in \mathbb{R}^{D_v}$, and σ denotes the sigmoid function. The key and value describe the information that node i can contribute as a source, whereas the query describes the information it seeks as a target; p_i is retained for the final output modulation.

Group construction requires weights that assign source nodes to the G groups, while group retrieval requires weights that route each target across them. To make these weights depend on both atmospheric state and location, a geographic encoder maps $[\sin\theta_i, \cos\theta_i, \sin\lambda_i, \cos\lambda_i]$ to $\xi_i \in \mathbb{R}^{D_g}$, where θ_i and λ_i are the latitude and longitude of node i, respectively, and D_g is the geographic-embedding dimension. Source assignments are predicted from the keys and geography, while target routes are predicted independently from the queries and geography:

$$a_{ig} = \mathrm{softmax}_g\big(f_a([\mathrm{Concat}_h(k_i^h); \xi_i])\big)\,, \tag{11}$$

$$r_{ig} = \mathrm{softmax}_g\big(f_r([\mathrm{Concat}_h(q_i^h); \xi_i])\big)\,, \qquad g = 1, \ldots, G. \tag{12}$$

Here, f_a and f_r are learned assignment and routing networks, Concat_h concatenates features across attention heads, and $\mathrm{softmax}_g$ normalizes the scores across the G groups. The assignment a_{ig} controls how source node i contributes to group g, whereas the route r_{ig} controls how strongly target node i uses that group. Both are soft distributions, so a source may contribute to several summaries and a target may combine information from several groups.

For each head, the source assignments aggregate keys and values into group-specific statistics:

$$S_g^h = \sum_{j=1}^N a_{jg}\, k_j^h (v_j^h)^\top, \qquad z_g^h = \sum_{j=1}^N a_{jg}\, k_j^h. \tag{13}$$

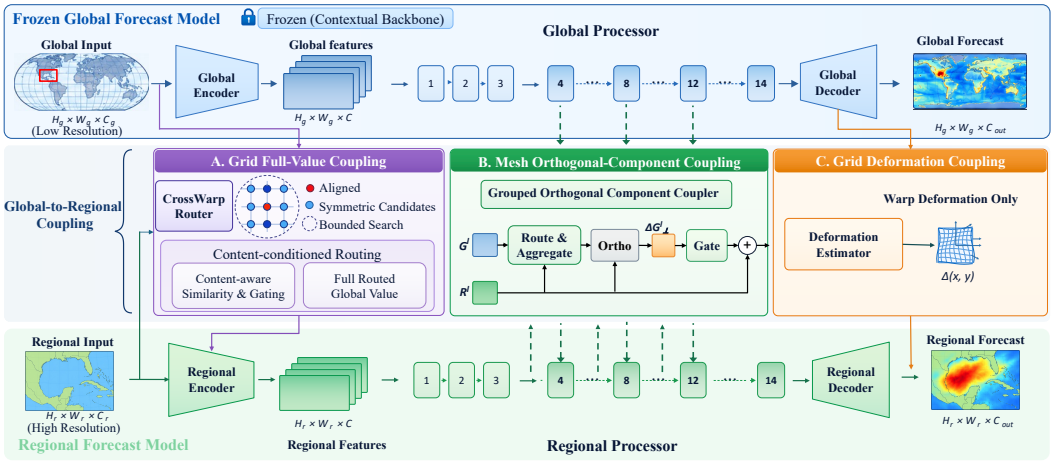

Figure 2: CrossWarp for global-to-regional forecasting. A frozen global backbone supplies features through three gated residual couplings: (A) full-value grid transport before regional mesh encoding, (B) group-wise orthogonal global component at selected processor stages, and (C) independently routed grid-feature correction after graph decoding and before regional variable prediction.

Here, $S_g^h \in \mathbb{R}^{d_h \times d_h}$ is the key–value statistic and $z_g^h \in \mathbb{R}^{d_h}$ is the key statistic for group g and head h. Each target uses its routing weights to combine the normalized group reads:

$$\bar{o}_i = \sum_{g=1}^{G} r_{ig}\, \mathrm{Concat}_h(o_{ig}^h), \qquad o_{ig}^h = \frac{(q_i^h)^\top S_g^h}{\max\big((q_i^h)^\top z_g^h, \epsilon\big)}. \tag{14}$$

Here, o_{ig}^h is the read from group g for target i and head h, $\bar{o}_i$ is the combined target representation, and $\epsilon > 0$ is a numerical-stability constant. The denominator $(q_i^h)^\top z_g^h$ normalizes each group read by the query-weighted key mass accumulated in that group. The combined target representation $\bar{o}_i$ is modulated by $p_i + b^\ell$, where $b^\ell \in \mathbb{R}^{D_v}$ is a learned stage-wise vector, and is then projected and added to u_i^ℓ via a residual connection. Source assignment and target routing therefore produce target-dependent long-range interaction without constructing an $N \times N$ attention matrix, so the model remains linear in N for fixed feature width and group count. Algorithm 1 gives full details.

4.4 CrossWarp: Adaptive Global-to-Regional Feature Coupling

CrossWarp introduces three global-to-regional couplings—at the regional encoder, selected processor stages, and decoder—that incorporate features from a frozen global model into a trainable regional one (see Figure 2). Each coupling uses geographic alignment as a reference and the receiving regional features to route information from nearby global candidates. The injected information is tailored to each representation: a confidence-weighted global value for encoder grid features, a group-wise orthogonal global component for processor mesh features, and a confidence-weighted routed-minus-aligned correction for decoder grid features.

We first describe the correspondence used by the encoder coupling and decoder coupling; each stage computes its own correspondence. Let G and R denote the global and regional grid feature fields at the stage under consideration. For a regional grid location p, let b_p be its geographically aligned coordinate on the global grid. CrossWarp constructs a bounded candidate set consisting of the geographically aligned location and additional directional candidates around it. Denote the resulting candidate set by $\mathcal{C}(p)$. For candidate $c \in \mathcal{C}(p)$, let $k_{p,c}$ and $v_{p,c}$ be the key and value sampled from G. A query q_p derived from R is compared with these keys. The resulting score $s_{p,c}$ combines the similarity with a learned candidate prior; its normalized weight $w_{p,c}$ and the routed global feature S_p are

$$w_{p,c} = \frac{\exp(s_{p,c})}{\sum_{c' \in \mathcal{C}(p)} \exp(s_{p,c'})}, \quad S_p = \sum_{c \in \mathcal{C}(p)} w_{p,c} v_{p,c}. \tag{15}$$

Let A_p be the global value sampled at b_p, and let χ_p be the confidence derived from the candidate-weight distribution. At the encoder, CrossWarp uses the confidence-weighted full-value transport

$$T_p = A_p + \chi_p(S_p - A_p). \tag{16}$$

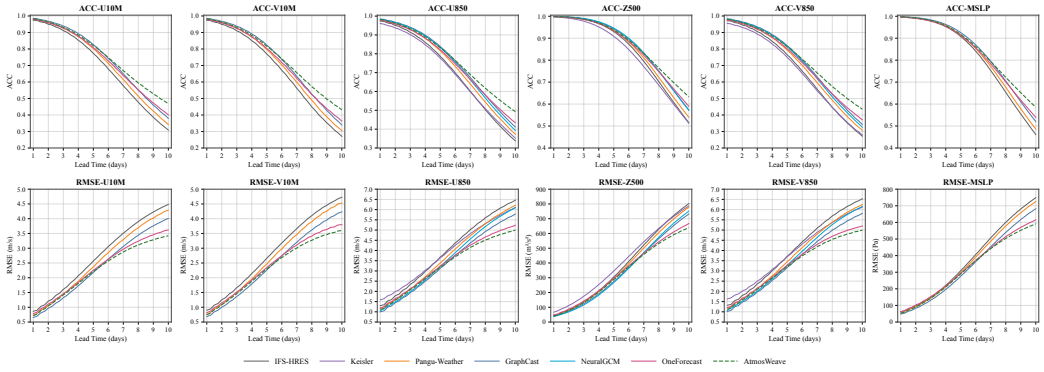

Figure 3: Compare ACC/RMSE among IFS-HRES, Keisler, Pangu-Weather, GraphCast, NeuralGCM, OneForecast, and ATMOSWEAVE across different lead times, with baselines from WeatherBench 2.

The transported value T_p is projected, gated, and added residually to the regional grid feature before regional grid-to-mesh encoding. Thus, geographic alignment remains the reference while learned routing adjusts the global information supplied to the regional encoder. At selected processor stages, CrossWarp instead routes features between the global and regional meshes using a sparse candidate set for each regional mesh node. For regional node i and channel group r, let $g^{\ell}_{i,r}$ be the projected and normalized transported global feature, and let $\widehat{v}^{\ell}_{i,r}$ be the normalized regional feature group. CrossWarp removes the component of $g^{\ell}_{i,r}$ parallel to $\widehat{v}^{\ell}_{i,r}$ and retains the orthogonal global component:

$$I^{\ell}_{i,r} = g^{\ell}_{i,r} - \langle g^{\ell}_{i,r}, \widehat{v}^{\ell}_{i,r} \rangle \widehat{v}^{\ell}_{i,r}, \tag{17}$$

for residual fusion. Here, $\langle \cdot, \cdot \rangle$ is the feature-space inner product. The processor orthogonalization is defined in learned feature space and does not imply statistical independence. The regional reference is normalized with stopped gradients. After the global and regional mesh-to-grid decoders, CrossWarp computes a new correspondence from their grid features rather than reusing the encoder route. It projects and gates only the confidence-weighted difference between the routed and geographically aligned global values, then adds this correction to the regional grid feature before prediction. The decoder update and normalization are detailed in Appendix D.3.

4.5 COMPLEXITY ANALYSIS

Let B be the batch size, N and M the mesh node and edge counts (Section 3.2), and D an upper bound on feature width. With K graph directions and bottleneck width B_d, SD-DMP costs $O(BMKD + BNKB_dD)$. With G groups and per-head width d_h, GS-GLA costs $O(BND^2 + BNGDd_h)$ without constructing an $N \times N$ attention matrix. CrossWarp routes over bounded candidate sets at each regional location, so its cost is linear in the number of regional grid and mesh points. For fixed widths, group and direction counts, and candidate-set sizes, all three components scale linearly with spatial size. Measured costs are reported in Appendix E.

5 EXPERIMENTS

5.1 DATASETS

Global dataset. We conduct the global forecasting experiments using ERA5 reanalysis fields (Hersbach et al., 2020). The dataset spans 1979–2020 on a 1.5° latitude–longitude grid with spatial shape 121×240. Atmospheric states are sampled every six hours, and each model step predicts the state six hours ahead. **Regional dataset.** The regional benchmark uses an ERA5 East Asia subset spanning 20°N–28°N and 110°E–126°E at 0.25° resolution. The dataset spans 1993–2021 with a six-hour temporal interval and contains three near-surface variables. Dataset and preprocessing details are given in Appendix A, and metric definitions in Appendix B.

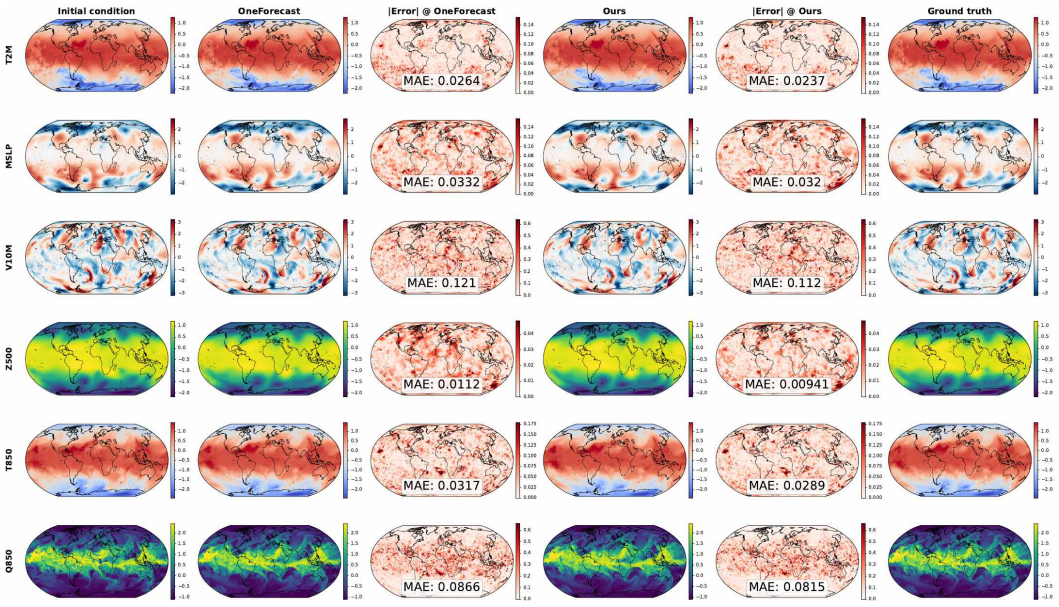

Figure 4: Visualization of 24-hour global weather forecasts for different atmospheric variables.

Table 1: Benchmark Ours against five baselines on Global Weather Forecasting. Lower RMSE (normalized, ↓) and higher ACC (denormalized, ↑) indicate better performance. All competitors are from EMFormer (Chen et al., 2026).

	6 h		1 day		4 days		7 days		10 days	
Model	RMSE↓	ACC↑	RMSE↓	ACC↑	RMSE↓	ACC↑	RMSE↓	ACC↑	RMSE↓	ACC↑
Pangu-Weather (Bi et al., 2023)	0.0826	0.9876	0.1571	0.9581	0.3380	0.8167	0.5092	0.5738	0.6215	0.3542
GraphCast (Lam et al., 2023)	0.0626	0.9928	0.1304	0.9705	0.2861	0.8705	0.4597	0.6692	0.6009	0.4275
FuXi (Chen et al., 2023b)	0.0987	0.9820	0.1708	0.9511	0.4128	0.7379	0.5972	0.4446	0.6981	0.2391
OneForecast (Gao et al., 2025)	**0.0549**	0.9943	0.1231	0.9737	0.2732	0.8825	0.4468	0.6888	0.5918	0.4457
EMFormer (Chen et al., 2026)	0.0626	0.9931	0.1219	0.9749	0.2673	0.8845	**0.4327**	0.6978	**0.5719**	0.4614
Ours	0.0610	**0.9953**	**0.1148**	**0.9772**	**0.2598**	**0.8957**	0.4366	**0.7092**	0.5866	**0.4680**

5.2 Global Forecasting Results

To account for differences in variable scale, Table 1 reports normalized RMSE and ACC. AtmosWeave achieves the highest ACC at every lead time and competitive normalized RMSE. Fig. 3 compares variable-wise ACC and RMSE with WeatherBench 2 baselines; additional variables appear in Appendix F.3. In the 24-hour case (Fig. 4), AtmosWeave more closely matches the ground truth and has lower displayed MAE than OneForecast.

5.3 High-Resolution Regional Forecasting

Table 2 summarizes the regional forecasting performance over East Asia at 0.25° resolution. For a fair comparison, we adopt the same RMSE/ACC computation used in STORM. Our method consistently outperforms all compared approaches across all lead times for both RMSE and ACC. The performance gains are maintained from short-range to extended forecasts, indicating that our framework effectively captures regional atmospheric evolution while preserving predictive accuracy over longer lead times. These results further demonstrate the effectiveness of our global-to-regional forecasting design for high-resolution regional prediction. We also present qualitative comparisons of 24-hour forecasts in Fig. 5. Compared with Direct Training and OneForecast, CrossWarp produces predictions that are more consistent with the ground truth for U10M, V10M, and T2M, while exhibiting substantially reduced spatial errors. The improvement is particularly evident in the wind fields, where CrossWarp better preserves the underlying spatial structures and suppresses localized error concentrations.

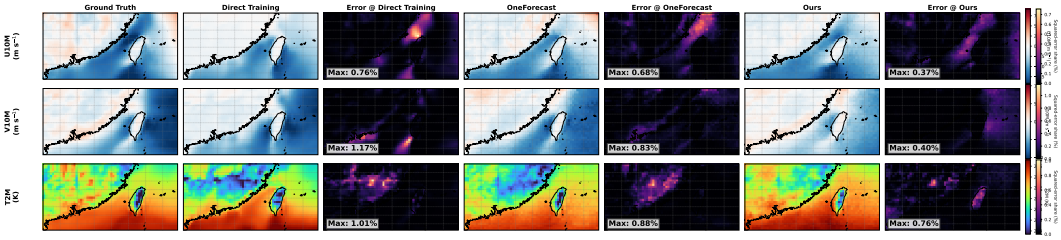

Figure 5: Regional 24-hour forecasts of U10M, V10M, and T2M. Columns show ground truth, Direct Training, OneForecast, and Ours, followed by the corresponding error maps.

Table 2: Regional high-resolution forecasting over East Asia at $0.25°$. Bold and underlined values mark the best and second-best results. An asterisk (*) denotes a model retrained on this regional dataset. All other competitors are collected from STORM (Huang et al., 2026).

Method	6 h		1 day		4 days		7 days		10 days	
	RMSE↓	ACC↑	RMSE↓	ACC↑	RMSE↓	ACC↑	RMSE↓	ACC↑	RMSE↓	ACC↑
UNet (Ronneberger et al., 2015)	1.528	0.873	2.696	0.661	6.994	0.291	11.500	0.248	15.620	0.252
SimVP (Gao et al., 2022)	0.947	0.939	1.619	0.843	3.110	0.554	3.603	0.476	3.891	0.426
FuXi (Chen et al., 2023b)	0.913	0.941	1.677	0.830	3.253	0.547	3.632	0.500	3.797	0.466
Pangu-Weather (Bi et al., 2023)	0.989	0.876	1.794	0.702	3.557	0.440	4.235	0.375	4.613	0.368
FCN (Pathak et al., 2022)	1.411	0.933	2.466	0.816	4.186	0.498	5.176	0.425	6.018	0.403
Triton (Wu et al., 2025)	0.927	0.941	1.808	0.817	3.689	0.480	4.687	0.347	5.433	0.251
STORM (Huang et al., 2026)	0.874	0.945	1.560	0.847	2.866	0.601	<u>3.322</u>	0.519	<u>3.510</u>	<u>0.499</u>
OneForecast* (Gao et al., 2025)	<u>0.852</u>	<u>0.977</u>	<u>1.558</u>	<u>0.901</u>	<u>2.806</u>	<u>0.670</u>	3.360	<u>0.540</u>	3.799	0.448
Ours	**0.834**	**0.978**	**1.545**	**0.904**	**2.762**	**0.673**	**3.182**	**0.580**	**3.339**	**0.540**

Table 3: Ablation studies for global and high-resolution regional forecasting. The best results are in bold.

Ablation Studies	**RMSE↓**	**ACC↑**
High-resolution Regional Forecasts		
w/o CROSSWARP	3.704	0.4433
w/o Adaptive CROSSWARP	3.152	0.5481
w CROSSWARP	**2.762**	**0.6730**
Low-resolution Global Forecasts		
Baseline	0.3225	0.8214
ATMOSWEAVE w/o SD-DMP	0.2832	0.8744
ATMOSWEAVE w/o GS-GLA	0.2967	0.8679
ATMOSWEAVE w/o GS-G	0.2721	0.8868
ATMOSWEAVE	**0.2654**	**0.8875**

Table 3 reports four-day ablations for both forecasting tasks. In the global block, the baseline excludes SD-DMP and GS-GLA. Removing either operator from the full model increases RMSE and lowers ACC. The *w/o GS-G* variant removes the grouping mechanism from GS-GLA while retaining linear attention. Its RMSE increases from 0.2654 to 0.2721, while ACC decreases from 0.8875 to 0.8868. These comparisons support both operators and the grouping mechanism.

In the regional block, *w/o* CROSSWARP has no global coupling, whereas *w/o Adaptive* CROSSWARP uses fixed geographic alignment. The latter reduces RMSE from 3.704 to 3.152 and raises ACC from 0.4433 to 0.5481. Full CROSSWARP further improves both metrics to 2.762 and 0.6730, showing the benefit of adaptive correspondence.

5.4 ABLATION STUDY

6 CONCLUSION

We presented **ATMOSWEAVE**, a unified graph-based framework that combines **Scale-Decoupled Directional Mesh Propagation (SD-DMP)** for multiscale local dynamics with **Geography- and State-Aware Grouped Linear Attention (GS-GLA)** for grouped, target-dependent long-range interactions while maintaining linear spatial complexity. For high-resolution regional forecasting, **CROSSWARP** models global-to-regional coupling as a learned feature flow, enabling the regional model to adaptively incorporate global context. Experiments on global and $0.25°$ East Asia regional forecasting demonstrate strong global forecasting performance and consistent regional improvements across multiple lead times, while ablation studies validate the complementary contributions of the proposed components.

AI USE STATEMENT

Generative AI tools were used to retrieve and discover related work, support research ideation and execution, draft selected sections, and aid and polish the writing. The authors reviewed the AI-assisted content and remain responsible for the scientific claims, references, results, and final manuscript.

REPRODUCIBILITY STATEMENT

We have taken care to ensure that our experiments can be replicated. The architecture of our model is described in Section 4 and Appendix D, while training details are given in Appendix C. All datasets are publicly accessible, and the preprocessing steps are documented in Appendix A.2. Furthermore, we provide anonymized code and usage instructions as supplementary material to support reproducibility of our results.

REFERENCES

Anna Allen, Stratis Markou, Will Tebbutt, et al. End-to-end data-driven weather prediction. *Nature*, 641:1172–1179, 2025. doi: 10.1038/s41586-025-08897-0.

Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018. URL https://arxiv.org/abs/1806.01261.

Kaifeng Bi, Lingxi Xie, Hengheng Zhang, Xin Chen, Xiaotao Gu, and Qi Tian. Accurate medium-range global weather forecasting with 3d neural networks. *Nature*, 619:533–538, 2023. doi: 10.1038/s41586-023-06185-3.

Cristian Bodnar, Wessel P. Bruinsma, Ana Lucic, et al. A foundation model for the earth system. *Nature*, 641:1180–1187, 2025. doi: 10.1038/s41586-025-09005-y.

Boris Bonev, Thorsten Kurth, Christian Hundt, Jaideep Pathak, Maximilian Baust, Karthik Kashinath, and Anima Anandkumar. Spherical fourier neural operators: Learning stable dynamics on the sphere. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 2806–2823. PMLR, 2023. URL https://proceedings.mlr.press/v202/bonev23a.html.

Hao Chen, Tao Han, Song Guo, Jie Zhang, Yonghan Dong, Yunlong Yu, and Lei Bai. VA-MoE: Variables-adaptive mixture of experts for incremental weather forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7915–7924, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Chen_VA-MoE_Variables-Adaptive_Mixture_of_Experts_for_Incremental_Weather_Forecasting_ICCV_2025_paper.html.

Hao Chen, Tao Han, Jie Zhang, Song Guo, Fenghua Ling, and Lei Bai. EMFormer: Efficient multi-scale transformer for accumulative context weather forecasting. *arXiv preprint arXiv:2602.01194*, 2026. URL https://arxiv.org/abs/2602.01194.

Kang Chen, Tao Han, Junchao Gong, Lei Bai, Fenghua Ling, Jing-Jia Luo, Xi Chen, Leiming Ma, Tianning Zhang, Rui Su, Yuanzheng Ci, Bin Li, Xiaokang Yang, and Wanli Ouyang. FengWu: Pushing the skillful global medium-range weather forecast beyond 10 days lead. *arXiv preprint arXiv:2304.02948*, 2023a. URL https://arxiv.org/abs/2304.02948.

Lei Chen, Xiaohui Zhong, Feng Zhang, et al. FuXi: A cascade machine learning forecasting system for 15-day global weather forecast. *npj Climate and Atmospheric Science*, 6:190, 2023b. doi: 10.1038/s41612-023-00512-1.

Peter D. Dueben and Peter Bauer. Challenges and design choices for global weather and climate models based on machine learning. *Geoscientific Model Development*, 11:3999–4009, 2018. doi: 10.5194/gmd-11-3999-2018.

Yuan Gao, Hao Wu, Ruiqi Shu, Huanshuo Dong, Fan Xu, Rui Ray Chen, Yibo Yan, Qingsong Wen, Xuming Hu, Kun Wang, Jiahao Wu, Li Qing, Hui Xiong, and Xiaomeng Huang. OneForecast: A universal framework for global and regional weather forecasting. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 18658–18697. PMLR, 2025. URL https://proceedings.mlr.press/v267/gao25r.html.

Zhangyang Gao, Cheng Tan, Lirong Wu, and Stan Z. Li. SimVP: Simpler yet better video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3170–3180, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Gao_SimVP_Simpler_Yet_Better_Video_Prediction_CVPR_2022_paper.html.

Hans Hersbach, Bill Bell, Paul Berrisford, et al. The ERA5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730):1999–2049, 2020. doi: 10.1002/qj.3803.

Qihe Huang, Zhengyang Zhou, Yangze Li, Jiaming Ma, Kuo Yang, Binwu Wang, Xu Wang, and Yang Wang. STORM: Synergistic cross-scale spatio-temporal modeling for weather forecasting. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=JLF6XDnscF.

James Jackaman and Oliver J. Sutton. Improving regional weather forecasts with neural interpolation. *Journal of Scientific Computing and Machine Learning*, 1:52–59, 2025. URL https://scml.jp/2025/paper/8.pdf.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5156–5165. PMLR, 2020.

Ryan Keisler. Forecasting global weather with graph neural networks. *arXiv preprint arXiv:2202.07575*, 2022. URL https://arxiv.org/abs/2202.07575.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=SJU4ayYgl.

Thorsten Kurth, Shashank Subramanian, Peter Harrington, Jaideep Pathak, Morteza Mardani, David Hall, Andrea Miele, Karthik Kashinath, and Animashree Anandkumar. FourCastNet: Accelerating global high-resolution weather forecasting using adaptive fourier neural operators. In *Proceedings of the Platform for Advanced Scientific Computing Conference*, pp. 1–11, 2023. doi: 10.1145/3592979.3593412.

Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, et al. Learning skillful medium-range global weather forecasting. *Science*, 382(6677):1416–1421, 2023. doi: 10.1126/science.adi2336.

Simon Lang, Mihai Alexe, Matthew Chantry, et al. AIFS—ECMWF's data-driven forecasting system. *arXiv preprint arXiv:2406.01465*, 2024. URL https://arxiv.org/abs/2406.01465.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=c8P9NQVtmnO.

Tung Nguyen, Johannes Brandstetter, Ashish Kapoor, Jayesh K. Gupta, and Aditya Grover. ClimaX: A foundation model for weather and climate. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 25904–25938. PMLR, 2023. URL https://proceedings.mlr.press/v202/nguyen23a.html.

Joel Oskarsson, Tomas Landelius, Marc Peter Deisenroth, and Fredrik Lindsten. Probabilistic weather forecasting with hierarchical graph neural networks. *arXiv preprint arXiv:2406.04759*, 2024. URL https://arxiv.org/abs/2406.04759.

Jaideep Pathak, Shashank Subramanian, Peter Harrington, et al. FourCastNet: A global data-driven high-resolution weather model using adaptive fourier neural operators. *arXiv preprint arXiv:2202.11214*, 2022. URL https://arxiv.org/abs/2202.11214.

Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W. Battaglia. Learning mesh-based simulation with graph networks. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=roNqYL0_XP.

Ilan Price, Alvaro Sanchez-Gonzalez, Ferran Alet, et al. Probabilistic weather forecasting with machine learning. *Nature*, 637:84–90, 2025. doi: 10.1038/s41586-024-08252-9.

Stephan Rasp and Nils Thuerey. Data-driven medium-range weather prediction with a ResNet pretrained on climate simulations: A new model for WeatherBench. *Journal of Advances in Modeling Earth Systems*, 13(2):e2020MS002405, 2021. doi: 10.1029/2020MS002405.

Stephan Rasp, Peter D. Dueben, Sebastian Scher, Jonathan A. Weyn, Mohamed Mouatadid, and Nils Thuerey. WeatherBench: A benchmark data set for data-driven weather forecasting. *Journal of Advances in Modeling Earth Systems*, 12(11):e2020MS002203, 2020. doi: 10.1029/2020MS002203.

Stephan Rasp, Stephan Hoyer, Alexander Merose, et al. WeatherBench 2: A benchmark for the next generation of data-driven global weather models. *Journal of Advances in Modeling Earth Systems*, 16(6):e2023MS004019, 2024. doi: 10.1029/2023MS004019.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pp. 234–241, 2015. URL https://arxiv.org/abs/1505.04597.

Sebastian Scher. Toward data-driven weather and climate forecasting: Approximating a simple general circulation model with deep learning. *Geophysical Research Letters*, 45(22):12616–12622, 2018. doi: 10.1029/2018GL080704.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Jonathan A. Weyn, Dale R. Durran, and Rich Caruana. Improving data-driven global weather prediction using deep convolutional neural networks on a cubed sphere. *Journal of Advances in Modeling Earth Systems*, 12(9):e2020MS002109, 2020. doi: 10.1029/2020MS002109. URL https://doi.org/10.1029/2020MS002109.

Hao Wu, Yuan Gao, Ruiqi Shu, Kun Wang, Ruijian Gou, Chuhan Wu, Xinliang Liu, Juncai He, Shuhao Cao, Junfeng Fang, Xingjian Shi, Feng Tao, Qi Song, Shengxuan Ji, Yanfei Xiang, Yuze Sun, Jiahao Li, Fan Xu, Huanshuo Dong, Haixin Wang, Fan Zhang, Penghao Zhao, Xian Wu, Qingsong Wen, Deliang Chen, and Xiaomeng Huang. Advanced long-term earth system forecasting by learning the small-scale nature. *arXiv preprint arXiv:2505.19432*, 2025. URL https://arxiv.org/abs/2505.19432v1.

Pengbo Xu, Xiaogu Zheng, Tianyan Gao, Yu Wang, Junping Yin, Juan Zhang, Xuanze Zhang, San Luo, Zhonglei Wang, Zhimin Zhang, Xiaoguang Hu, and Xiaoxu Chen. An artificial intelligence-based limited area model for forecasting of surface meteorological variables. *Communications Earth & Environment*, 6:372, 2025. doi: 10.1038/s43247-025-02347-5.

Appendix

Table of Contents

A Data Details **14**

A.1 Dataset 14

A.2 Data Processing 14

B Metric Details **15**

C Training Configuration **16**

C.1 Global Training 16

C.2 Regional Training 16

D Architecture and Algorithms **16**

D.1 Global Forecasting Architecture 16

D.2 Global Processor Details and Pseudocode 17

D.3 CrossWarp Architecture 18

D.4 Tensor Shapes 21

E Efficiency Analysis **22**

F Additional Results **23**

F.1 Ablation Analysis 23

F.2 Finetuning Results 23

F.3 Additional Variable-wise Comparison 23

F.4 Forecast Visualizations 24

Table 4: Atmospheric variables and dataset configurations for global and regional weather forecasting. The 13 pressure levels are 50, 100, 150, 200, 250, 300, 400, 500, 600, 700, 850, 925, and 1000 hPa. Single-level variables correspond to surface or near-surface fields.

Name	Description	Levels	Resolution	Lat–lon range	Time
Low-resolution Global Weather Forecasting					
Z	Geopotential	13	1.5° (121×240)	Global	1979–2020
Q	Specific humidity	13	1.5° (121×240)	Global	1979–2020
T	Temperature	13	1.5° (121×240)	Global	1979–2020
U	Zonal wind	13	1.5° (121×240)	Global	1979–2020
V	Meridional wind	13	1.5° (121×240)	Global	1979–2020
U10M	10 m zonal wind	1	1.5° (121×240)	Global	1979–2020
V10M	10 m meridional wind	1	1.5° (121×240)	Global	1979–2020
T2M	2 m temperature	1	1.5° (121×240)	Global	1979–2020
MSLP	Mean sea-level pressure	1	1.5° (121×240)	Global	1979–2020
SP	Surface pressure	1	1.5° (121×240)	Global	1979–2020
High-resolution Regional Weather Forecasting					
U10M	10 m zonal wind	1	0.25°	20°N–28°N, 110°E–126°E	1993–2021
V10M	10 m meridional wind	1	0.25°	20°N–28°N, 110°E–126°E	1993–2021
T2M	2 m temperature	1	0.25°	20°N–28°N, 110°E–126°E	1993–2021

A Data Details

A.1 Dataset

In this section, we describe the datasets used for global and regional forecasting. For global forecasting, we use ERA5 reanalysis fields (Hersbach et al., 2020) distributed through WeatherBench 2 (Rasp et al., 2024). The selected records span 1979–2020 at six-hour intervals on a 1.5° latitude–longitude source grid (121×240). We use 1979–2019 for training and 2020 for testing. Each atmospheric state contains 70 channels: geopotential (Z), specific humidity (Q), temperature (T), zonal wind (U), and meridional wind (V) at 13 pressure levels (50, 100, 150, 200, 250, 300, 400, 500, 600, 700, 850, 925, and 1000 hPa), plus five surface fields: U10M, V10M, T2M, MSLP, and surface pressure (SP).

For regional forecasting, we use ERA5 fields available from the Climate Data Store (CDS; `https://cds.climate.copernicus.eu/`). The data cover the East Asia domain 20°N–28°N and 110°E–126°E at 0.25° resolution and six-hour intervals during 1993–2021. The regional model uses 1993–2017 for training and 2020–2021 for testing. Its inputs and targets contain U10M, V10M, and T2M. Table 4 summarizes the variables and coverage of both datasets.

A.2 Data Processing

Both datasets are stored in annual HDF5 files with arrays indexed by time, channel, latitude, and longitude. Global preprocessing retains all 70 channels, the first 121 latitude rows, and all 240 longitude columns of the source grid, excluding its southernmost latitude row. With six-hour sampling and no history input, the global state X_t is paired with X_{t+1} for one-step prediction.

The regional fields are selected on the model domain at their native 0.25° spacing, without spatial interpolation. U10M, V10M, and T2M are read in that channel order. Annual regional files are paired with the corresponding global context files by year and time index, and files with unequal time lengths are rejected. Each regional state is paired with concurrent global context; subsequent regional states at six-hour intervals form the forecast targets.

At load time, NaN values in inputs and targets are replaced with zero. Each channel c is standardized as $(x_c - \mu_c)/\sigma_c$, using channel-wise means μ_c and standard deviations σ_c from the respective training split: 1979–2019 for global fields and 1993–2017 for regional fields. Regional inputs and targets use the regional statistics, whereas paired global context uses the global statistics. The corresponding statistics are reused across data splits.

B METRIC DETAILS

We evaluate the forecasting performance of ATMOSWEAVE using root mean square error (RMSE) and anomaly correlation coefficient (ACC). RMSE measures the magnitude of prediction errors, while ACC measures the agreement between predicted and observed anomalies. The metric definitions are given below.

Let $\widehat{X}^t_{i,j,c}$ and $X^t_{i,j,c}$ be the prediction and ground truth for variable c at forecast lead t and grid point (i, j). Using the training-period mean μ_c and standard deviation σ_c, define

$$Z^t_{i,j,c} = \frac{X^t_{i,j,c} - \mu_c}{\sigma_c}, \qquad \widehat{Z}^t_{i,j,c} = \frac{\widehat{X}^t_{i,j,c} - \mu_c}{\sigma_c}. \tag{18}$$

The latitude-weighted normalized RMSE and ACC are

$$\mathrm{NRMSE}_c(t) = \sqrt{\frac{1}{N_{\mathrm{lat}} N_{\mathrm{lon}}} \sum_{i=1}^{N_{\mathrm{lat}}} \sum_{j=1}^{N_{\mathrm{lon}}} L_i \big(\widehat{Z}^t_{i,j,c} - Z^t_{i,j,c}\big)^2}, \tag{19}$$

$$\mathrm{ACC}_c(t) = \frac{\sum_{i=1}^{N_{\mathrm{lat}}} \sum_{j=1}^{N_{\mathrm{lon}}} L_i \widetilde{\widehat{X}}^t_{i,j,c} \widetilde{X}^t_{i,j,c}}{\sqrt{\sum_{i=1}^{N_{\mathrm{lat}}} \sum_{j=1}^{N_{\mathrm{lon}}} L_i (\widetilde{\widehat{X}}^t_{i,j,c})^2} \sqrt{\sum_{i=1}^{N_{\mathrm{lat}}} \sum_{j=1}^{N_{\mathrm{lon}}} L_i (\widetilde{X}^t_{i,j,c})^2}}. \tag{20}$$

Here N_{lat} and N_{lon} are the grid dimensions; the tilde denotes an anomaly relative to the same climatological field. For latitude ϕ_i,

$$L_i = \frac{N_{\mathrm{lat}} \cos \phi_i}{\sum_{i'=1}^{N_{\mathrm{lat}}} \cos \phi_{i'}}. \tag{21}$$

For regional forecasting, we follow STORM (Huang et al., 2026). Let $\widehat{x}_{\ell hw}$ and $y_{\ell hw}$ be the prediction and ground truth in physical units at forecast step ℓ and grid point (h, w), with L forecast steps and an $H \times W$ grid:

$$\mathrm{RMSE}_{\mathrm{STORM}} = \frac{1}{L} \sum_{\ell=1}^{L} \sqrt{\frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} \alpha(h) \big(y_{\ell hw} - \widehat{x}_{\ell hw}\big)^2}, \tag{22}$$

$$\mathrm{ACC}_{\mathrm{STORM}} = \frac{\sum_{\ell,h,w} \alpha(h) \widetilde{y}_{\ell hw} \widetilde{\widehat{x}}_{\ell hw}}{\sqrt{\sum_{\ell,h,w} \alpha(h) \widetilde{y}^2_{\ell hw}} \sqrt{\sum_{\ell,h,w} \alpha(h) \widetilde{\widehat{x}}^2_{\ell hw}}}. \tag{23}$$

The weight and anomalies are

$$\begin{aligned} \alpha(h) &= \frac{\cos \phi_h}{H^{-1} \sum_{h'=1}^{H} \cos \phi_{h'}}, \\ C &= \frac{1}{LHW} \sum_{\ell=1}^{L} \sum_{h=1}^{H} \sum_{w=1}^{W} y_{\ell hw}, \\ \widetilde{y}_{\ell hw} &= y_{\ell hw} - C, \\ \widetilde{\widehat{x}}_{\ell hw} &= \widehat{x}_{\ell hw} - C, \end{aligned} \tag{24}$$

where ϕ_h is the latitude of row h.

The published STORM table labels two numerical blocks as Continental South America at 1° (Cont-SA) and East Asia at 0.25° (Reg-EA). The STORM authors confirmed that these block labels were interchanged. The block containing the STORM RMSE sequence 0.874, 1.560, 2.866, 3.322, and 3.510 corresponds to Reg-EA; Table 2 uses the comparator rows from this author-confirmed East Asia block.

C Training Configuration

In this section, we describe how the global model and the regional CrossWarp model are trained. We first give the optimizer, learning-rate schedule, and loss used for the global model. We then describe which regional components are updated and how their learning rates and offset schedule are set.

C.1 Global Training

For the global model, we use FusedAdam with a batch size of 8 for 100 epochs. The learning rate starts at 2×10^{-4} and follows a cosine annealing schedule with a minimum of 3×10^{-7}.

In addition, the training objective combines channel-wise relative ℓ_2 error with MetNet3-style variable balancing. A cosine-latitude-weighted mean squared error term is added with coefficient 0.1, so the objective accounts for the unequal areas represented by latitude–longitude grid cells.

C.2 Regional Training

For regional training, we load the pretrained global provider and keep its parameters frozen. The regional backbone and CrossWarp are trained with a channel-wise ℓ_2 loss and MetNet3-style variable scaling. The CrossWarp training path uses Adam with separate parameter groups: the regional backbone and CrossWarp each start at a learning rate of 10^{-4}, while the offset parameters start at 10^{-5}. To introduce the learned offsets gradually, their effective scale is held at zero during the first two epochs and then increased linearly over the next six epochs. Training uses seed 42 and FP32 on the NVIDIA H100 GPUs, with one sample per GPU. Finally, an exponential moving average with decay 0.999 is maintained for the trainable CrossWarp parameters.

D Architecture and Algorithms

In this section, we provide the architectural and algorithmic details behind the forecasting pipeline introduced in the main text. We first follow the global state from grid embedding through mesh processing and back to the forecast grid. We then expand the directional propagation and grouped interaction within each processor stage, before describing how CrossWarp transfers frozen global features to the regional model. The two algorithms summarize these computations, and a final subsection records the principal tensor shapes.

D.1 Global Forecasting Architecture

The global forecaster uses a hidden feature width of 512 and 16 graph-processing layers: one encoder layer, 14 central processor stages, and one decoder layer. As shown in Figure 1, it follows an encoder–processor–decoder sequence. The grid embedding first represents the atmospheric variables at each latitude–longitude location. The encoder then transfers these features to a multilevel spherical mesh, where each processor stage combines local directional propagation with grouped long-range interaction. Finally, the decoder returns the processed representation to the grid for next-state prediction.

D.1.1 Grid Embedding

The global model operates on the preprocessed 121×240 latitude–longitude grid described in Appendix A.2. Starting from the atmospheric state X_t, grid embedding produces a feature vector for each of its $N_g = 29{,}040$ grid locations:

$$F_t = \mathrm{GridEmbed}(X_t). \tag{25}$$

D.1.2 Grid-to-Mesh Encoder

Next, the grid-to-mesh encoder connects each grid location to nearby vertices on the spherical mesh and transfers its embedded state along these edges. It produces the initial mesh-node and mesh-edge features:

$$(Z^0, E^0) = \mathrm{Enc}(F_t, \mathcal{E}_{g\to m}), \tag{26}$$

where $\mathcal{E}_{g\to m}$ denotes the grid-to-mesh edge set. Mesh vertices carry fixed geometric coordinates, while edge features describe the relative displacement in the receiver's local frame and the corresponding normalized distance.

D.1.3 MESH PROCESSOR

The processor operates on S nested icosahedral mesh levels whose edges retain the level at which they first appear. These levels share vertex indices, allowing messages from different spatial scales to meet at a common mesh node while remaining distinguishable during aggregation. Each connection is represented in both directions, and self-loops are omitted.

At each of the $L = 14$ central processor stages, indexed by $\ell = 0, \ldots, L-1$, the mesh-node and mesh-edge states are updated as

$$(Z^{\ell+1}, E^{\ell+1}) = \mathrm{ProcessorStage}_\ell(Z^\ell, E^\ell; \mathcal{E}). \tag{27}$$

The edge update comes first, followed by SD-DMP, GS-GLA, and a feed-forward residual. To supply the fixed graph coordinates used by SD-DMP, we form the normalized Laplacian $\mathcal{L} = I - D^{-1/2}AD^{-1/2}$ from the undirected adjacency A of the finest canonical mesh and its degree matrix D. The first K nontrivial eigenvectors provide spectral directions; differences between their source and receiver values define the directional weights in Appendix D.2.1.

D.1.4 MESH-TO-GRID DECODER AND PREDICTION HEAD

After mesh processing, the mesh-to-grid decoder transfers information from nearby mesh vertices back to each grid location. Its edges use the same receiver-centered geometric representation as the encoder. The decoded features and next atmospheric state are

$$F^{\mathrm{out}} = \mathrm{Dec}(Z^L, E^L; \mathcal{E}_{m\to g}), \tag{28}$$

$$\widehat{X}_{t+1} = \mathrm{Head}(F^{\mathrm{out}}), \tag{29}$$

where $\mathcal{E}_{m\to g}$ is the mesh-to-grid edge set. These encoder and decoder transitions surround the central processor stages; they do not replace the local and global updates within those stages.

D.2 GLOBAL PROCESSOR DETAILS AND PSEUDOCODE

With the global feature path established, we now detail the operations inside one processor stage. The local update uses spectral-coordinate differences to recover directional variation that a level-wise mean cannot represent. Its output then enters the grouped long-range interaction described in the main text. We define the local weights below and collect the full stage sequence in Algorithm 1.

D.2.1 DIRECTIONAL PROPAGATION DETAILS

For the kth nontrivial eigenvector ϕ_k of $\mathcal{L}$, the difference $\phi_k(j) - \phi_k(i)$ assigns an orientation to an edge $j \to i$. Within mesh level s, its magnitude and signed value give the normalized weights

$$\alpha_{ji}^{k,s} = \frac{|\phi_k(j) - \phi_k(i)| + \epsilon}{\sum_{u:(u,i)\in\mathcal{E}_s}(|\phi_k(u) - \phi_k(i)| + \epsilon)}, \tag{30}$$

$$\beta_{ji}^{k,s} = \frac{\phi_k(j) - \phi_k(i)}{\max\Big(\sum_{u:(u,i)\in\mathcal{E}_s} |\phi_k(u) - \phi_k(i)|, \epsilon\Big)}. \tag{31}$$

The unsigned weights compare incoming updated edge features with the level-fused mean m_i from Eq. 7, while the signed weights measure variation in the current node states. Writing e'_{ji} for the updated edge feature in the current processor stage gives

$$c_i^k = \sum_s \bar{\pi}_{is} \sum_{j:(j,i)\in\mathcal{E}_s} \alpha_{ji}^{k,s} e'_{ji} - m_i, \tag{32}$$

$$r_i^k = \left| \sum_s \bar{\pi}_{is} \sum_{j:(j,i)\in\mathcal{E}_s} \beta_{ji}^{k,s} (z_j^\ell - z_i^\ell) \right|. \tag{33}$$

Because the sign of an eigenvector is arbitrary, taking the absolute value makes r_i^k invariant to that choice. We then normalize the two signals separately and fuse them through a bottleneck of width B_d:

$$\delta_i = W_o \, \mathrm{GELU}\left(\sum_{k=1}^{K} \left[P_k^c \, \mathrm{LN}(c_i^k) + P_k^r \, \mathrm{LN}(r_i^k) \right] \right), \tag{34}$$

Here D_e and D_v are the edge and node feature widths, respectively, with $P_k^c \in \mathbb{R}^{B_d \times D_e}$, $P_k^r \in \mathbb{R}^{B_d \times D_v}$, and $W_o \in \mathbb{R}^{D_e \times B_d}$. Adding δ_i to the level-fused message gives the node update in Eq. 9. Thus, the directional correction changes with the evolving edge and node features even though the spectral coordinates remain fixed.

D.2.2 Global Processor

Algorithm 1 follows the same order as the global processor in the main text: update edges, combine level-specific messages, apply the directional correction, read from geography- and state-aware groups, and finish with the feed-forward residual. The source assignments and target routes are computed independently, so each target can select among summaries built from the full mesh without forming pairwise node attention.

Algorithm 1 One AtmosWeave global processor stage

Require: Node states Z^ℓ, edge states E^ℓ, directed multimesh $\mathcal{E}$, levels $s(e)$, coordinates g_i, and precomputed $\alpha_{ji}^{k,s}, \beta_{ji}^{k,s}$
Ensure: Updated states $Z^{\ell+1}, E^{\ell+1}$
for all $(j \to i) \in \mathcal{E}$ **do**
  $e_{ji}^{\ell+1} \leftarrow e_{ji}^\ell + f_e([e_{ji}^\ell, Z_j^\ell, Z_i^\ell])$
end for
for $s = 1, \ldots, S$ **do**
  $d_i^s \leftarrow |\{j : (j,i) \in \mathcal{E}_s\}|$
  $\mu_i^s \leftarrow (d_i^s)^{-1} \sum_{j:(j,i)\in\mathcal{E}_s} e_{ji}^{\ell+1}$ for active levels
end for
$\pi \leftarrow \mathrm{softmax}(\eta^\ell)$
Renormalize π_s over active levels to obtain $\bar{\pi}_{is}$
$m_i \leftarrow \sum_s \bar{\pi}_{is} m_i^s$
$d_i \leftarrow 0$
for $k = 1, \ldots, K$ **do**
  $c_i^k \leftarrow \sum_s \bar{\pi}_{is} \sum_{j:(j,i)\in\mathcal{E}_s} \alpha_{ji}^{k,s} e_{ji}^{\ell+1} - m_i$
  $r_i^k \leftarrow \left| \sum_s \bar{\pi}_{is} \sum_{j:(j,i)\in\mathcal{E}_s} \beta_{ji}^{k,s} (Z_j^\ell - Z_i^\ell) \right|$
  $d_i \leftarrow d_i + P_k^c \, \mathrm{LN}(c_i^k) + P_k^r \, \mathrm{LN}(r_i^k)$
end for
$\delta_i \leftarrow W_o \, \mathrm{GELU}(d_i)$
$U_i \leftarrow Z_i^\ell + f_v([m_i + \delta_i; Z_i^\ell])$
$\widetilde{U}_i \leftarrow \mathrm{LN}(U_i)$ and compute p_i, q_i^h, k_i^h, v_i^h for each head h using Eq. 10
Compute source assignments and target routes using Eqs. equation 11 and equation 12, respectively
Accumulate S_g^h and z_g^h for all groups and heads using Eq. 13
for all nodes i **do**
  $o_{ig}^h \leftarrow (q_i^h)^\top S_g^h / \max((q_i^h)^\top z_g^h, \epsilon)$
  $\bar{o}_i \leftarrow \sum_g r_{ig} \, \mathrm{Concat}_h(o_{ig}^h)$
  $\Delta_i^\ell \leftarrow W_O^\ell(\mathrm{LN}(\bar{o}_i) \odot (p_i + b^\ell))$
  $V_i \leftarrow U_i + \Delta_i^\ell$
  $Z_i^{\ell+1} \leftarrow V_i + \mathrm{MLP}(\mathrm{LN}(V_i))$
end for
return $Z^{\ell+1}, E^{\ell+1}$

D.3 CrossWarp Architecture

We next describe the regional CrossWarp architecture. As summarized in Section 4.4, a frozen global provider supplies context at three points along the regional feature path: grid features before mesh encoding, mesh features during processing, and grid features after decoding. All three couplings begin

from geographic alignment, but they transfer different components of the routed global information. In particular, encoder and decoder grid correspondences are computed independently; shared notation below refers to a new correspondence at each stage.

D.3.1 GRID CORRESPONDENCE AND CONFIDENCE

At either grid coupling, let $G \in \mathbb{R}^{B\times C_g\times H_g\times W_g}$ and $R \in \mathbb{R}^{B\times C_r\times H_r\times W_r}$ denote the global and regional feature fields, where B is batch size, C_g, C_r are channel widths, and $H_g \times W_g, H_r \times W_r$ are their grid sizes. For regional location p, let $b_p \in [-1, 1]^2$ be its coordinate-aligned anchor on the global grid, and let Sample denote grid sampling. With learned feature projections ψ_r and ψ_g, the routing context combines the receiving regional feature with global context at this anchor:

$$c_p = [\psi_r(R)_p,\ \mathrm{Sample}(\psi_g(G), b_p)]. \tag{35}$$

Around this anchor, CrossWarp considers a bounded set of sampling positions that includes the aligned position and nearby directional alternatives. Within this subsection, h indexes these positions and H_w denotes their number; this index is distinct from the attention-head index used for GS-GLA. A fixed direction $\mathbf{d}_h \in \mathbb{R}^2$ and the bounded output $\widetilde{\Delta}_{ph} = [\tanh f_\Delta(c_p)]_h$ of a learned offset predictor f_Δ define each offset:

$$\Delta_{ph} = s_{\mathrm{off}}(e)\rho_h \left(\lambda_{\mathrm{base}}\mathbf{d}_h + \lambda_{\mathrm{res}} m_h \widetilde{\Delta}_{ph}\right), \tag{36}$$

Here e denotes the training epoch, $s_{\mathrm{off}}(e) \in [0, 1]$ gradually activates the offsets, ρ_h bounds their radius, and m_h keeps the aligned position fixed. The coefficients λ_{base} and λ_{res} balance the fixed directional component and learned residual, after which the offset is projected into the radius-ρ_h disk. Let K_{grid} and V_{grid} be key and value maps projected from G. With $\kappa = (2/W_g, 2/H_g)$, grid sampling gives

$$\begin{aligned} K_{ph} &= \mathrm{Sample}(K_{\mathrm{grid}}, b_p + \kappa \odot \Delta_{ph}), \\ V_{ph} &= \mathrm{Sample}(V_{\mathrm{grid}}, b_p + \kappa \odot \Delta_{ph}). \end{aligned} \tag{37}$$

The regional feature supplies a query q_p. Content similarity between this query and each sampled key, together with a learned prior, gives the routing weights u_{ph}. Hats on queries and keys denote normalized feature vectors. The route entropy then determines a confidence χ_p:

$$\begin{aligned} u_{ph} &= \mathrm{softmax}_h\left(\frac{\langle \widehat{q}_p, \widehat{K}_{ph}\rangle}{\tau} + f_u(c_p)_h\right), \\ \chi_p &= \epsilon_c + (1-\epsilon_c)\left(1 - \frac{\mathcal{H}(u_p)}{\log H_w}\right). \end{aligned} \tag{38}$$

Here τ is the routing temperature, ϵ_c is the confidence floor, and $\mathcal{H}(u_p)$ is the entropy of the routing distribution. The aligned value is $A_p = \mathrm{Sample}(V_{\mathrm{grid}}, b_p)$, whereas $S_p = \sum_h u_{ph} V_{ph}$ combines the sampled values. At the encoder, Eq. 16 uses confidence to interpolate between these values before a gated residual update to the regional grid feature. Thus, alignment remains the reference while the regional state controls the additional global information.

D.3.2 SPARSE MESH CORRESPONDENCE

After the regional features have entered the mesh processor, CrossWarp uses sparse global-to-regional mesh correspondence at selected stages $\ell \in \mathcal{D}_{\mathrm{proc}}$. Each regional node i reads from a geographically initialized set $\mathcal{C}(i)$ of global mesh nodes. Let q_i^ℓ be the regional query and k_j^ℓ, v_j^ℓ the key and value for global node j. At a later coupling stage ℓ, the routing distribution $\pi_i^{\ell^-}$ from the preceding coupling stage ℓ^- contributes an entropy-adaptive persistence term:

$$\lambda_i^\ell = \lambda_0\left(1 - \frac{\mathcal{H}(\pi_i^{\ell^-})}{\log|\mathcal{C}(i)|}\right), \tag{39}$$

$$\pi_{ij}^\ell = \mathrm{softmax}_{j\in\mathcal{C}(i)}\left(\frac{\langle \widehat{q}_i^\ell, \widehat{k}_j^\ell\rangle}{\tau} + \lambda_i^\ell \log \pi_{ij}^{\ell^-}\right). \tag{40}$$

Here λ_0 controls persistence, which is omitted at the first coupling stage. The aligned global value a_i^ℓ and routed value $s_i^\ell = \sum_{j \in \mathcal{C}(i)} \pi_{ij}^\ell v_j^\ell$ are blended using confidence $\chi_i^\ell = 1 - \mathcal{H}(\pi_i^\ell)/\log|\mathcal{C}(i)|$ to give $t_i^\ell = a_i^\ell + \chi_i^\ell(s_i^\ell - a_i^\ell)$. Unlike encoder grid transport, the processor injects only the component of this global feature that is orthogonal to the receiving regional feature within each channel group. Let $g_{i,r}^\ell$ be the projected and normalized transported global feature, $v_{i,r}^\ell$ the regional reference feature, and R_i^ℓ the regional mesh state:

$$\widehat{v}_{i,r}^\ell = \mathrm{stopgrad}\big(v_{i,r}^\ell / \max(\|v_{i,r}^\ell\|_2, \epsilon)\big)\,, \tag{41}$$

$$I_{i,r}^\ell = g_{i,r}^\ell - \langle g_{i,r}^\ell, \widehat{v}_{i,r}^\ell \rangle \widehat{v}_{i,r}^\ell, \tag{42}$$

$$R_i^{\ell,+} = R_i^\ell + \mathrm{Concat}_r[\alpha_r^\ell \gamma_{i,r}^\ell I_{i,r}^\ell]. \tag{43}$$

Here α_r^ℓ is a residual scale and $\gamma_{i,r}^\ell$ is its gate. The stopped-gradient normalization defines the direction removed from the global feature; this feature-space operation does not imply statistical independence between global and regional forecasts.

D.3.3 Decoder Residual

Finally, after the global and regional mesh-to-grid decoders, CrossWarp computes a new grid correspondence from their decoded features. The decoder retains only the confidence-weighted difference between its routed and aligned global values:

$$\begin{aligned} T_p &= \chi_p(S_p - A_p), \\ \widetilde{R}'_p &= \widetilde{R}_p + \widetilde{\alpha} \odot \widetilde{\gamma}_p \odot P_o(T_p). \end{aligned} \tag{44}$$

Here $\widetilde{R}_p$ is the decoded regional grid feature, P_o projects the transported difference, and $\widetilde{\alpha}$ and $\widetilde{\gamma}_p$ scale and gate the residual update. The regional prediction head then maps $\widetilde{R}'_p$ to forecast variables. This differs from encoder transport, which supplies a full global value, and from processor transport, which retains an orthogonal feature component.

D.3.4 CrossWarp Forward Algorithm

Algorithm 2 follows the regional feature stream through these three couplings. It keeps the global backbone frozen, performs encoder grid transport before regional mesh processing, inserts sparse mesh transport at selected processor stages, and computes an independent decoder correction before prediction.

Algorithm 2 CrossWarp adaptive global-to-regional coupling

Require: Frozen global encoder, processor, and decoder features; regional input X^r; aligned coordinates b_p; grid directions $\mathbf{d}_h$; mesh candidate sets $\mathcal{C}(i)$; processor coupling depths $\mathcal{D}_{\text{proc}}$

Ensure: Regional forecast $\widehat{Y}^r$

Encode X^r as regional grid features R; keep the global features fixed
Project the global grid features to key and value maps K_{grid} and V_{grid}
for all regional grid locations p **do**
    Form $c_p = [\psi_r(R)_p, \text{Sample}(\psi_g(G), b_p)]$
    Predict bounded offsets Δ_{ph} with Eq. 36; keep the aligned candidate fixed
    Sample global keys and values at $b_p + \kappa \odot \Delta_{ph}$
    Compute candidate routes u_{ph} and confidence χ_p using Eq. 38
    $A_p \leftarrow \text{Sample}(V_{\text{grid}}, b_p)$
    $S_p \leftarrow \sum_h u_{ph} V_{ph}$
    $T_p \leftarrow A_p + \chi_p(S_p - A_p)$
    $R_p \leftarrow R_p + \alpha \odot \gamma_p \odot P_o(T_p)$
end for
Map R to regional mesh states and initialize the previous coupling route as empty
for $\ell = 1, \ldots, L_r$ **do**
    Apply regional processor stage ℓ
    if $\ell \in \mathcal{D}_{\text{proc}}$ **then**
        Build regional queries and read global keys and values on $\mathcal{C}(i)$
        Compute π_{ij}^{ℓ} with Eq. 40; omit persistence at the first coupling depth
        $a_i^{\ell} \leftarrow$ aligned value
        $s_i^{\ell} \leftarrow \sum_{j \in \mathcal{C}(i)} \pi_{ij}^{\ell} v_j^{\ell}$
        $\chi_i^{\ell} \leftarrow 1 - \mathcal{H}(\pi_i^{\ell}) / \log |\mathcal{C}(i)|$
        $t_i^{\ell} \leftarrow a_i^{\ell} + \chi_i^{\ell}(s_i^{\ell} - a_i^{\ell})$
        Split $\text{LN}(P_o t_i^{\ell})$ and the normalized regional state into channel groups $g_{i,r}^{\ell}, v_{i,r}^{\ell}$
        $\widehat{v}_{i,r}^{\ell} \leftarrow \text{stopgrad}(v_{i,r}^{\ell} / \max(\|v_{i,r}^{\ell}\|_2, \epsilon))$
        $I_{i,r}^{\ell} \leftarrow g_{i,r}^{\ell} - \langle g_{i,r}^{\ell}, \widehat{v}_{i,r}^{\ell} \rangle \widehat{v}_{i,r}^{\ell}$
        $R_i \leftarrow R_i + \text{Concat}_r[\alpha_r^{\ell} \gamma_{i,r}^{\ell} I_{i,r}^{\ell}]$
        Store π_i^{ℓ} for the next coupling depth
    end if
end for
Decode the regional mesh and project global decoder features to new key and value maps
Independently recompute decoder offsets, routes, A_p, S_p, and χ_p
$T_p \leftarrow \chi_p(S_p - A_p)$
$\widetilde{R}_p \leftarrow \widetilde{R}_p + \widetilde{\alpha} \odot \widetilde{\gamma}_p \odot P_o(T_p)$
Predict $\widehat{Y}^r$ from $\widetilde{R}$
return $\widehat{Y}^r$

D.4 Tensor Shapes

To close the architectural description, Table 5 records the principal feature shapes used by the global processor and regional couplings. Let B, N, M, D_v, D_e denote batch size, mesh-node count, edge count, node width, and edge width. Within the global processor, S, K, G denote mesh levels, spectral directions, and learned groups, and d_h is the width of an attention head; the CrossWarp equations separately use G for a global grid feature field. For CrossWarp, H_w is the number of grid sampling positions, D_w their value width, N_r the number of regional mesh nodes per graph, C_m the size of a sparse mesh candidate set, and D_r its value width.

Table 5: Principal tensor shapes. LN and learned maps act on the final feature dimension.

Stage	Shape	Operation
Edge update	$[B, M, D_e]$	residual edge MLP
Level means	$[B, N, S, D_e]$ $[B, N, K, D_e]$	scatter by destination and mesh level
Directional fields	$[B, N, K, D_v]$	smoothing contrast and centered derivative
Q, K, V	$[B, N, D_v]$	independent projections partitioned into heads; sigmoid on Q, K
Routes	$[B, N, G]$ each	source assignment and target routing
Statistics	$[B, G, d_h, d_h]$ per head	grouped key–value summaries
Grid samples	$[B, H_w, D_w, H_r, W_r]$	aligned and bounded global values at regional grid locations
Mesh samples (per graph)	$[N_r, C_m, D_r]$	sparse global values read by regional mesh nodes

Table 6: Comparative analysis of parameters and MACs for deep learning weather forecasting models, with hardware and training time. Asterisks indicate training times that exclude finetuning.

Model	Params (M)	MACs (G)	Hardware	Training time
FengWu (Chen et al., 2023a)	153.49	132.83	32 A100	17 days
FourCastNet (Kurth et al., 2023)	79.60	111.62	64 A100	16 hours
GraphCast (Lam et al., 2023)	28.95	1639.26	32 TPUv4	4 weeks
Pangu-Weather (Bi et al., 2023)	23.83	142.39	192 V100	64 days
VA-MoE (Chen et al., 2025)	665.37	–	16 A100	12 days*
OneForecast (Gao et al., 2025)	24.76	509.27	16 A100	8 days*
EMFormer (Chen et al., 2026)	157.90	102.54	16 A100	60 hours*
ATMOSWEAVE	43.27	466.88	8 H100	5 days

Table 7: Regional subsystem cost. The CrossWarp + Regional Model and Regional Model columns exclude the frozen global provider; measurements across external models are not hardware-normalized.

Metric	CrossWarp + Regional Model	Regional Model	STORM	Pangu-Weather	FCN	FuXi
Parameters (M) ↓	4.63	3.14	15.77	97.48	64.65	661.01
FLOPs (G) ↓	42.81	25.83	84.59	91.26	58.50	302.46
Inference time (s/100 samples) ↓	6.66	5.87	0.87	5.11	6.99	19.37
Peak GPU memory (MB; batch 1) ↓	738.29	489.26	729.22	408.75	265.95	2533.25

E EFFICIENCY ANALYSIS

In this section, we examine the cost of global and regional forecasting in terms of model size, computation, and measured resource use. For global forecasting, Table 6 reports parameter counts, multiply–accumulate operations (MACs), training hardware, and training time. The comparator values follow Table 15 of EMFormer (Chen et al., 2026), whereas the ATMOSWEAVE values are measured in this work.

As shown in Table 6, ATMOSWEAVE has 43.27 million parameters and requires 466.88 billion MACs under the reported measurement setting. Its global training run used eight H100 GPUs and took five days. Because the models were trained with different hardware resources, the reported durations should be read alongside their hardware configurations rather than treated as directly comparable training speeds.

For regional forecasting, Table 7 compares the regional model alone with the same model equipped with CROSSWARP, alongside reported costs for other methods. The two ATMOSWEAVE columns measure only the regional subsystem and exclude the frozen global provider. This separation shows the additional cost of adaptive global-to-regional coupling.

Adding CROSSWARP increases the regional subsystem from 3.14 to 4.63 million parameters and from 25.83 to 42.81 GFLOPs. In the reported measurements, inference time rises from 5.87 to 6.66 seconds per 100 samples, and peak GPU memory rises from 489.26 to 738.29 MB at batch size one. Inference time and memory values for the external models are not hardware-normalized, so the table primarily characterizes the cost within each reported setup.

Table 8: Finetuned global forecasting results. Lower normalized RMSE (↓) and higher ACC (↑) indicate better performance.

	6 h		1 day		4 days		7 days		10 days	
Model	RMSE↓	ACC↑	RMSE↓	ACC↑	RMSE↓	ACC↑	RMSE↓	ACC↑	RMSE↓	ACC↑
EMFormer (w/ finetuning) (Chen et al., 2026)	0.0599	0.9949	0.1139	0.9775	0.2539	0.8936	0.4072	0.7223	0.5094	0.5389
AtmosWeave (w/ finetuning)	0.0573	0.9949	0.1136	0.9774	0.2361	0.9099	0.3792	0.7548	0.4968	0.5386

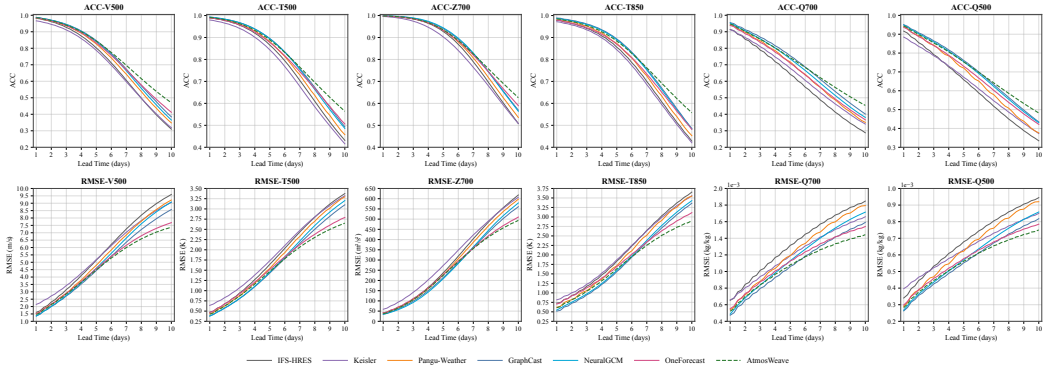

Figure 6: ACC (top row) and RMSE (bottom row) comparisons for V500, T500, Z700, T850, Q700, and Q500 across forecast lead times. Curves show IFS-HRES, Keisler, Pangu-Weather, GraphCast, NeuralGCM, OneForecast, and AtmosWeave; baseline results are from WeatherBench 2 (Rasp et al., 2024).

F Additional Results

In this section, we provide supporting results for the global and regional forecasting experiments. We first describe the component ablations, then compare finetuned global forecasts and extend the variable-wise evaluation. The final subsection presents forecast visualizations at several lead times.

F.1 Ablation Analysis

Table 3 reports four-day component comparisons using the metrics defined in Appendix B. All global variants use the same 1999–2020 ERA5 subset, data split, initial conditions, and evaluation setting. The baseline omits both proposed global operators. The remaining variants remove either SD-DMP or GS-GLA, replace grouped attention with ungrouped linear attention, or retain both operators in the complete AtmosWeave. These comparisons isolate the contribution of each operator and the grouping mechanism.

For regional forecasting, the ablation compares a model without global coupling, one that transfers features from geographically aligned global locations, and the complete CrossWarp with adaptive correspondence. The three variants share the same experimental setting, allowing the effects of global context and learned correspondence to be examined separately.

F.2 Finetuning Results

Next, Table 8 compares finetuned global forecasting results at the five lead times used in the main evaluation. This comparison complements the results reported for the original models in Section 5.

After finetuning, AtmosWeave has lower normalized RMSE than EMFormer at every reported lead time. Its ACC matches EMFormer at 6 hours, is higher at 4 and 7 days, and is slightly lower at 1 and 10 days.

F.3 Additional Variable-wise Comparison

Finally, Figure 6 extends the ACC and RMSE comparison in Figure 3 to six additional atmospheric variables. The lead-time curves provide a more detailed view of performance across variables beyond those highlighted in the main text.

F.4 FORECAST VISUALIZATIONS

The following figures show global weather forecasts from 6 hours to 10 days in chronological order. After the 1-day forecast, the sequence includes both full-day and intervening half-day lead times.

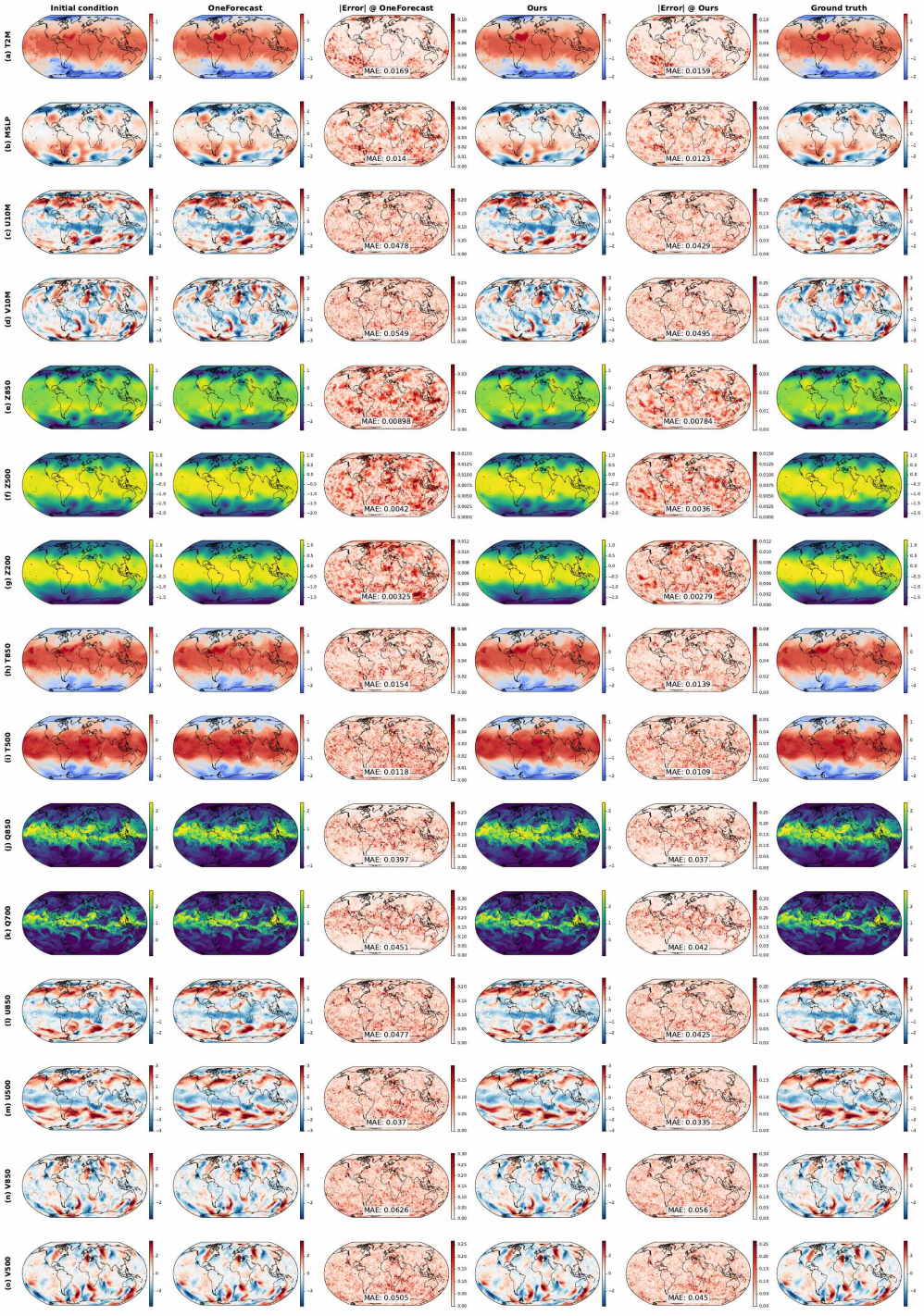

Figure 7: Visualization of global weather forecasts at a lead time of 6 hours for different atmospheric variables.

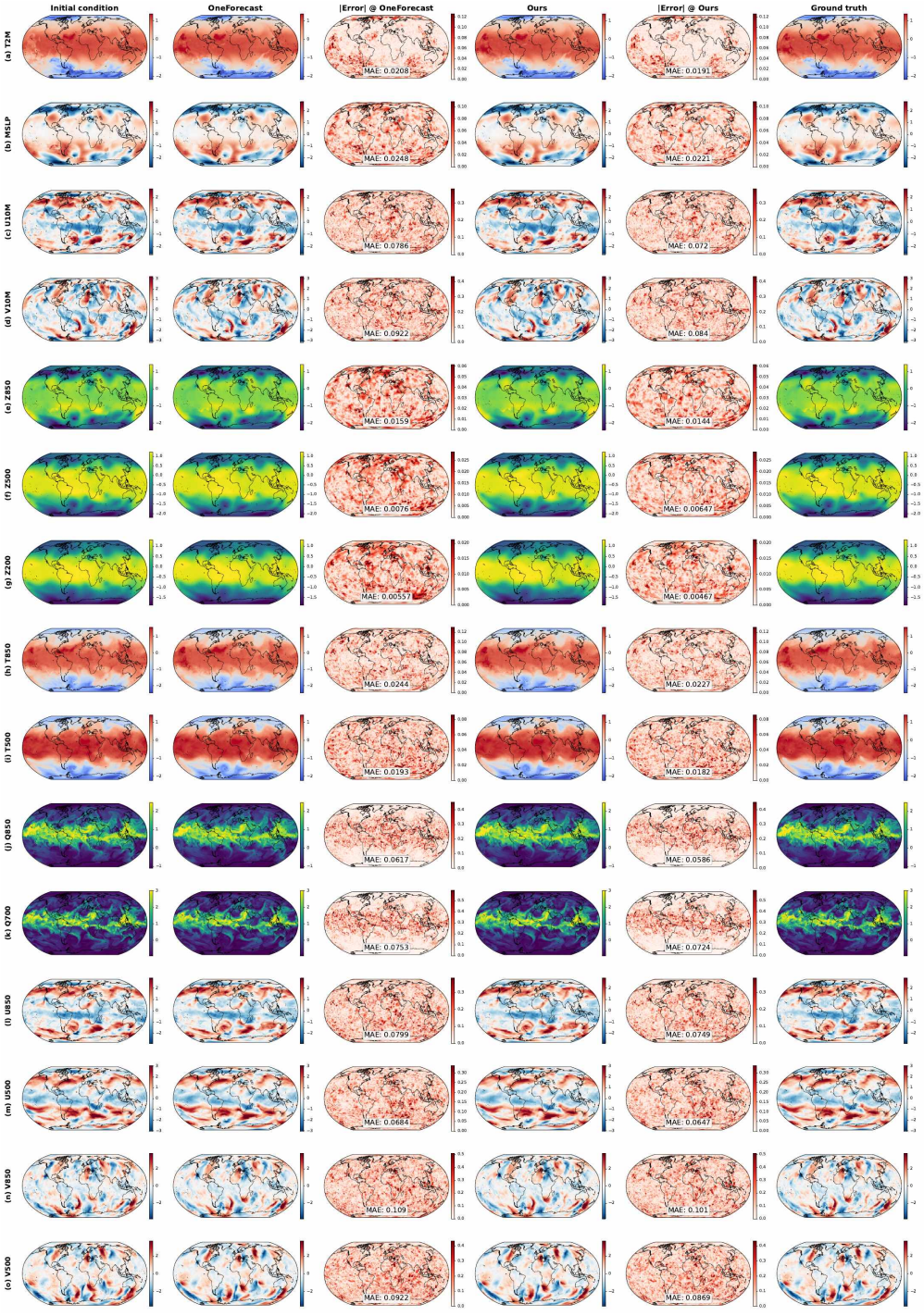

Figure 8: Visualization of global weather forecasts at a lead time of 12 hours for different atmospheric variables.

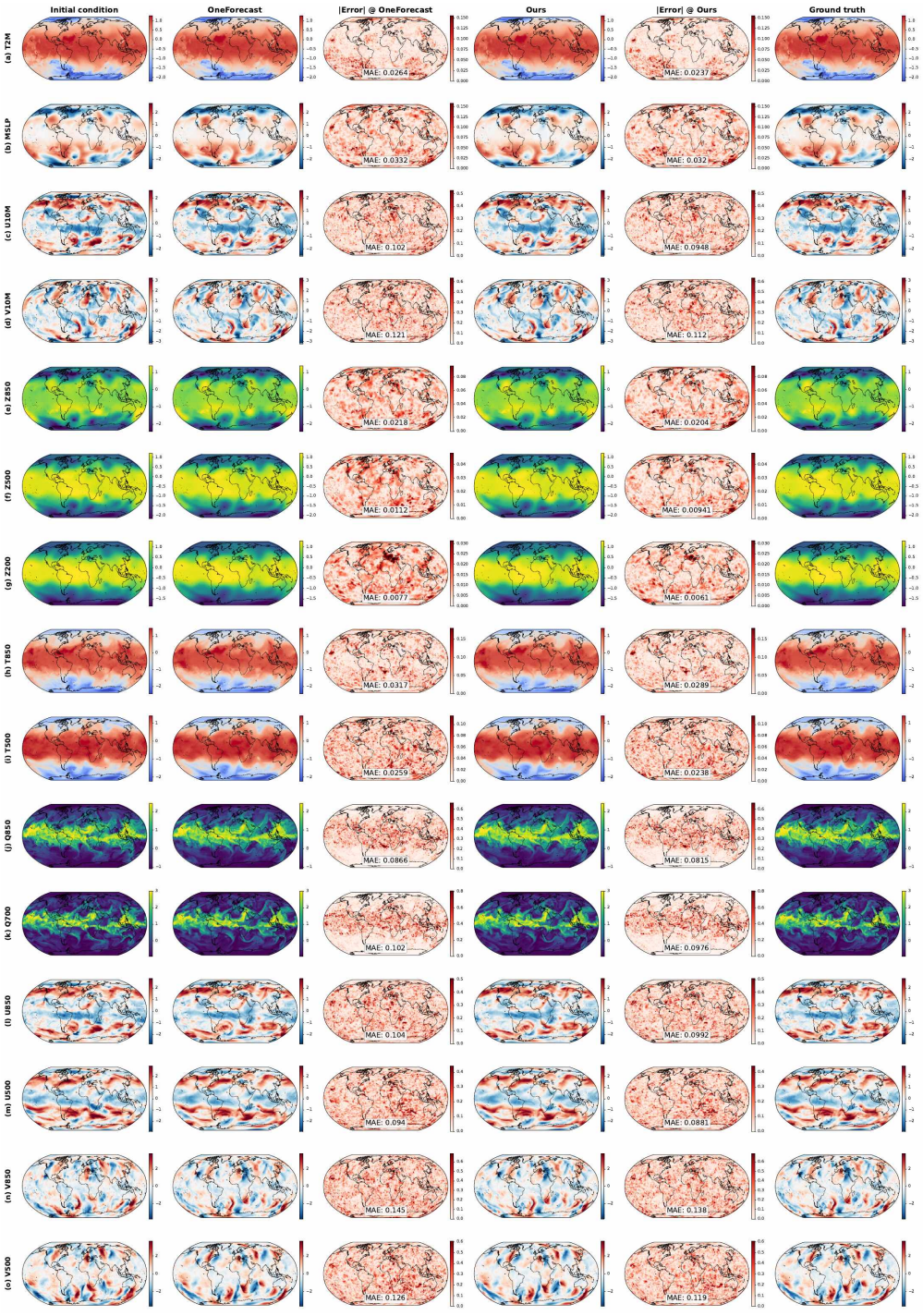

Figure 9: Visualization of global weather forecasts at a lead time of 1 day for different atmospheric variables.

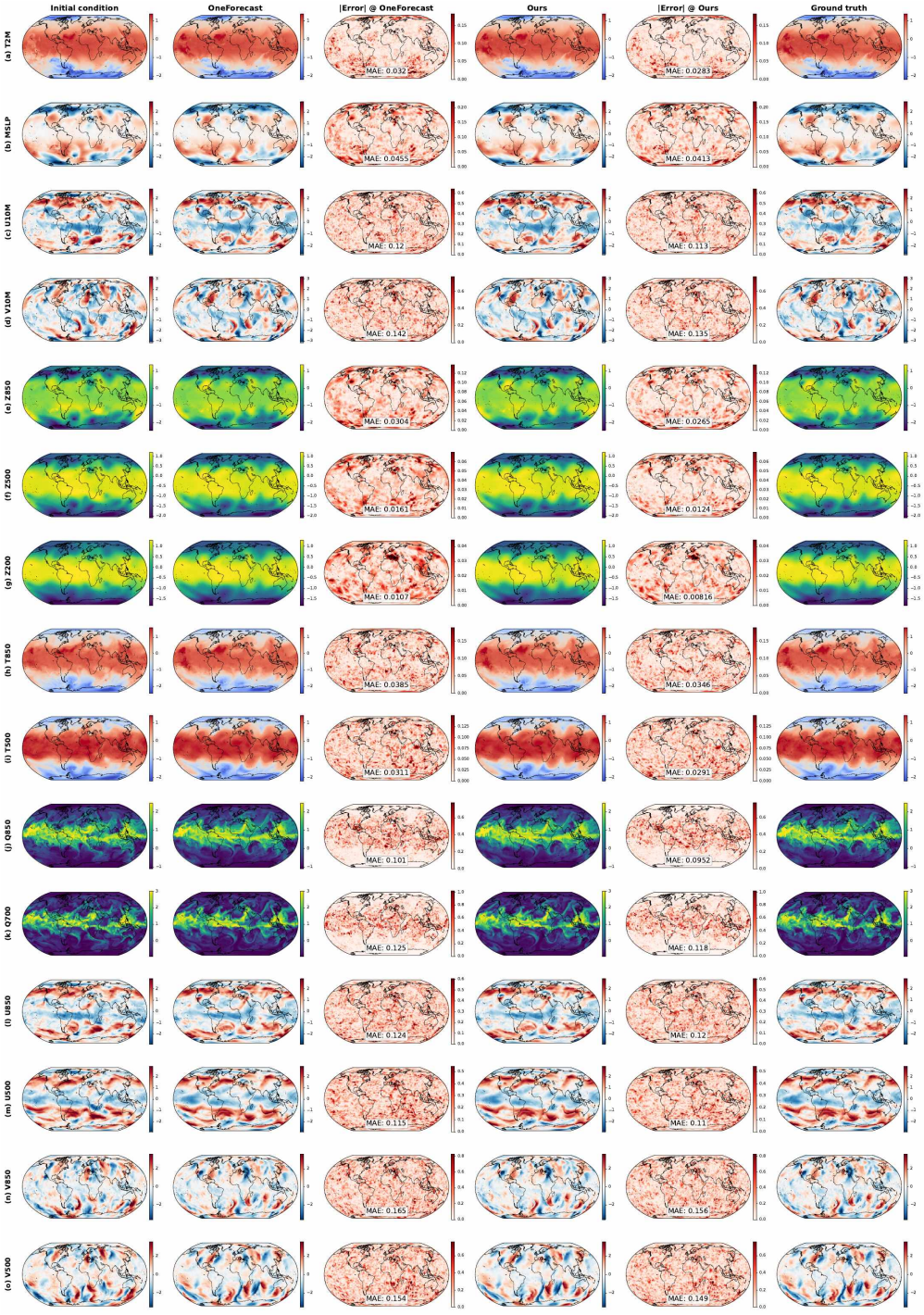

Figure 10: Visualization of global weather forecasts at a lead time of 1.5 days for different atmospheric variables.

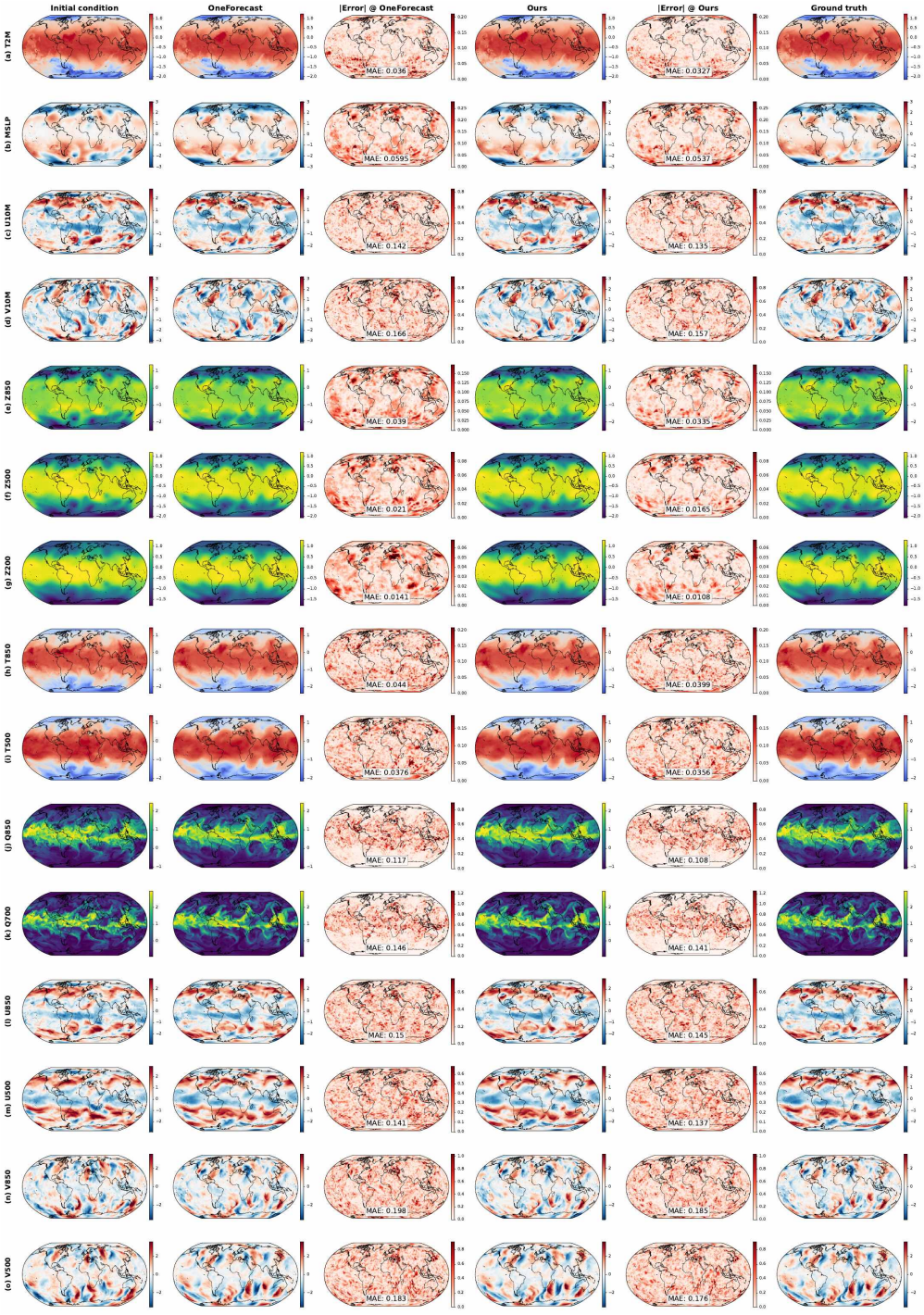

Figure 11: Visualization of global weather forecasts at a lead time of 2 days for different atmospheric variables.

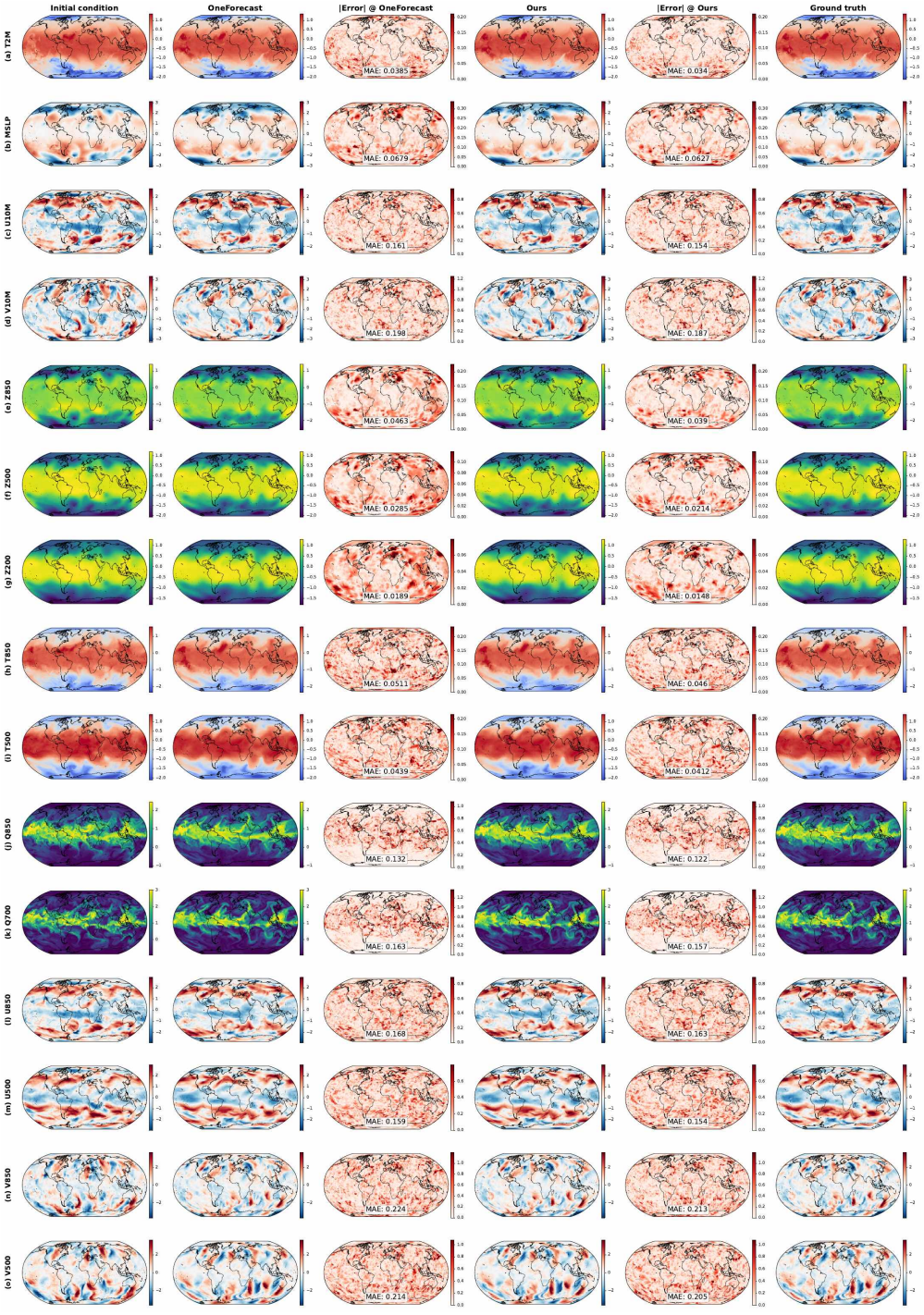

Figure 12: Visualization of global weather forecasts at a lead time of 2.5 days for different atmospheric variables.

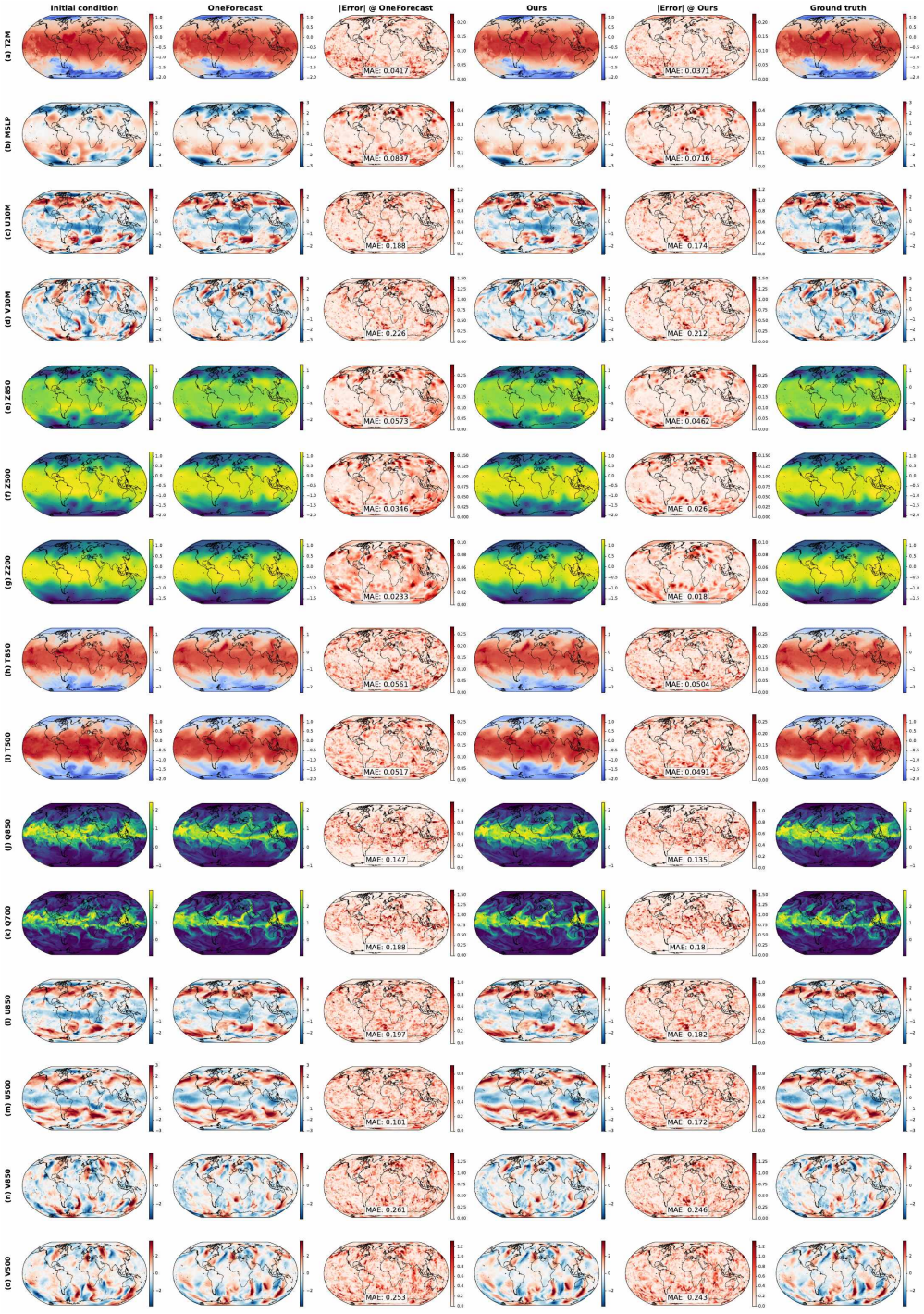

Figure 13: Visualization of global weather forecasts at a lead time of 3 days for different atmospheric variables.

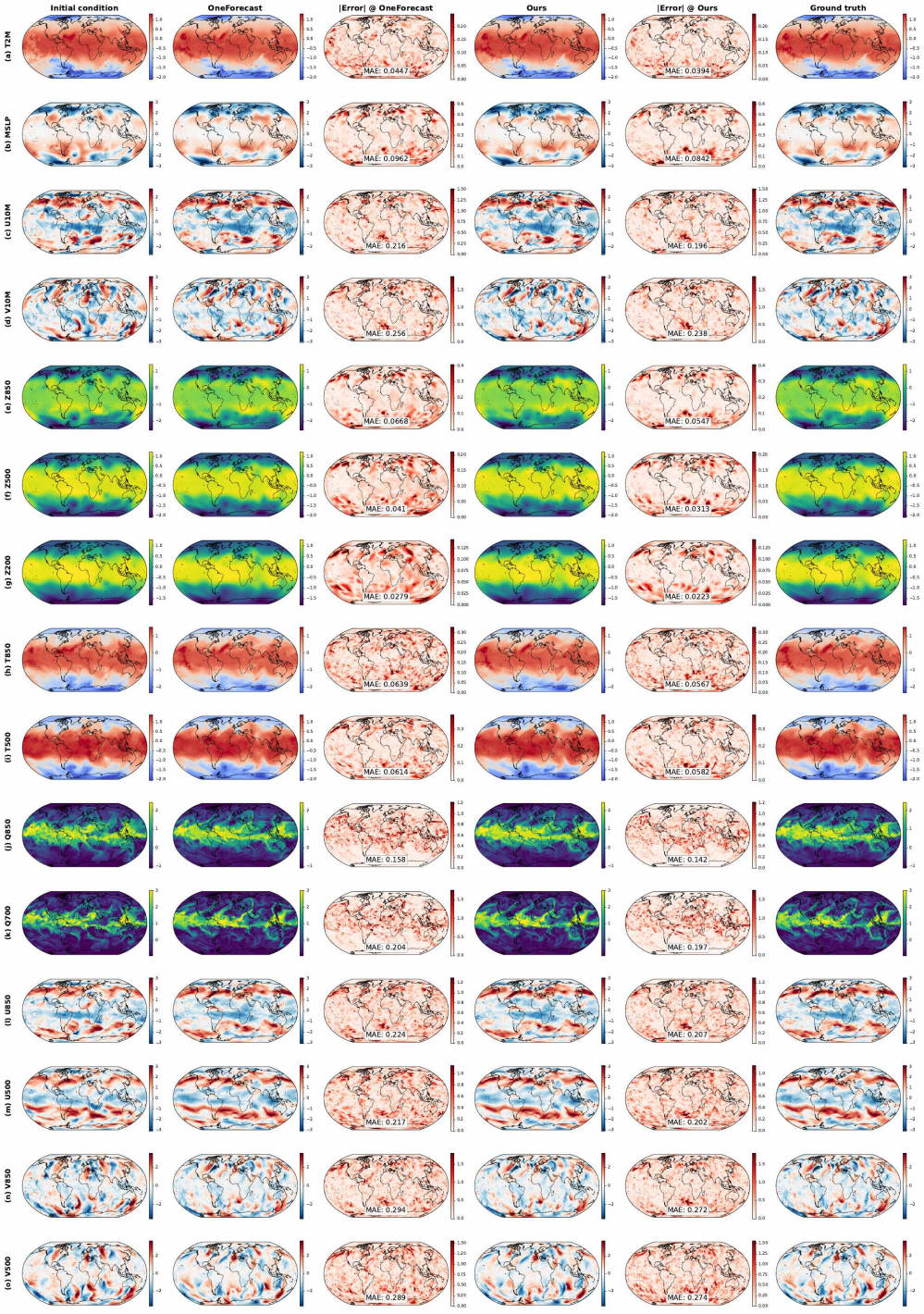

Figure 14: Visualization of global weather forecasts at a lead time of 3.5 days for different atmospheric variables.

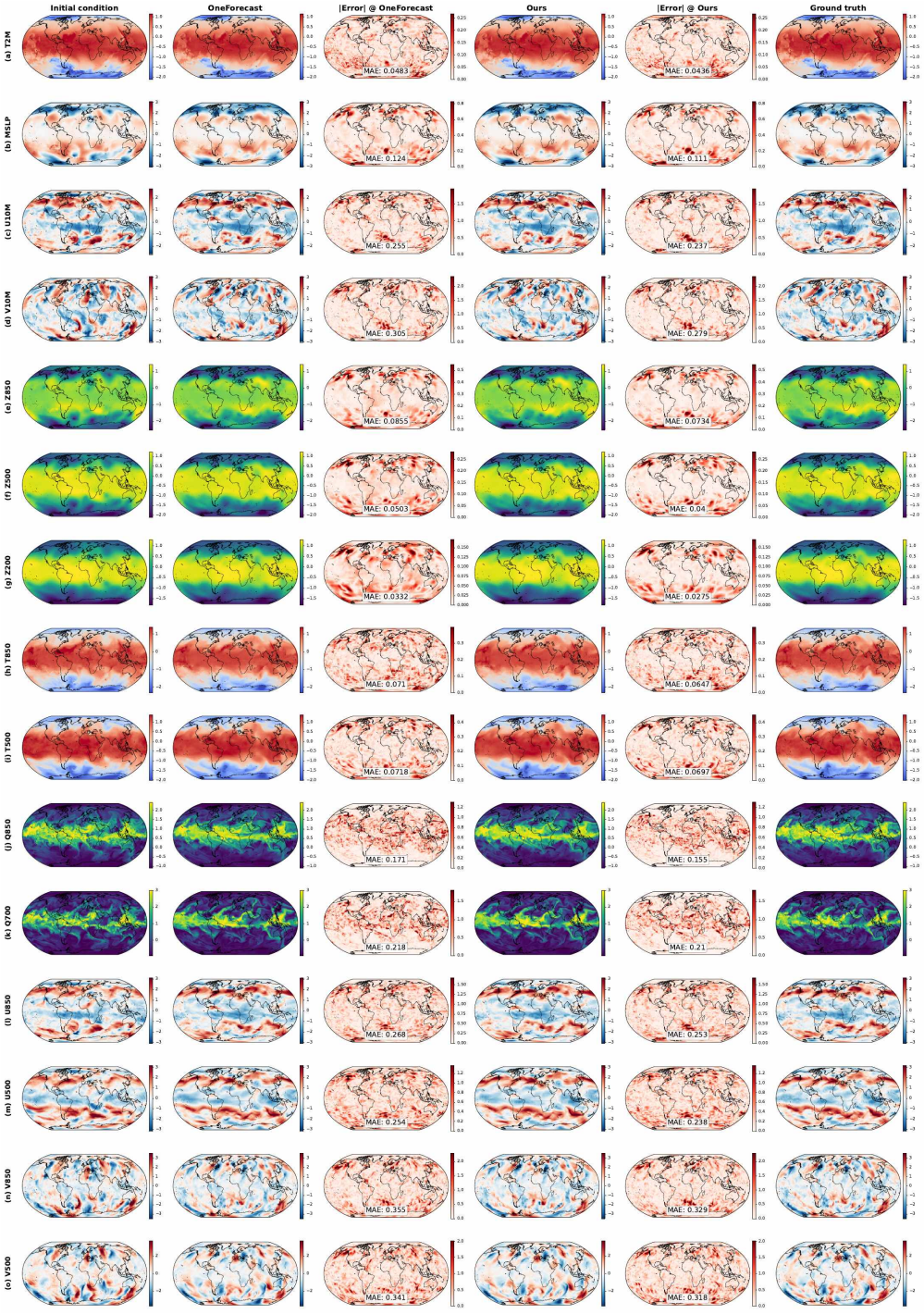

Figure 15: Visualization of global weather forecasts at a lead time of 4 days for different atmospheric variables.

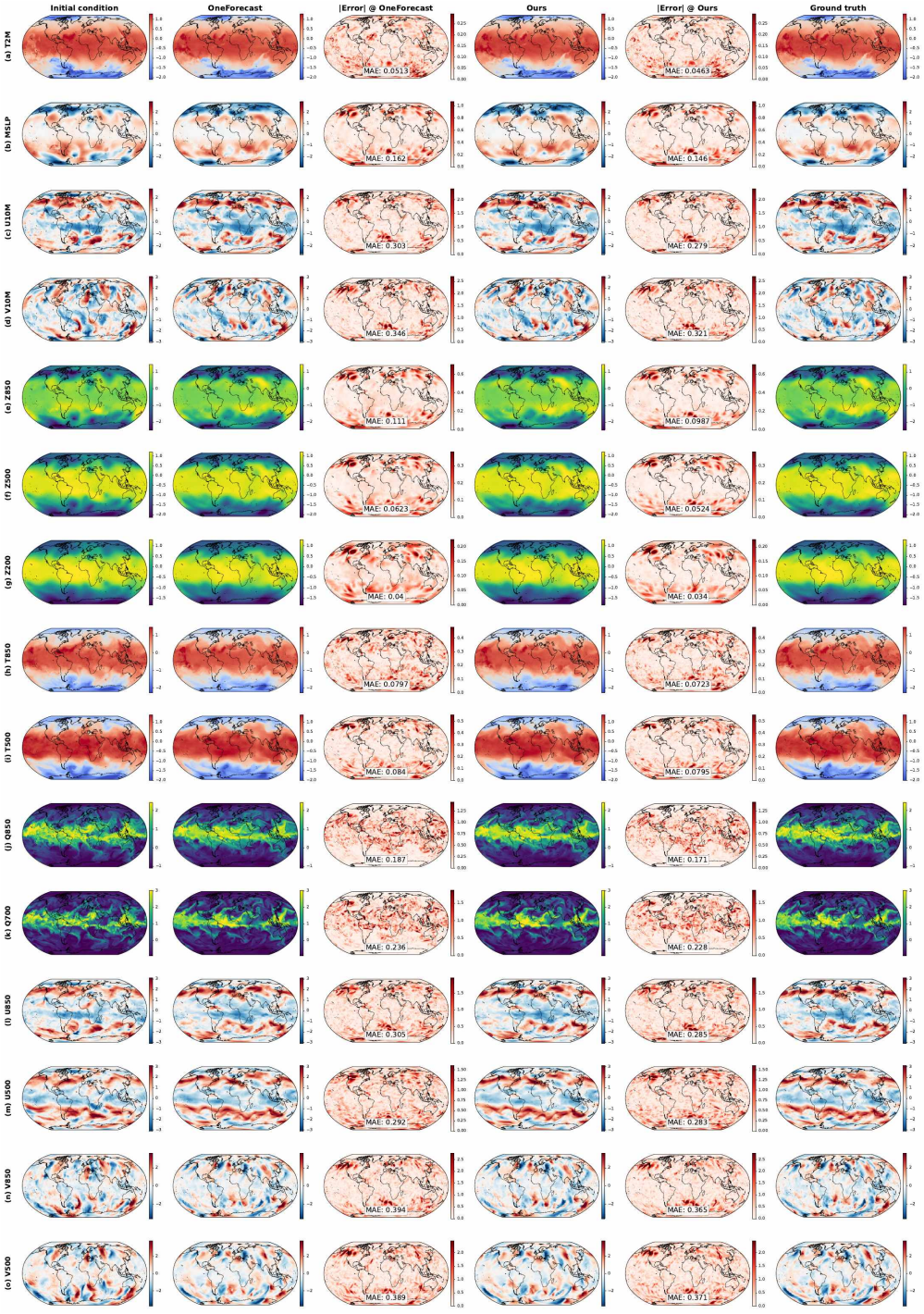

Figure 16: Visualization of global weather forecasts at a lead time of 4.5 days for different atmospheric variables.

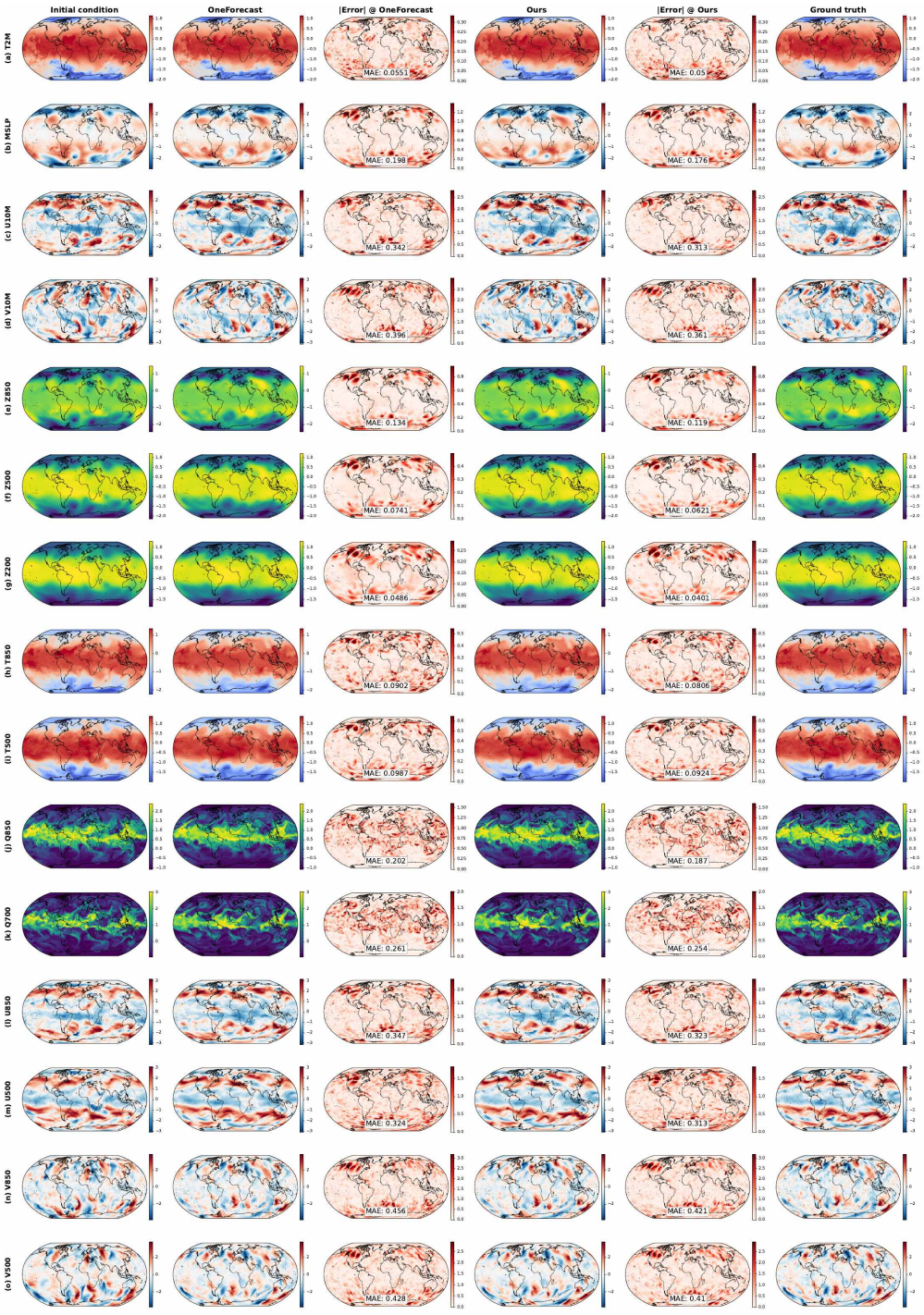

Figure 17: Visualization of global weather forecasts at a lead time of 5 days for different atmospheric variables.

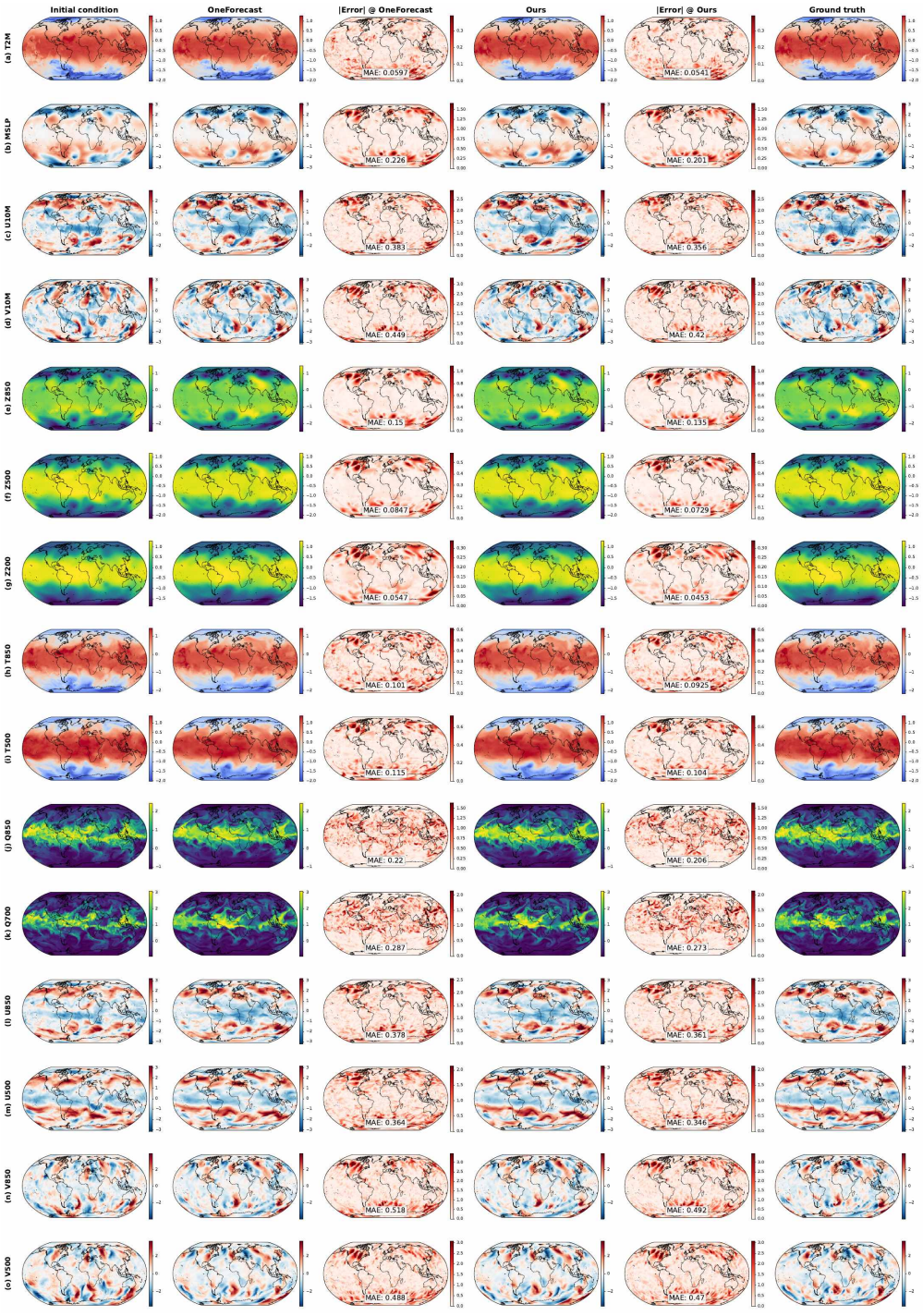

Figure 18: Visualization of global weather forecasts at a lead time of 5.5 days for different atmospheric variables.

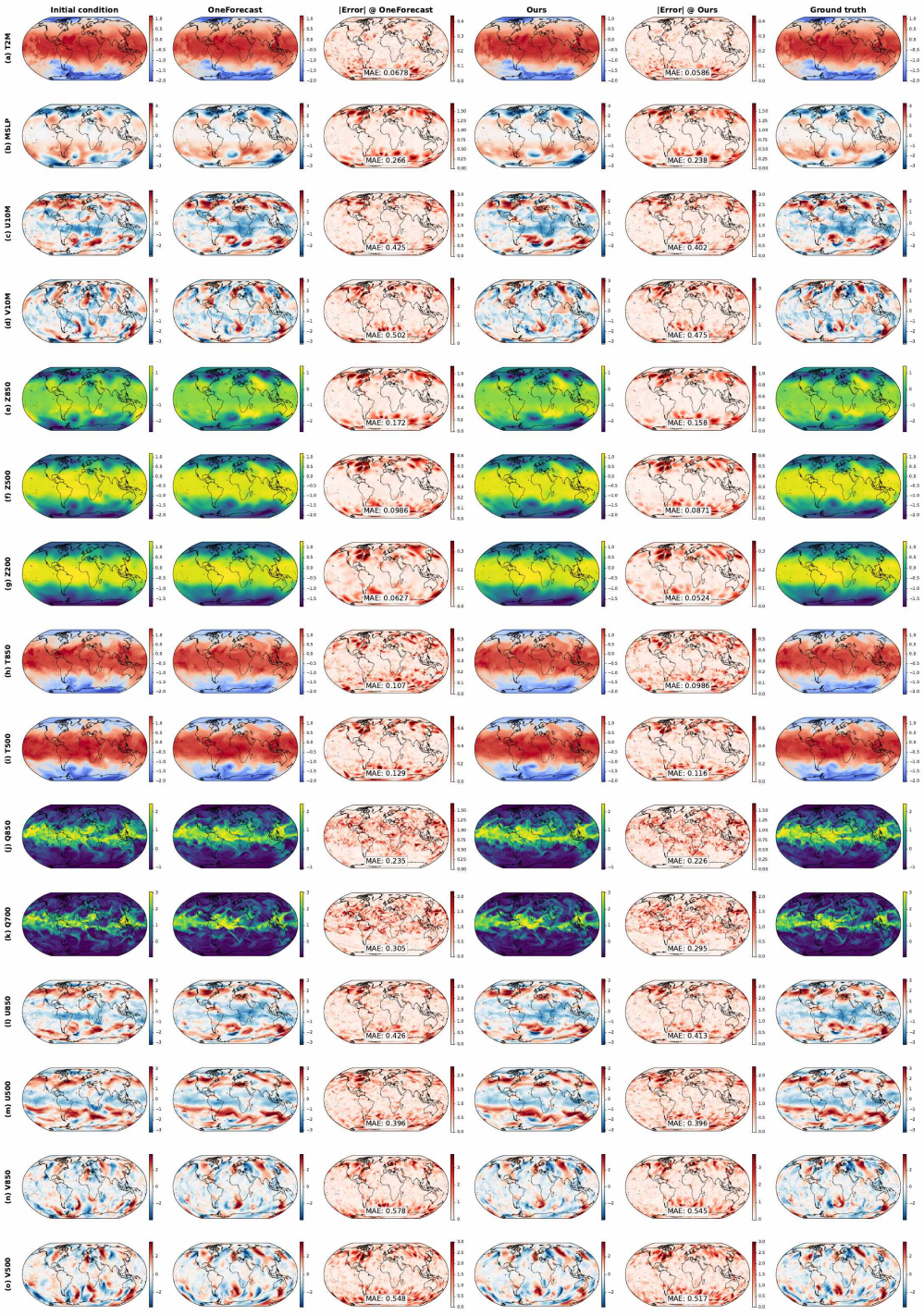

Figure 19: Visualization of global weather forecasts at a lead time of 6 days for different atmospheric variables.

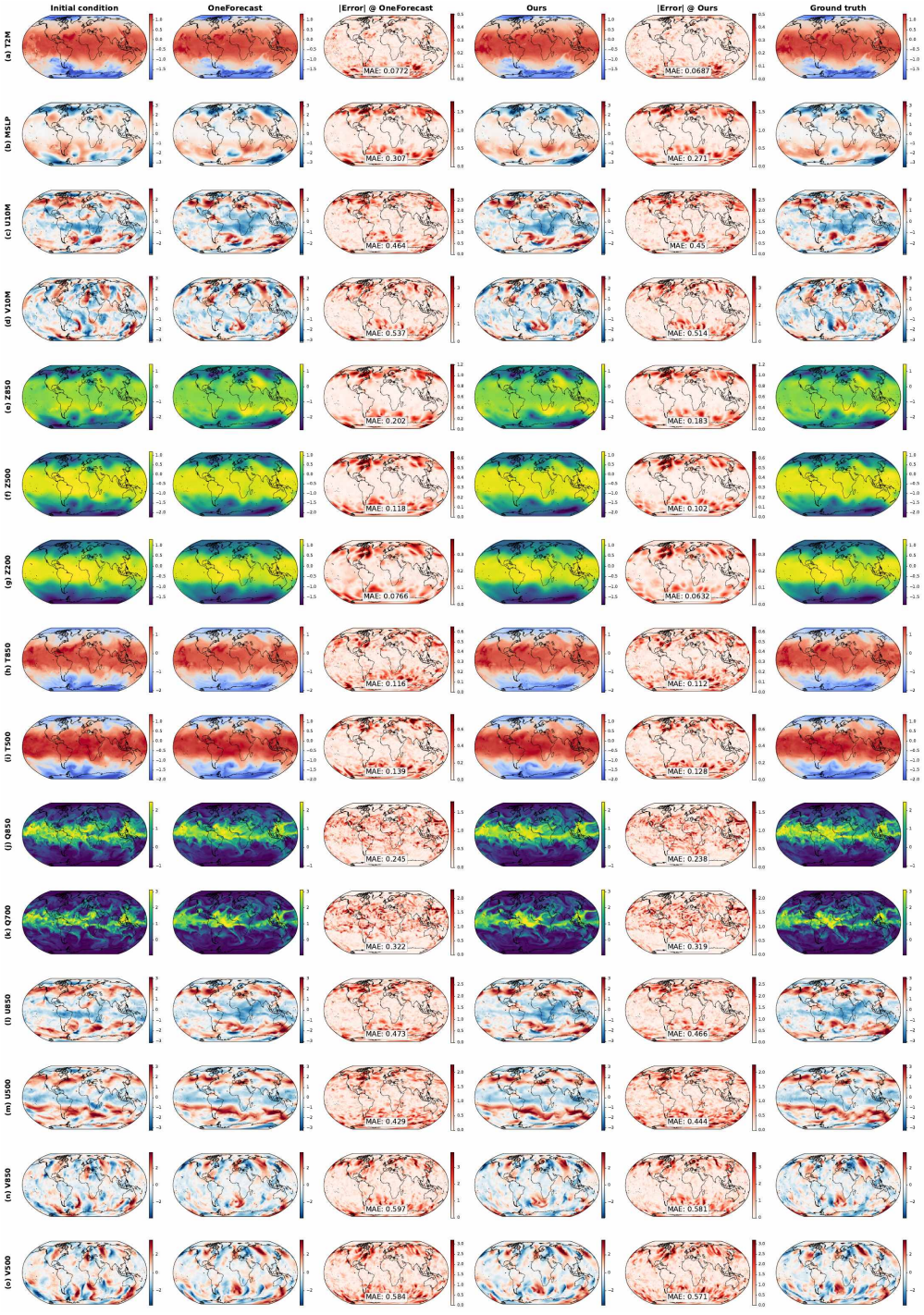

Figure 20: Visualization of global weather forecasts at a lead time of 6.5 days for different atmospheric variables.

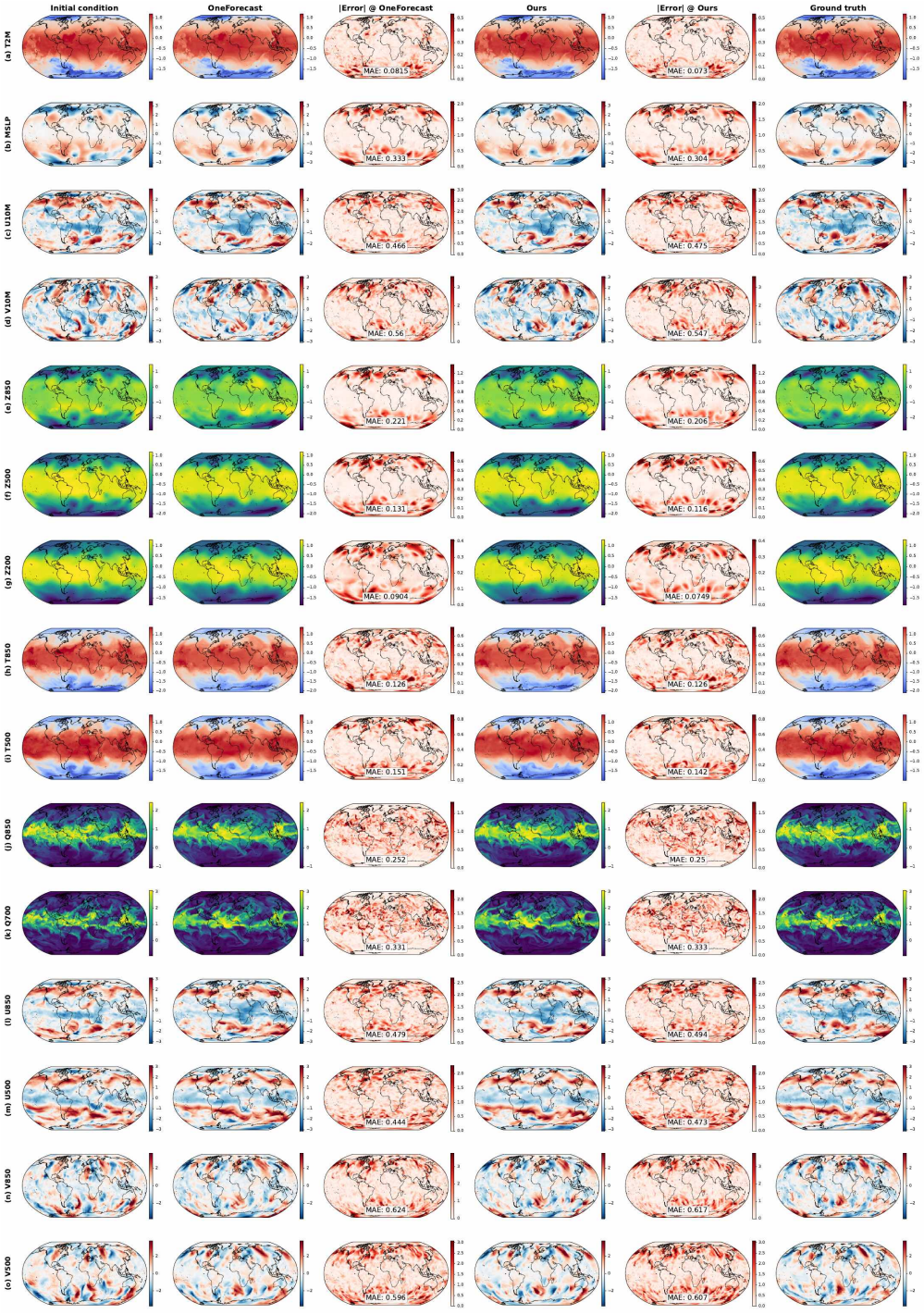

Figure 21: Visualization of global weather forecasts at a lead time of 7 days for different atmospheric variables.

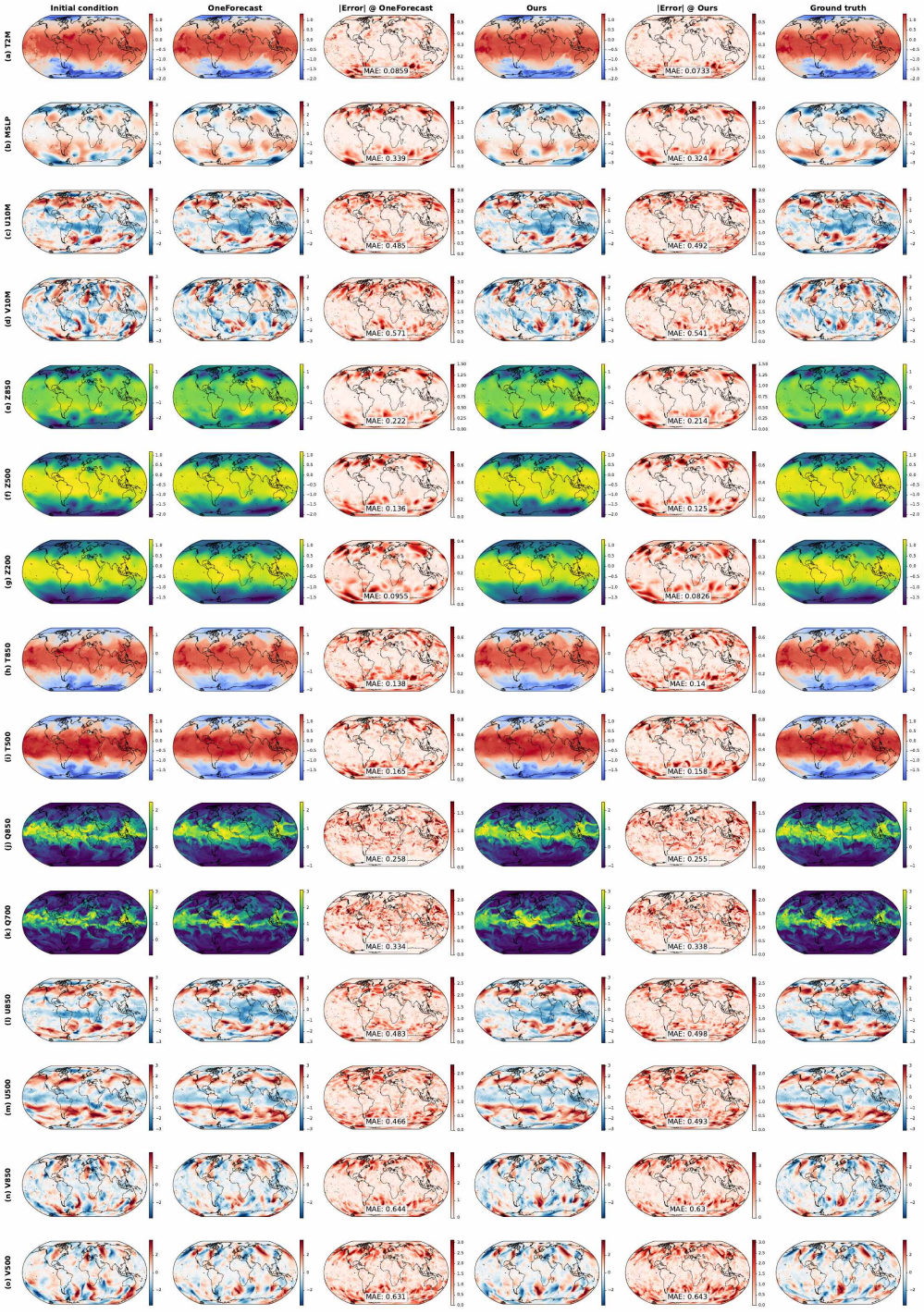

Figure 22: Visualization of global weather forecasts at a lead time of 7.5 days for different atmospheric variables.

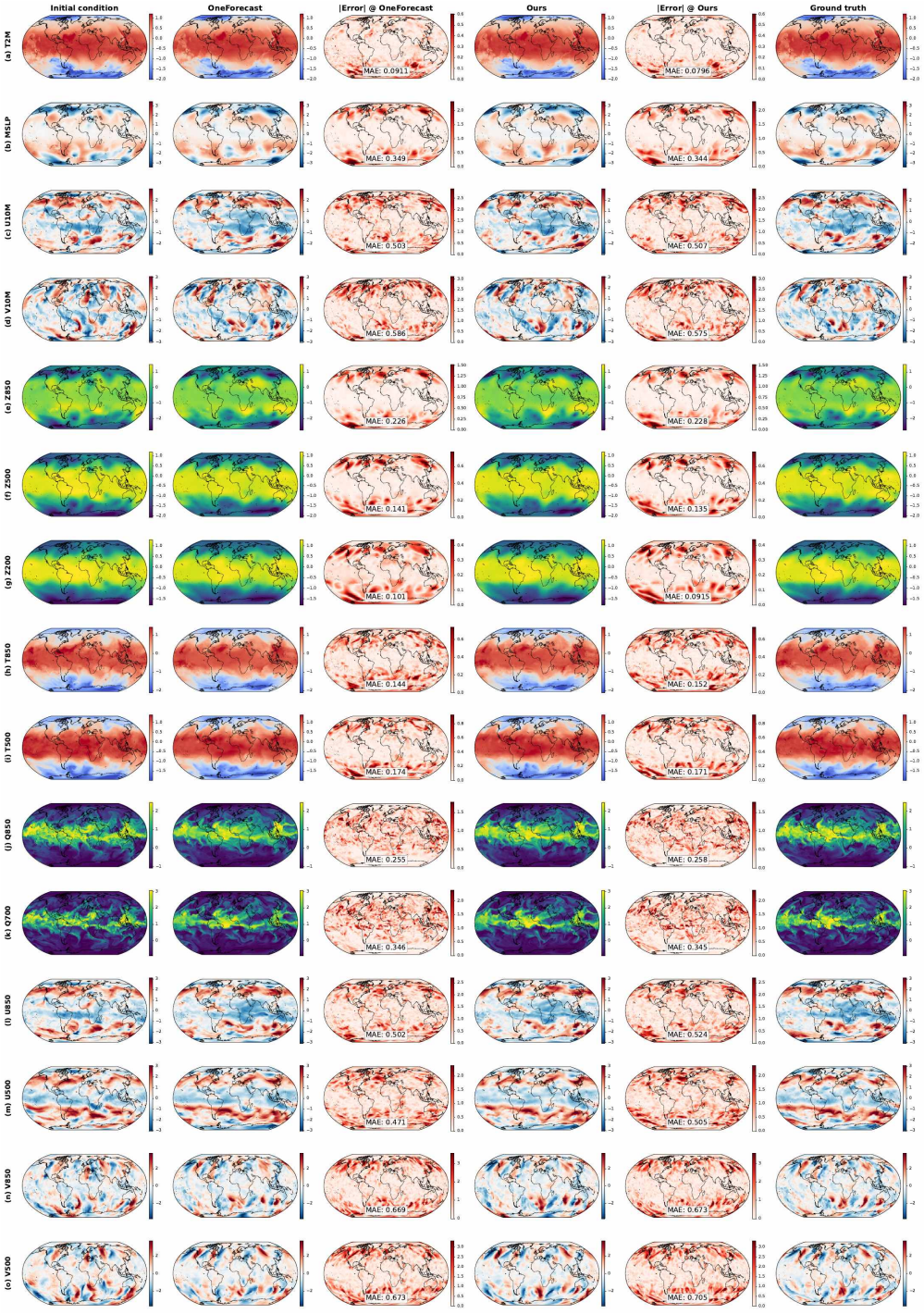

Figure 23: Visualization of global weather forecasts at a lead time of 8 days for different atmospheric variables.

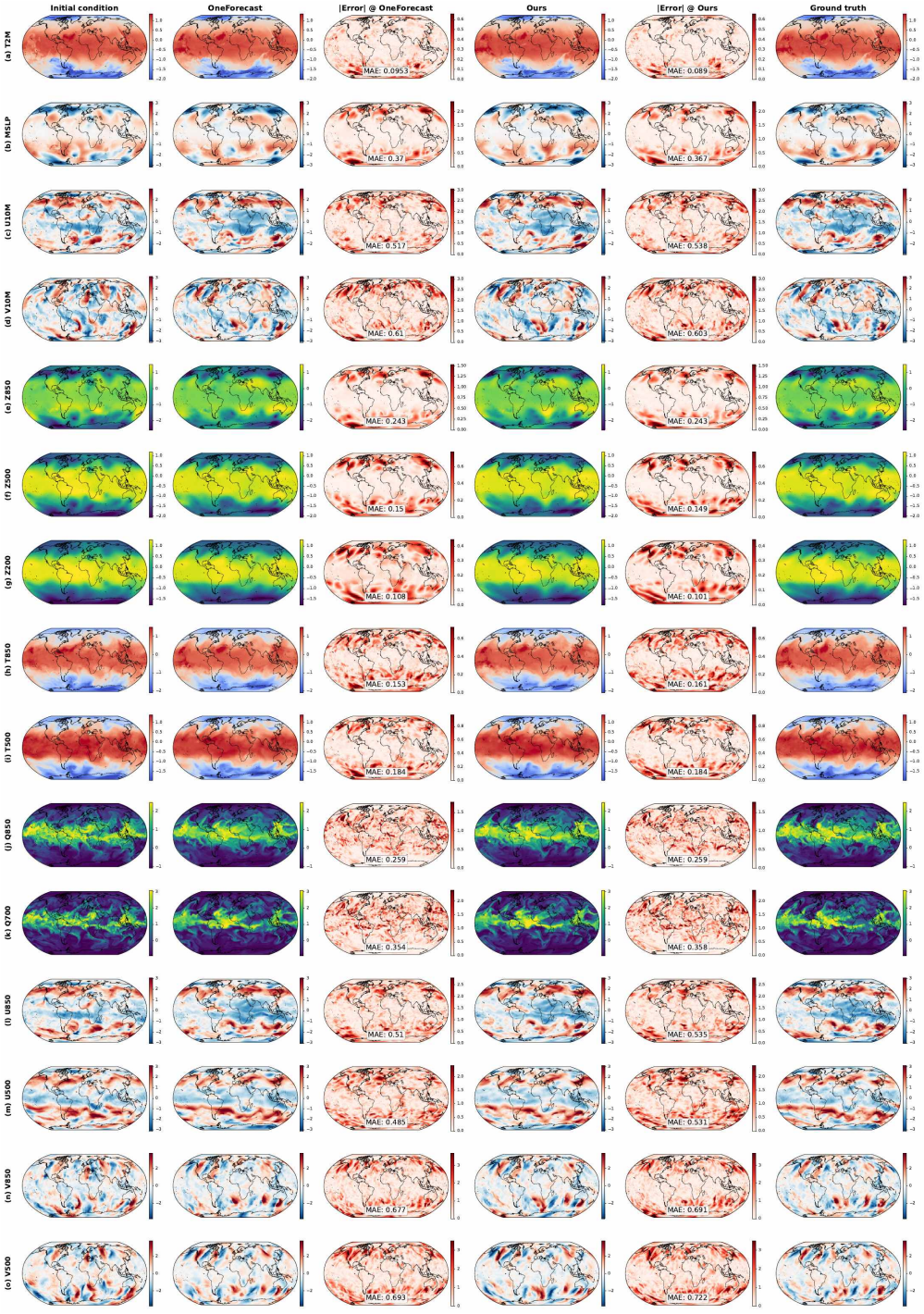

Figure 24: Visualization of global weather forecasts at a lead time of 8.5 days for different atmospheric variables.

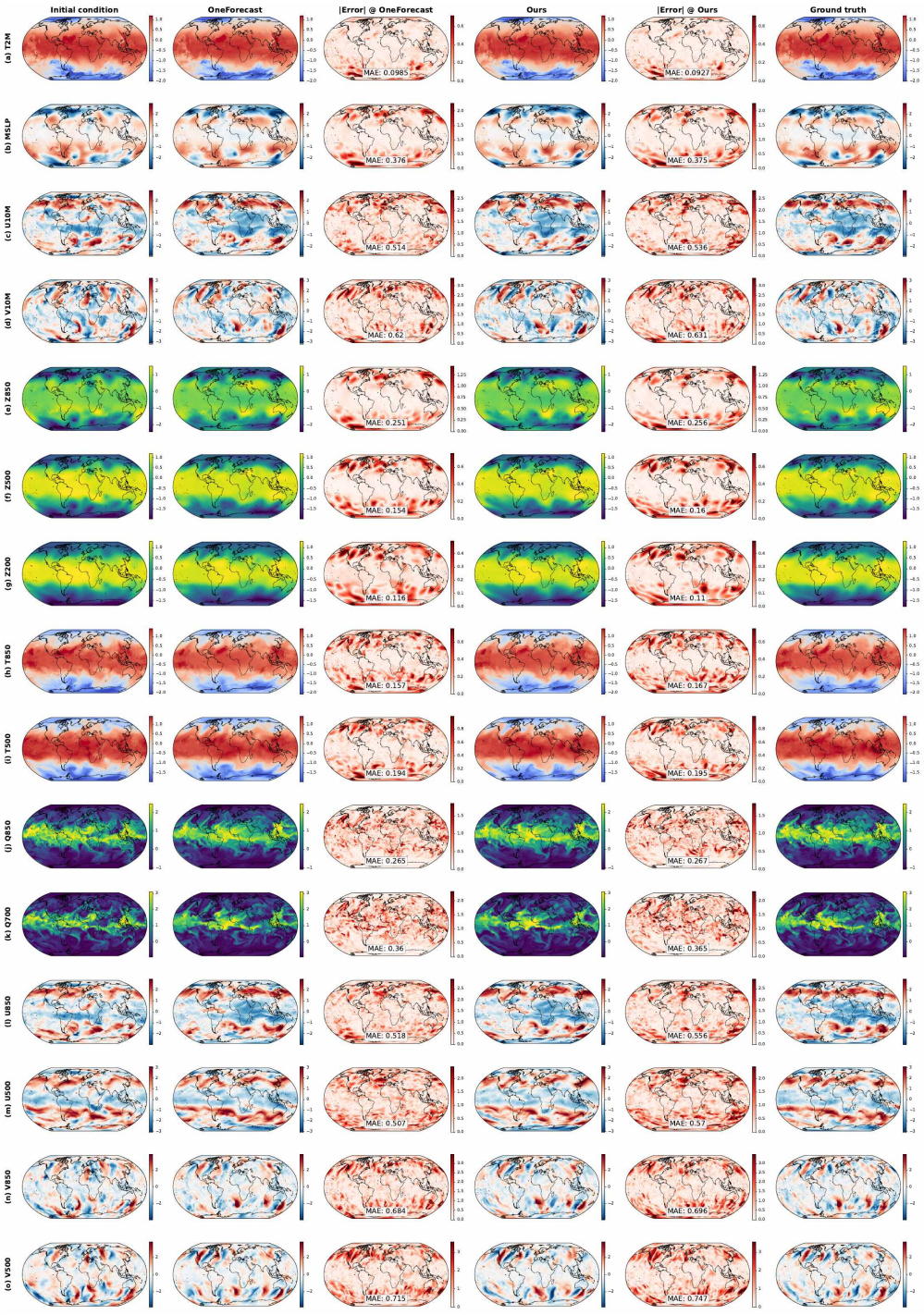

Figure 25: Visualization of global weather forecasts at a lead time of 9 days for different atmospheric variables.

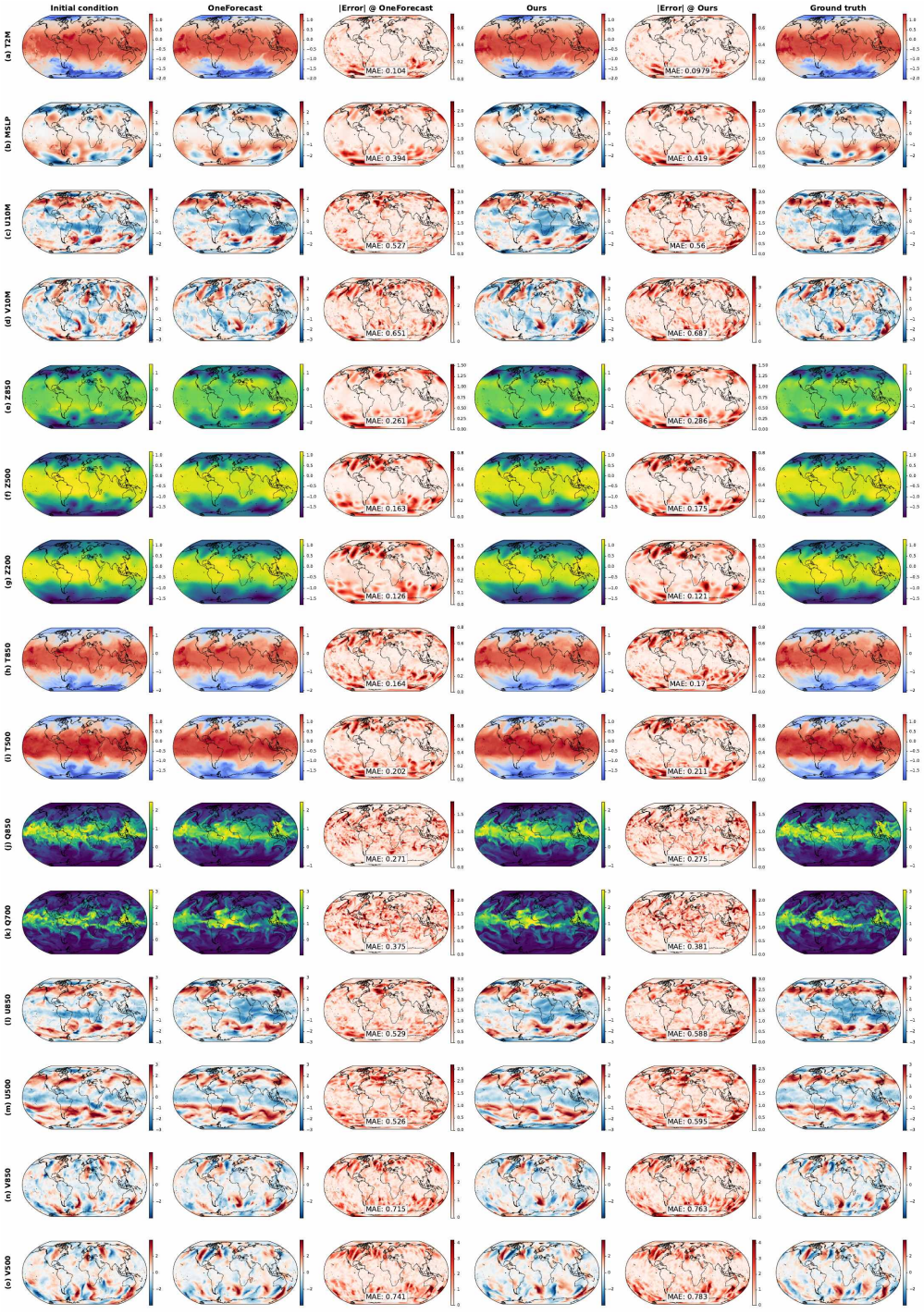

Figure 26: Visualization of global weather forecasts at a lead time of 9.5 days for different atmospheric variables.

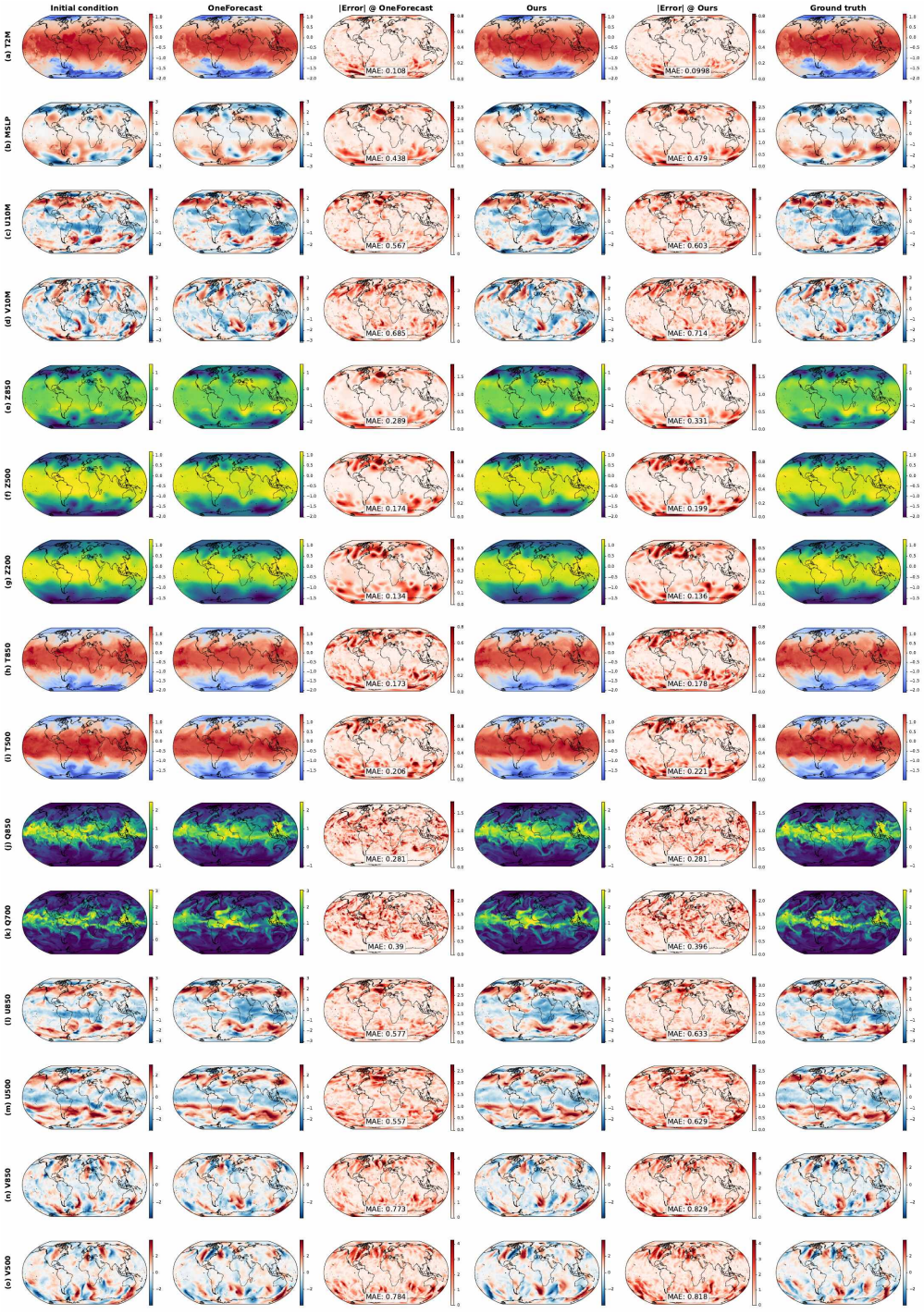

Figure 27: Visualization of global weather forecasts at a lead time of 10 days for different atmospheric variables.