PROLA: Principal-Orthogonal Low-rank Adaptation for Predictive Spatiotemporal Weather Downscaling

Minseo Yoon **Minseong Bae** **Sojin Lee** **Hyunwoo J. Kim***
Korea Advanced Institute of Science and Technology (KAIST)
{yms020615, bms2002, sojin.lee, hyunwoojkim}@kaist.ac.kr

Abstract

High-resolution weather prediction is important for resolving local atmospheric patterns missed by coarse global forecasts. Large pretrained weather backbones have improved global forecasting, but adapting them to produce future high-resolution fields from recent coarse states remains challenging and costly. We formulate this setting as Predictive Spatiotemporal Weather Downscaling (**PSWD**), a task that jointly refines spatial and temporal resolution while predicting future high-resolution trajectories. To this end, we propose a framework that decomposes prediction into spatial and temporal stages using a shared pretrained backbone, fixed numerical scaffolds, and lightweight residual heads. For backbone adaptation, we introduce **PROLA** (PRincipal-Orthogonal Low-rank Adaptation), which allocates a fixed low-rank budget between pretrained weights' principal directions and their orthogonal complement. PROLA further rescales rank-1 optimizer updates using gradient signals, improving adaptation without increasing trainable rank. Experiments in the PSWD setting show that PROLA outperforms representative low-rank adaptation baselines. Further experiments across diverse datasets and tasks support the broader applicability of PROLA, showing that its benefits extend beyond the primary weather downscaling setting.

1 Introduction

Machine learning-based weather modeling has primarily developed along two paradigms: weather forecasting [1–4], which predicts future states at fixed spatial resolution, and weather downscaling [5–8], which refines either spatial resolution at a fixed time or temporal resolution at a fixed grid. These standard formulations typically emphasize a subset of three axes: future prediction, spatial refinement, and temporal refinement. However, real-world applications that forecast a high-resolution target from a different, lower-resolution modality naturally involve all three axes. For example, minute-scale wind-power forecasting for renewable integration [9], sub-hourly precipitation nowcasting for flood-risk monitoring [10], and fine-scale aviation turbulence guidance for flight safety [11] all require future predictions at fine spatial and temporal resolution, a regime that no prior work has explored. This raises a natural question: *Can a single model jointly refine spatial and temporal resolution while extending predictions into the future?*

To answer this, we introduce **Predictive Spatiotemporal Weather Downscaling** (PSWD), the task of mapping past low-resolution states to future high-resolution states, with resolution gaps in both space and time. PSWD formalizes joint spatiotemporal refinement and future prediction as a unified learning task. It extends task-specific adaptation benchmarks for weather foundation models (WFMs) [12] to a setting that requires all three capabilities jointly. PSWD requires a model to infer fine-scale spatial structure and its evolution from the same coarse history. We address this task with a *scaffold-and-correct* pipeline built around a shared pretrained weather backbone. The spatial stage

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

predicts and refines a high-resolution state at the final lead time. The temporal stage then uses this predicted endpoint and the upsampled present state to construct the target trajectory. Fixed spatial upsampling and temporal interpolation provide the scaffolds, while lightweight correction heads learn the residual details.

In the scaffold-and-correct pipeline, the shared backbone provides the forecast that anchors both spatial and temporal refinement. To align this forecast with the final trajectory objective, we adapt the backbone jointly with the correction heads. Since the backbone already contains useful forecasting knowledge, this adaptation should build on its pretrained structure. At the same time, predicting high-resolution future trajectories may require corrections beyond the dominant pretrained directions. A single low-rank update provides a compact parameterization for parameter-efficient fine-tuning (PEFT) [13], but it does not explicitly separate these two needs under a fixed rank budget. This motivates an adaptation design that initializes one update path from the pretrained singular structure [14] while reserving another path for corrections outside the selected principal directions [15].

To this end, we propose **PROLA** (**PR**incipal-**O**rthogonal **L**ow-rank **A**daptation), a dual-branch low-rank adaptation method for the shared backbone. PROLA divides the fixed rank budget between a principal branch initialized from dominant singular components and an orthogonal branch projected outside the selected principal subspaces. The principal factors remain freely trainable after initialization, while the orthogonal branch uses fixed projectors defined by the pretrained weight. Beyond this allocation, PROLA rescales rank-1 optimizer updates using gradient-based branch and component scores, without increasing trainable rank. Branch scores adjust the relative update strength between the two paths, while component scores modulate individual updates within each path.

Experiments on WeatherBench 2 [16] with the pretrained WFM backbone [4] show that PROLA improves over representative low-rank PEFT baselines under matched low-rank budgets for multi-variable, multi-lead global weather downscaling. Ablations and diagnostics further support the scaffold-and-correct pipeline, the dual-branch design of PROLA, and gradient-guided scaling of optimizer updates. Together, these results support PSWD as a challenging evaluation setting and PROLA as an effective adaptation method for pretrained weather backbones, with broader applicability across datasets and tasks.

Our contributions are fourfold:

- We formulate **Predictive Spatiotemporal Weather Downscaling** (PSWD) to predict future high-resolution weather trajectories from recent coarse-resolution states, and introduce a **scaffold-and-correct** pipeline to address it.
- We propose **PROLA** (**PR**incipal-**O**rthogonal **L**ow-rank **A**daptation), a low-rank PEFT method that allocates a fixed rank budget between principal and orthogonal branches and rescales optimizer updates using gradient-based component scores.
- We show that PROLA outperforms representative PEFT baselines under a matched low-rank budget and narrows the performance gap between low-rank PEFT and full fine-tuning in our PSWD setting.
- We demonstrate PROLA's effectiveness across datasets and the transferability of its adaptation design beyond weather downscaling.

2 Predictive Spatiotemporal Weather Downscaling

In Section 2.1, we first formulate predictive spatiotemporal weather downscaling (PSWD) as a challenging task that jointly requires forecasting, spatial downscaling, and temporal downscaling. We then introduce a scaffold-and-correct pipeline designed to address this task in Section 2.2.

2.1 Task Formulation

Notation. Let $\mathbf{x}$ denote a context input and $\mathbf{y}$ a prediction target. For an atmospheric state with V variable channels, the input on the low-resolution grid is represented by $\mathbf{x}_t^{\mathrm{LR}} \in \mathbb{R}^{V \times H_{\mathrm{LR}} \times W_{\mathrm{LR}}}$ and the target on the high-resolution grid by $\mathbf{y}_t^{\mathrm{HR}} \in \mathbb{R}^{V \times H_{\mathrm{HR}} \times W_{\mathrm{HR}}}$. Let $\mathcal{H}_c$ denote the set of input lags at the coarse sampling interval δ_c, and $\mathcal{H}_f$ the set of target leads at the fine sampling interval δ_f. Relative to the anchor time t, the input at lag $h \in \mathcal{H}_c$ is $\mathbf{x}_{t-h}^{\mathrm{LR}}$, and the target at lead $h \in \mathcal{H}_f$ is $\mathbf{y}_{t+h}^{\mathrm{HR}}$.

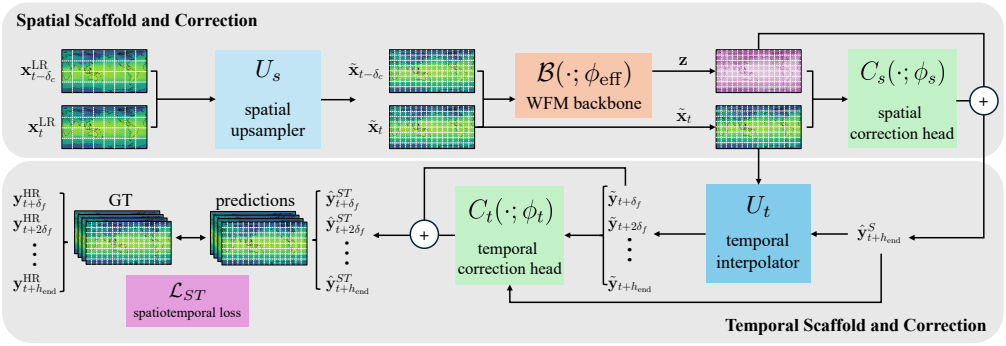


Figure 1: **Overview of the proposed scaffold-and-correct pipeline for PSWD.** Fixed numerical operators U_s and U_t and WFM backbone $\mathcal{B}$ build spatial and temporal scaffolds, while the residual heads C_s and C_t provide learned corrections.

With this indexing, $\mathcal{H}_c$ consists of the lags in $\{k\delta_c \mid k \in \mathbb{Z}_{\geq 0}\}$ within the context window, and $\mathcal{H}_f$ consists of the leads in $\{k\delta_f \mid k \in \mathbb{Z}_{>0}\}$ within the forecast window.

Comparison with existing weather tasks. We compare PSWD with related weather problem settings along three axes: forecasting, spatial downscaling and temporal downscaling. ***Forecasting*** predicts future states from past observations, changing only the forecast horizon while typically keeping the spatial and temporal grids fixed. ***Spatial downscaling*** reconstructs an HR state from its LR counterpart at the same valid time, $\mathbf{x}_t^{\mathrm{LR}} \mapsto \mathbf{y}_t^{\mathrm{HR}}$. It refines spatial resolution by recovering unresolved local structure, but does not change the temporal horizon. ***Temporal downscaling*** maps a coarsely sampled trajectory to a denser one on the same spatial grid over the same time interval. Existing settings usually address a single axis, whereas our target setting, ***Predictive Spatiotemporal Weather Downscaling*** **(PSWD)**, combines all three. At forecast initialization time t, PSWD takes coarse-step past and present LR inputs and predicts fine-step future HR targets:

$$\mathcal{X}_t^{\mathrm{LR}} = (\mathbf{x}_{t-h}^{\mathrm{LR}})_{h\in\mathcal{H}_c}, \qquad \mathcal{Y}_t^{\mathrm{HR}} = (\mathbf{y}_{t+h}^{\mathrm{HR}})_{h\in\mathcal{H}_f}, \tag{1}$$

where both sequences are ordered by increasing valid time and the targets are sampled more frequently than the inputs, *i.e.,* $\delta_f < \delta_c$. Thus, PSWD requires learning the mapping $\mathcal{X}_t^{\mathrm{LR}} \mapsto \mathcal{Y}_t^{\mathrm{HR}}$ to predict a future, higher-resolution, and temporally denser target trajectory.

Why PSWD is challenging. *(i) Coupled spatiotemporal prediction.* PSWD combines three task axes that are typically handled separately. Each target state is simultaneously a future prediction, a spatially refined field, and a temporally denser sub-step. The model must therefore infer fine-resolution future trajectories from coarse past inputs, making PSWD a coupled spatiotemporal prediction problem rather than a single-axis task shift. *(ii) Limited adaptation capacity.* While pretrained weather foundation models (WFMs) provide useful large-scale forecasting knowledge, PSWD requires adapting them to a richer output regime with finer spatial grids and denser future time steps. Since modern WFMs are increasingly large, fully fine-tuning them for every target resolution, lead-time structure, or data regime is often impractical. Recent work [12] therefore emphasizes PEFT as an important paradigm for WFM adaptation. PSWD tests whether limited trainable capacity can support a coupled shift in forecast horizon, spatial resolution, and temporal density.

2.2 Scaffold-and-Correct Pipeline

PSWD requires the model to forecast future atmospheric states while downscaling both spatially and temporally. We therefore introduce a unified *scaffold-and-correct* pipeline in Figure 1 that organizes prediction into spatial and temporal stages.

Overview. We adopt a *scaffold-and-correct* architecture: a coarse scaffold built primarily from pretrained or fixed components, followed by a learnable correction head that adds the missing detail. First, a fixed spatial upsampler U_s lifts coarse low-resolution inputs to the target HR grid, and a PEFT-adapted WFM backbone $\mathcal{B}$ predicts a spatial scaffold $\mathbf{z}$ at the fixed final lead time $h_{\mathrm{end}} = \max \mathcal{H}_f$. Then, the spatial correction head C_s adds a learned residual to produce the refined endpoint. A fixed temporal interpolator U_t bridges the upsampled present input and the predicted spatial endpoint to

form temporal scaffolds $\tilde{\mathbf{y}}_{t+h}$ over the target states at $h \in \mathcal{H}_f$, and the temporal correction head C_t adds another residual to produce the final outputs. The entire architecture shares a single backbone and is trained end-to-end under a single loss. The superscripts S and ST denote the spatial-stage and final spatiotemporal predictions, respectively.

Spatial scaffold and correction. Let $\phi_{\text{eff}} = (\phi_{\mathcal{B}}, \phi_{\text{PEFT}})$ collect the frozen pretrained backbone parameters and the trainable PEFT parameters, and let ϕ_s, ϕ_t denote the parameters of the two correction heads C_s, C_t. For the common two-input case $\mathcal{H}_c = \{\delta_c, 0\}$, a fixed spatial upsampler U_s (e.g., bilinear interpolation) aligns the coarse inputs to the target HR grid:

$$\tilde{\mathbf{x}}_{t-\delta_c} = U_s(\mathbf{x}^{\text{LR}}_{t-\delta_c}), \qquad \tilde{\mathbf{x}}_t = U_s(\mathbf{x}^{\text{LR}}_t). \tag{2}$$

The PEFT-adapted WFM backbone $\mathcal{B}$ produces a spatial scaffold $\mathbf{z}$ at the final lead time, and the spatial correction head C_s adds a learned residual:

$$\mathbf{z} = \mathcal{B}(\tilde{\mathbf{x}}_{t-\delta_c}, \tilde{\mathbf{x}}_t; \phi_{\text{eff}}), \qquad \hat{\mathbf{y}}^S_{t+h_{\text{end}}} = \mathbf{z} + C_s(\tilde{\mathbf{x}}_t, \mathbf{z}; \phi_s). \tag{3}$$

Here, U_s transports large-scale synoptic structure to the target grid and $\mathcal{B}$ advances it to the final lead time using its pretrained forecasting dynamics. Finally, C_s supplies the sub-grid corrections that interpolation alone cannot recover.

Temporal scaffold and correction. The endpoint prediction $\hat{\mathbf{y}}^S_{t+h_{\text{end}}}$ then anchors the temporal downscaling. For each $h \in \mathcal{H}_f$, a fixed temporal interpolator U_t (e.g., linear interpolation) between the upsampled present input $\tilde{\mathbf{x}}_t$ and the predicted endpoint $\hat{\mathbf{y}}^S_{t+h_{\text{end}}}$ forms a temporal scaffold $\tilde{\mathbf{y}}_{t+h}$, and the lead-time-conditioned temporal correction head C_t adds a residual:

$$\tilde{\mathbf{y}}_{t+h} = U_t(\tilde{\mathbf{x}}_t, \hat{\mathbf{y}}^S_{t+h_{\text{end}}}; h), \qquad \hat{\mathbf{y}}^{ST}_{t+h} = \tilde{\mathbf{y}}_{t+h} + C_t(\tilde{\mathbf{y}}_{t+h}, \hat{\mathbf{y}}^S_{t+h_{\text{end}}}, h; \phi_t). \tag{4}$$

U_t preserves the temporal continuity implied by the two endpoint states, while C_t learns the sub-step deviations that linear interpolation cannot capture.

The full pipeline thus concentrates the adaptation burden in the WFM backbone $\mathcal{B}$ and the two lightweight correction heads C_s and C_t, while U_s and U_t remain non-learnable scaffolds that preserve large-scale spatial and temporal structure.

Training objective. We train end-to-end with the variable-weighted state loss ℓ_{PT} used to pretrain the backbone, applied uniformly to all trajectory outputs:

$$\mathcal{L}_{ST} = \frac{1}{|\mathcal{H}_f|} \sum_{h \in \mathcal{H}_f} \ell_{\text{PT}}\big(\hat{\mathbf{y}}^{ST}_{t+h},\, \mathbf{y}^{\text{HR}}_{t+h}\big)\,. \tag{5}$$

No auxiliary loss is attached to the spatial endpoint $\hat{\mathbf{y}}^S_{t+h_{\text{end}}}$: the spatial stage is supervised through gradients flowing back from the temporal stage, ensuring both stages co-adapt for the joint trajectory rather than optimizing toward stage-specific targets. Section 5.1 evaluates these design choices.

3 PROLA: PRincipal-Orthogonal Low-rank Adaptation

Low-rank PEFT reduces the number of trainable parameters, but under a fixed budget, performance depends on how the limited capacity is allocated in the pretrained weight space and rescaled during optimization. PROLA addresses these two aspects through rank allocation and gradient-guided update scaling (Figure 2). In Section 3.1, we allocate the rank budget between principal and orthogonal branches, with the former supporting pretrained-aligned adaptation and the latter capturing residual corrections outside the selected principal directions. In Section 3.2, we further rescale updates from the base optimizer using rank-1 component scores computed from gradient magnitudes. Together, these mechanisms determine where the limited rank budget is allocated and how strongly each component is updated, allowing PROLA to preserve pretrained structure while directing updates toward task-specific corrections.

3.1 Budget Allocation via Dual-Branch Low-Rank Adaptation

We first allocate the fixed low-rank budget in the pretrained weight space. Given a pretrained weight matrix $W_0 \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, PROLA splits the total rank budget $r = r_p + r_o$, with fixed hyperparameters

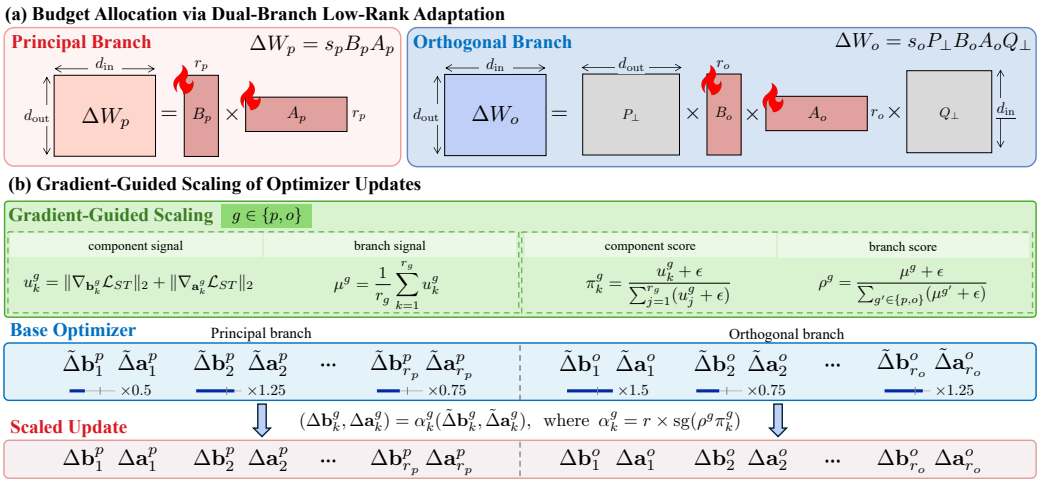


Figure 2: **Overview of PROLA.** **(a)** A fixed rank budget $r = r_p + r_o$ is split into a principal branch $\Delta W_p = s_p B_p A_p$, initialized from the top singular components of W_0, and an orthogonal branch $\Delta W_o = s_o P_\perp B_o A_o Q_\perp$, which captures corrections outside the selected principal subspaces. The effective weight combines the frozen residual weight and the two trainable branches as $W^{\text{eff}} = W^{\text{res}} + \Delta W_p + \Delta W_o$. **(b)** Updates from the base optimizer are scaled using gradient-guided branch and component scores. For each factor pair, α_k^g scales $(\tilde{\Delta}\mathbf{b}_k^g, \tilde{\Delta}\mathbf{a}_k^g)$, yielding $(\Delta \mathbf{b}_k^g, \Delta \mathbf{a}_k^g)$.

$r_p, r_o > 0$, between a principal branch for pretrained-aligned adaptation and an orthogonal branch for corrections outside the selected principal directions [14, 15]. The effective weight is

$$W^{\text{eff}} = W^{\text{res}} + \Delta W^{\text{dual}}, \qquad \Delta W^{\text{dual}} = \underbrace{\Delta W_p}_{\text{principal update}} + \underbrace{\Delta W_o}_{\text{orthogonal update}}, \tag{6}$$

where W^{res} is a frozen residual weight defined in Eq. (9). The two branches contain $r(d_{\text{out}} + d_{\text{in}})$ trainable parameters, matching a rank-r adapter.

Principal branch. We compute the SVD $W_0 = U \Sigma V^\top$ and select the top-r_p singular directions U_p, V_p and singular values Σ_p. The principal update is

$$\Delta W_p = s_p B_p A_p, \tag{7}$$

where $B_p \in \mathbb{R}^{d_{\text{out}} \times r_p}$, $A_p \in \mathbb{R}^{r_p \times d_{\text{in}}}$, and $s_p > 0$ is a scaling coefficient. We initialize $B_{p,0} = U_p \Sigma_p^{1/2}$ and $A_{p,0} = \Sigma_p^{1/2} V_p^\top$. After initialization, both factors are optimized without additional constraints.

Orthogonal branch. The remaining rank r_o is assigned to corrections outside the selected principal directions. With $P_\perp = I_{d_{\text{out}}} - U_p U_p^\top$ and $Q_\perp = I_{d_{\text{in}}} - V_p V_p^\top$, we define

$$\Delta W_o = s_o P_\perp B_o A_o Q_\perp, \tag{8}$$

where $B_o \in \mathbb{R}^{d_{\text{out}} \times r_o}$, $A_o \in \mathbb{R}^{r_o \times d_{\text{in}}}$, and $s_o > 0$ is a scaling coefficient. By construction, $U_p^\top \Delta W_o = 0$ and $\Delta W_o V_p = 0$, so this branch provides dedicated capacity outside both principal subspaces.

Function-preserving initialization. Following PiSSA [14], we residualize the frozen weight:

$$W^{\text{res}} = W_0 - \Delta W_{p,0}, \qquad \Delta W_{p,0} = s_p B_{p,0} A_{p,0}. \tag{9}$$

We set $B_{o,0} = 0$ and sample the entries of $A_{o,0}$ from $\mathcal{N}(0, \sigma_o^2)$, where σ_o controls the initialization scale. This gives $\Delta W_{o,0} = 0$ and $W_0^{\text{eff}} = W_0$, preserving the pretrained function.

Role of the rank split. The principal factors remain freely trainable, so a principal-only adapter can also learn corrections outside its initial singular subspaces. PROLA assigns r_p factors to the principal branch for pretrained-aligned adaptation and reserves r_o factors for a separate projected branch dedicated to orthogonal corrections. The rank split fixes the capacity assigned to these two paths under the same total rank budget, while the learned factors determine their contributions to the

effective update. To examine how the resulting update is distributed across the pretrained subspaces, let $\Delta W^{\text{eff}} := W^{\text{eff}} - W_0$, $P = U_p U_p^\top$, and $Q = V_p V_p^\top$, with $P_\perp = I - P$ and $Q_\perp = I - Q$. Then

$$\Delta W^{\text{eff}} = \underbrace{P\Delta W^{\text{eff}} Q}_{\text{principal block}} + \underbrace{P\Delta W^{\text{eff}} Q_\perp + P_\perp \Delta W^{\text{eff}} Q}_{\text{cross blocks}} + \underbrace{P_\perp \Delta W^{\text{eff}} Q_\perp}_{\text{orthogonal block}}. \tag{10}$$

Section 5.3 further compares the update energy across these blocks.

3.2 Gradient-Guided Scaling of Optimizer Updates

In Section 3.1, we allocate the fixed rank budget across the principal and orthogonal branches. We next rescale how strongly each rank-1 component is updated during optimization. For $g \in \{p, o\}$, where g indexes the principal branch p or the orthogonal branch o, let r_g denote the corresponding branch rank. Let $\mathbf{b}_k^g$ be the k-th column of B_g, and let $(\mathbf{a}_k^g)^\top$ be the k-th row of A_g. Then $(\mathbf{b}_k^g, \mathbf{a}_k^g)$ forms the k-th trainable rank-1 component of branch g, with $B_g A_g = \sum_{k=1}^{r_g} \mathbf{b}_k^g (\mathbf{a}_k^g)^\top$. For the orthogonal branch, this component is defined before applying the fixed projectors $P_\perp$ and $Q_\perp$.

Scaling rank-1 updates. Given this component-wise view, we attach a rank-1 update scale α_k^g to each factor pair. Let $\tilde{\Delta}\mathbf{b}_k^g$ and $\tilde{\Delta}\mathbf{a}_k^g$ denote the raw updates proposed by the base optimizer. The rescaled updates are

$$\Delta\mathbf{b}_k^g = \alpha_k^g \tilde{\Delta}\mathbf{b}_k^g, \qquad \Delta\mathbf{a}_k^g = \alpha_k^g \tilde{\Delta}\mathbf{a}_k^g. \tag{11}$$

This post-optimizer scaling leaves the base optimizer's raw-update computation unchanged and introduces no additional trainable parameters.

Gradient-guided scale computation. We compute α_k^g from gradient-based signals at the component and branch levels. The component signal u_k^g measures the gradient magnitude of the k-th rank-1 component, and the branch signal μ^g averages these signals within branch g:

$$u_k^g = \left\| \nabla_{\mathbf{b}_k^g} \mathcal{L}_{ST} \right\|_2 + \left\| \nabla_{\mathbf{a}_k^g} \mathcal{L}_{ST} \right\|_2, \qquad \mu^g = \frac{1}{r_g} \sum_{k=1}^{r_g} u_k^g. \tag{12}$$

These signals are normalized into branch and component scores and converted into the final rank-1 update scale:

$$\pi_k^g = \frac{u_k^g + \epsilon}{\sum_{j=1}^{r_g} (u_j^g + \epsilon)}, \qquad \rho^g = \frac{\mu^g + \epsilon}{\sum_{g' \in \{p,o\}} (\mu^{g'} + \epsilon)}, \qquad \alpha_k^g = r \times \mathrm{sg}(\rho^g \pi_k^g), \tag{13}$$

where π_k^g is the component score, ρ^g is the branch score, $r = r_p + r_o$, $\epsilon > 0$ is a numerical constant, and $\mathrm{sg}(\cdot)$ denotes stop-gradient.

Since ρ^g and π_k^g are normalized across branches and within each branch, respectively, α_k^g has unit average over all r low-rank components. Thus, the scaling reallocates update magnitude toward high-signal components while preserving the average layer-level update scale. Additional details and diagnostics are provided in Appendix B.

4 Experiments

4.1 Experimental Settings

For global PSWD, we adapt Aurora [4] on WeatherBench 2 [16]. Given $\mathcal{X}_t^{\text{LR}} = (\mathbf{x}_{t-\delta_c}^{\text{LR}}, \mathbf{x}_t^{\text{LR}})$ on the 5.625° grid with $\delta_c = 6\text{h}$, we predict $\mathcal{Y}_t^{\text{HR}}$ on the 1.5° grid at leads $\mathcal{H}_f = \{1\text{h}, \ldots, 6\text{h}\}$, *i.e.,* $\delta_f = 1\text{h}$. The 68 target channels comprise three surface variables and five upper-air variables at 13 pressure levels. We report results for five representative variables: 500 hPa geopotential (Z500), 850 hPa temperature (T850), 2m temperature (T2M), and 10m wind components (U10 and V10). Table 1 reports latitude-weighted RMSE and ACC at 3h (an intermediate lead) and 6h (the endpoint) as mean $\pm$ standard deviation over three seeds. All methods train both correction heads within the same scaffold-and-correct pipeline; Full FT also updates the backbone, while Head only keeps it frozen. The low-rank methods adapt the same attention projections. PROLA, LoRA [13], DoRA [17], PiSSA [14], and OPLoRA [15] use total rank $r = 8$, and AdaLoRA [18] targets rank 8.

Table 1: **Quantitative results on WeatherBench 2 [16].** Results are reported at 3h and 6h lead times. We report latitude-weighted RMSE (↓) and ACC (↑). Each cell shows the mean above and ± standard deviation below, over three seeds. Best displayed mean results among PEFT methods are boldfaced.

Method	Trainable Params	Z500		T850		T2M		U10		V10	
		RMSE	ACC	RMSE	ACC	RMSE	ACC	RMSE	ACC	RMSE	ACC
3h lead											
Full FT	1.26B	60.13 ± 1.61	0.998 ± 0.000	0.813 ± 0.002	0.987 ± 0.000	1.064 ± 0.009	0.979 ± 0.000	0.902 ± 0.004	0.975 ± 0.000	0.925 ± 0.009	0.974 ± 0.001
Head only	1.07M	94.93 ± 0.49	0.996 ± 0.000	1.049 ± 0.006	0.978 ± 0.000	2.083 ± 0.009	0.915 ± 0.001	1.648 ± 0.018	0.914 ± 0.001	1.563 ± 0.018	0.924 ± 0.002
LoRA	3.92M	54.72 ± 0.36	**0.999** ± 0.000	0.898 ± 0.002	0.984 ± 0.000	1.199 ± 0.007	0.973 ± 0.000	1.219 ± 0.008	0.954 ± 0.000	1.181 ± 0.002	0.957 ± 0.000
DoRA	4.16M	54.32 ± 0.35	**0.999** ± 0.000	0.893 ± 0.002	0.984 ± 0.000	1.191 ± 0.006	0.973 ± 0.000	1.209 ± 0.008	0.955 ± 0.000	1.173 ± 0.001	0.958 ± 0.000
AdaLoRA	5.35M	53.40 ± 0.27	**0.999** ± 0.000	0.884 ± 0.006	0.984 ± 0.000	1.176 ± 0.005	0.974 ± 0.000	1.175 ± 0.005	0.957 ± 0.000	1.158 ± 0.002	0.959 ± 0.000
PiSSA	3.92M	54.55 ± 0.89	**0.999** ± 0.000	0.893 ± 0.002	0.984 ± 0.000	1.198 ± 0.005	0.973 ± 0.000	1.215 ± 0.005	0.954 ± 0.000	1.175 ± 0.003	0.958 ± 0.000
OPLoRA	3.92M	54.94 ± 0.39	**0.999** ± 0.000	0.899 ± 0.003	0.984 ± 0.000	1.204 ± 0.007	0.973 ± 0.000	1.223 ± 0.007	0.954 ± 0.000	1.185 ± 0.003	0.957 ± 0.000
PROLA	3.92M	**50.58** ± 0.42	**0.999** ± 0.000	**0.864** ± 0.003	**0.985** ± 0.000	**1.129** ± 0.002	**0.976** ± 0.000	**1.143** ± 0.003	**0.960** ± 0.000	**1.120** ± 0.001	**0.962** ± 0.000
6h lead											
Full FT	1.26B	106.68 ± 2.63	0.995 ± 0.000	1.114 ± 0.008	0.976 ± 0.000	1.323 ± 0.022	0.970 ± 0.001	1.135 ± 0.012	0.961 ± 0.001	1.209 ± 0.010	0.956 ± 0.001
Head only	1.07M	144.03 ± 1.47	0.990 ± 0.000	1.315 ± 0.008	0.965 ± 0.001	2.656 ± 0.021	0.856 ± 0.001	1.947 ± 0.024	0.877 ± 0.002	1.921 ± 0.018	0.883 ± 0.003
LoRA	3.92M	79.17 ± 0.33	**0.997** ± 0.000	0.994 ± 0.003	0.980 ± 0.000	1.280 ± 0.012	0.969 ± 0.001	1.314 ± 0.005	0.946 ± 0.000	1.306 ± 0.002	0.948 ± 0.000
DoRA	4.16M	78.40 ± 0.30	**0.997** ± 0.000	0.989 ± 0.003	0.980 ± 0.000	1.272 ± 0.013	0.970 ± 0.001	1.306 ± 0.006	0.947 ± 0.000	1.298 ± 0.001	0.948 ± 0.000
AdaLoRA	5.35M	77.51 ± 0.94	**0.997** ± 0.000	0.983 ± 0.006	0.981 ± 0.000	1.251 ± 0.002	0.971 ± 0.001	1.282 ± 0.004	0.949 ± 0.000	1.279 ± 0.002	0.950 ± 0.000
PiSSA	3.92M	78.43 ± 0.93	**0.997** ± 0.000	0.989 ± 0.004	0.980 ± 0.000	1.251 ± 0.009	0.971 ± 0.001	1.298 ± 0.004	0.948 ± 0.000	1.289 ± 0.003	0.949 ± 0.000
OPLoRA	3.92M	79.60 ± 0.38	**0.997** ± 0.000	0.996 ± 0.003	0.980 ± 0.000	1.281 ± 0.012	0.969 ± 0.000	1.319 ± 0.005	0.946 ± 0.000	1.311 ± 0.002	0.947 ± 0.000
PROLA	3.92M	**72.35** ± 0.64	**0.997** ± 0.000	**0.956** ± 0.002	**0.982** ± 0.000	**1.212** ± 0.006	**0.973** ± 0.001	**1.240** ± 0.005	**0.952** ± 0.000	**1.234** ± 0.002	**0.953** ± 0.000

We extend the evaluation to ERA5-to-CERRA downscaling [19, 20], using coarse ERA5 inputs at $t - 6\text{h}$ and t to predict high-resolution CERRA fields over Central Europe at leads $\mathcal{H}_f = \{3\text{h}, 6\text{h}\}$. Table 2 reports latitude-weighted RMSE for the same five variables at 3h. Regional precipitation forecasting follows the protocol in [12], using ERA5 inputs at $t - 12\text{h}$ and t to predict 6h precipitation totals ending at each forecast valid time. Table 3 reports SEEPS, RMSE, and ACC at 12h. Both tasks compare PROLA with LoRA, PiSSA, and OPLoRA. See Appendices C–F for experimental details and metric definitions.

4.2 Main Results

Table 1 shows that PROLA achieves the lowest RMSE and highest ACC among PEFT methods across all reported variables and leads. Relative to AdaLoRA (36% more trainable parameters), PROLA reduces Z500 RMSE by 5.3% at 3h (50.58 vs. 53.40) and 6.7% at 6h (72.35 vs. 77.51), with reductions of 2.3–4.0% and 2.7–3.5% on the remaining variables. Notably, despite adapting only 3.92M parameters (under 0.4% of Full FT), PROLA outperforms Full FT on Z500 at both leads and on T850 and T2M at 6h. These results suggest that structured low-rank adaptation provides useful implicit regularization in PSWD by limiting backbone update flexibility.

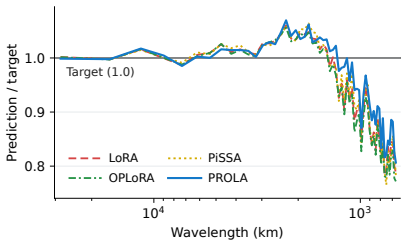


Figure 3: **Spatial spectral fidelity.** Prediction-to-target ratios of mean spherical harmonic power for T2M at 6h over test cases. A ratio of one matches target power.

Figure 3 compares prediction-to-target ratios of mean spherical harmonic power for T2M at 6h. The ratios fall below one at short wavelengths, indicating underestimated fine-scale temperature variability. PROLA remains closer to one at shorter wavelengths, supporting improved spatial spectral fidelity. Full per-variable and per-lead results, further analysis, and qualitative forecasts are provided in Appendices G and H.

Table 2: **ERA5-to-CERRA.**
Latitude-weighted RMSE (↓) at 3h.

Method	Z500	T850	T2M	U10	V10
LoRA	72.14	1.076	2.150	1.370	1.327
PiSSA	71.80	1.072	2.126	1.354	1.311
OPLoRA	72.56	1.081	2.132	1.371	1.332
PROLA	**68.44**	**1.028**	**2.021**	**1.316**	**1.274**

Table 3: **Precipitation.**
6h totals at 12h lead (RMSE in cm).

Method	SEEPS ↓	RMSE ↓	ACC ↑
LoRA	0.608	0.212	0.660
PiSSA	0.612	0.213	0.656
OPLoRA	0.605	0.212	0.661
PROLA	**0.590**	**0.208**	**0.676**

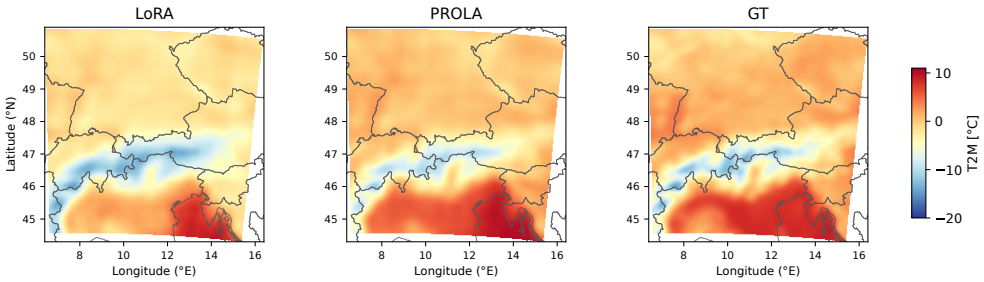


Figure 4: **Qualitative ERA5-to-CERRA prediction at 6h.** T2M predictions from LoRA and PROLA, and the CERRA target, valid at 12:00 UTC on 4 February 2018.

4.3 Generalization to Regional Weather Tasks

ERA5-to-CERRA downscaling. Table 2 shows that PROLA achieves the lowest latitude-weighted RMSE on all five reported variables at 3h. Relative to PiSSA, the strongest baseline in this comparison, PROLA reduces Z500 RMSE by 4.7% (68.44 vs. 71.80) and T2M RMSE by 4.9% (2.021 vs. 2.126). Figure 4 compares the 6h T2M forecasts with the CERRA target. In this example, PROLA reproduces the contrast between the cooler Alpine region and the warmer areas to its south more closely than LoRA. PROLA also better matches the warmer temperatures north of the Alps, where LoRA predicts a cooler field. See Appendix G for the full per-lead results.

Regional precipitation forecasting. Table 3 shows that PROLA achieves the lowest SEEPS and RMSE and the highest ACC among the compared PEFT methods at the 12h lead. Compared with OPLoRA, PROLA reduces SEEPS from 0.605 to 0.590 and RMSE from 0.212 to 0.208 cm, improving the prediction of both precipitation categories and accumulated amounts. The higher ACC (0.676 vs. 0.661) also indicates closer agreement with the spatial pattern of precipitation anomalies. Together with the ERA5-to-CERRA results, these gains show that PROLA outperforms the low-rank baselines across both regional downscaling and precipitation forecasting. See Appendix G for results at additional lead times.

5 Analysis

5.1 Ablation of Scaffold-and-Correct Pipeline

Table 4 compares direct prediction across PEFT methods and ablates the scaffold-and-correct pipeline with PROLA. Direct prediction jointly outputs all target lead times without the spatial-to-temporal decomposition. All pipeline ablations use PROLA and remove the spatial correction head C_s, the temporal correction head C_t, or both, leaving the corresponding scaffolds uncorrected. The full pipeline achieves the lowest RMSE across all five variables at 3h, reducing Z500 RMSE from 53.13 with direct prediction to 50.30. Removing both correction heads raises 3h Z500 RMSE to 87.73, indicating that the scaffolds alone are insufficient for PSWD. Removing C_t alone yields a much higher 3h Z500 RMSE than removing C_s alone (84.53 vs. 53.35), highlighting the role of temporal correction at intermediate leads. Although $w/o\ C_s$ remains competitive on T2M and U10 at 6h, its higher Z500 and T850 errors support spatial endpoint correction as a temporal anchor.

Table 4: **Direct prediction and pipeline ablation.** Latitude-weighted RMSE (↓) at 3h and 6h. Best results in the pipeline ablation are boldfaced.

	3h lead					6h lead				
Method / Variant	Z500	T850	T2M	U10	V10	Z500	T850	T2M	U10	V10
Direct prediction										
LoRA	58.91	0.979	1.256	1.481	1.484	129.60	1.214	1.394	1.755	1.781
PiSSA	56.47	0.942	1.191	1.272	1.239	128.00	1.152	1.292	1.392	1.405
OPLoRA	58.15	0.983	1.254	1.463	1.457	131.16	1.219	1.388	1.741	1.735
PROLA	53.13	0.898	1.165	1.211	1.177	125.07	1.095	1.289	1.358	1.344
Scaffold-and-correct ablation (PROLA)										
w/o C_s, C_t	87.73	0.959	1.307	1.277	1.270	111.87	1.037	1.289	1.339	1.371
w/o C_t	84.53	0.938	1.290	1.259	1.254	108.80	1.016	1.281	1.332	1.365
w/o C_s	53.35	0.880	1.138	1.155	1.133	78.25	0.974	**1.209**	**1.233**	1.237
Full pipeline	**50.30**	**0.866**	**1.128**	**1.140**	**1.119**	**73.07**	**0.958**	1.211	1.236	**1.234**

Table 5: **Ablation studies on PROLA design.** Latitude-weighted RMSE (↓) at 3h and 6h. The design ablations use $r_p + r_o = 8$. ✓marks indicate update scaling; ρ and π denote branch score and component score. Best results are boldfaced.

				3h lead					6h lead				
Configuration	r_p	r_o	Scaling	Z500	T850	T2M	U10	V10	Z500	T850	T2M	U10	V10
(i) *Branch composition and update scaling*													
Principal	8	0	–	53.66	0.894	1.193	1.209	1.178	78.55	0.987	1.248	1.294	1.290
Principal	8	0	✓	53.06	0.887	1.184	1.198	1.166	77.71	0.979	1.241	1.289	1.285
Orthogonal	0	8	–	54.54	0.899	1.197	1.216	1.188	79.83	0.994	1.272	1.314	1.314
Orthogonal	0	8	✓	52.81	0.886	1.175	1.189	1.157	76.67	0.977	1.260	1.284	1.278
Both branches	4	4	–	51.36	0.875	1.143	1.165	1.133	74.38	0.967	1.221	1.256	1.250
(ii) *Rank allocation*													
Principal-heavy	6	2	✓	51.46	0.876	1.153	1.161	1.136	74.19	0.970	1.225	1.249	1.244
Orthogonal-heavy	2	6	✓	52.52	0.890	1.176	1.200	1.160	76.35	0.983	1.263	1.285	1.282
(iii) *Scaling components*													
ρ only	4	4	✓	52.16	0.885	1.168	1.188	1.151	76.06	0.978	1.240	1.273	1.264
π only	4	4	✓	51.94	0.882	1.155	1.176	1.144	75.45	0.976	1.235	1.267	1.261
PROLA	4	4	✓	**50.30**	**0.866**	**1.128**	**1.140**	**1.119**	**73.07**	**0.958**	**1.211**	**1.236**	**1.234**

5.2 PROLA Branch and Scaling Ablation

Table 5 compares branch configurations and update scaling at a fixed total rank of $r_p + r_o = 8$. The results in **(i)** show that even without update scaling, 3h Z500 RMSE is lower with both branches (51.36) than with the principal branch alone (53.66) or the orthogonal branch alone (54.54). Update scaling improves both single-branch variants and further lowers 3h Z500 RMSE to 50.30 when both branches are used. With scaling enabled, the balanced $4+4$ split outperforms both uneven allocations in **(ii)** on all five variables at both leads. At 6h, Z500 RMSE is 73.07 for $4+4$, compared with 74.19 for $6+2$ and 76.35 for $2+6$. The comparisons in **(iii)** keep the $4+4$ split fixed. Using the branch score ρ alone raises 6h Z500 RMSE to 76.06, while using the component score π alone raises it to 75.45. Full PROLA (bottom row) combines both scores and achieves the lowest RMSE across all five variables at 3h and 6h. See Appendix I for the full per-lead comparisons.

5.3 Update Geometry under a Fixed Rank Budget

We apply Eq. (10) to compare how PROLA and PiSSA use the same rank budget. Let $\mathcal{A}$ index the adapted attention matrices. For each $\ell \in \mathcal{A}$, we consider the effective update $\Delta W_\ell^{\text{eff}} = W_\ell^{\text{eff}} - W_{\ell,0}$, which includes the learned changes from both branches in PROLA. We decompose this update using the top r left and right singular vectors of the pretrained weight $W_{\ell,0}$. All methods are evaluated using these same reference subspaces. Let ΔW_ℓ^b denote the component within both reference subspaces ($b = \text{prin}$), outside both (orth), or within one and outside the other (cross, summing both cross blocks). The energy share of block b is obtained by summing its

squared Frobenius norm across adapted matrices and normalizing by the total update energy: $\bar{e}_b = \sum_{\ell \in \mathcal{A}} \|\Delta W_\ell^b\|_F^2 / \sum_{\ell \in \mathcal{A}} \|\Delta W_\ell^{\text{eff}}\|_F^2$.

Figure 5 shows that PROLA more than doubles the orthogonal share of update energy relative to PiSSA (34% vs. 16%). PROLA retains 23% of its update energy in the principal block and 42% in the cross blocks, whereas PiSSA places 18% in the principal block and concentrates 66% in the cross blocks. The principal factors in PiSSA remain freely trainable after initialization, allowing it to learn corrections outside its initial principal subspaces. Under the same rank budget, PROLA achieves a larger orthogonal share, accompanied by a lower cross-block share and a higher principal share. Thus, its increased share of orthogonal corrections is maintained alongside updates within the pretrained principal subspaces. For reference, the orthogonal share is 93% for unconstrained Full FT under the same partition. See Appendix A for the projector definitions and energy aggregation used in this comparison.

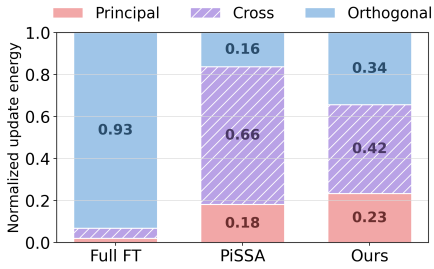


Figure 5: **Blockwise update energy.** Fractions of total update energy in the principal, cross, and orthogonal blocks.

6 Related Work

Data-driven weather modeling. Data-driven weather forecasting has progressed from early global models [21, 22] to systems achieving competitive or state-of-the-art performance on global benchmarks [1–3, 16, 23–26], alongside hybrid physics-ML and probabilistic systems [27, 28]. Weather downscaling reconstructs high-resolution fields from low-resolution inputs through spatial statistical or generative refinement [5, 8, 29–33], with fewer studies addressing time-evolving or joint spatiotemporal refinement [7, 34, 35]. Pretrained weather foundation models (WFMs) support diverse atmospheric tasks [4, 36–38], with their growing scale motivating PEFT [12]. Existing work often treats forecasting, spatial downscaling, temporal refinement, and WFM adaptation separately, whereas PSWD couples forecasting with spatiotemporal refinement in one objective and provides a comprehensive PEFT benchmark for WFMs.

Low-rank parameter-efficient fine-tuning. PEFT avoids costly full fine-tuning by adapting large pretrained backbones with few trainable parameters [39]. LoRA trains low-rank factors with a frozen backbone [13]; variants add quantization, dynamic rank selection, parameter sharing, adaptive rank allocation, factor-wise learning rates, and magnitude-direction decomposition [17, 18, 40–43]. Other methods use SVD/decomposition-based initialization [14, 44, 45] to exploit pretrained weight structure, or orthogonality constraints to reduce interference with pretrained or previously learned subspaces [15, 46, 47]. Related approaches distribute principal components across devices [48], restrict orthogonal transformations to selected subspaces [49, 50], or combine coefficient updates within a fixed principal basis with orthogonal-complement adaptation [51]. PROLA splits a fixed rank budget between principal and orthogonal branches, trains its principal factors, and scales optimizer updates using gradient-based branch and rank-1 component scores.

7 Conclusion

We formulated Predictive Spatiotemporal Weather Downscaling (PSWD) as predicting future high-resolution global weather trajectories from recent coarse-resolution global states. We addressed PSWD with a spatial-to-temporal scaffold-and-correct pipeline that reuses a shared pretrained backbone and reconstructs target trajectories with fixed numerical scaffolds and lightweight residual heads. For parameter-efficient backbone adaptation, we proposed PROLA (Principal-Orthogonal Low-rank Adaptation), which splits a fixed rank budget across principal and orthogonal branches and rescales rank-1 optimizer updates using gradient signals. Experiments show that PROLA outperforms low-rank PEFT baselines under matched low-rank budgets, indicating that preserving pretrained structure while reserving capacity for orthogonal corrections is effective for PSWD.

Acknowledgments and Disclosure of Funding

This research was partly supported by the ASTRA Project through the National Research Foundation (NRF) funded by the Ministry of Science and ICT (No. RS-2024-00439619, 90%) and by the National Research Foundation of Korea (RS-2022-NR068758, 10%). The authors declare no competing interests.

References

[1] Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, et al. Learning skillful medium-range global weather forecasting. *Science*, 2023.

[2] Kaifeng Bi, Lingxi Xie, Hengheng Zhang, Xin Chen, Xiaotao Gu, and Qi Tian. Accurate medium-range global weather forecasting with 3D neural networks. *Nature*, 2023.

[3] Simon Lang, Mihai Alexe, Matthew Chantry, Jesper Dramsch, Florian Pinault, Baudouin Raoult, Mariana C. A. Clare, Christian Lessig, Michael Maier-Gerber, Linus Magnusson, et al. AIFS—ECMWF's data-driven forecasting system. *arXiv preprint arXiv:2406.01465*, 2024.

[4] Cristian Bodnar, Wessel P. Bruinsma, Ana Lucic, Megan Stanley, Anna Allen, Johannes Brandstetter, Patrick Garvan, Maik Riechert, Jonathan A. Weyn, Haiyu Dong, et al. A foundation model for the Earth system. *Nature*, 2025.

[5] Thomas Vandal, Evan Kodra, Sangram Ganguly, Andrew Michaelis, Ramakrishna Nemani, and Auroop R. Ganguly. DeepSD: Generating high resolution climate change projections through single image super-resolution. In *KDD*, 2017.

[6] Norihiro Oyama, Noriko N. Ishizaki, Satoshi Koide, and Hiroaki Yoshida. Deep generative model super-resolves spatially correlated multiregional climate data. *Sci. Rep.*, 2023.

[7] Luca Glawion, Julius Polz, Harald Kunstmann, Benjamin Fersch, and Christian Chwala. Global spatio-temporal ERA5 precipitation downscaling to km and sub-hourly scale using generative AI. *npj Clim. Atmos. Sci.*, 2025.

[8] Morteza Mardani, Noah Brenowitz, Yair Cohen, Jaideep Pathak, Chieh-Yu Chen, Cheng-Chin Liu, Arash Vahdat, Mohammad Amin Nabian, Tao Ge, Akshay Subramaniam, et al. Residual corrective diffusion modeling for km-scale atmospheric downscaling. *Commun. Earth Environ.*, 2025.

[9] Ines Würth, Laura Valldecabres, Elliot Simon, Corinna Möhrlen, Bahri Uzunoğlu, Ciaran Gilbert, Gregor Giebel, David Schlipf, and Anton Kaifel. Minute-scale forecasting of wind power—results from the collaborative workshop of IEA Wind Task 32 and 36. *Energies*, 2019.

[10] Suman Ravuri, Karel Lenc, Matthew Willson, Dmitry Kangin, Remi Lam, Piotr Mirowski, Megan Fitzsimons, Maria Athanassiadou, Sheleem Kashem, Sam Madge, et al. Skilful precipitation nowcasting using deep generative models of radar. *Nature*, 2021.

[11] R. D. Sharman, L. B. Cornman, G. Meymaris, J. Pearson, and T. Farrar. Description and derived climatologies of automated in situ eddy-dissipation-rate reports of atmospheric turbulence. *J. Appl. Meteorol. Climatol.*, 2014.

[12] Shilei Cao, Hehai Lin, Jiashun Cheng, Yang Liu, Guowen Li, Xuehe Wang, Juepeng Zheng, Haoyuan Liang, Meng Jin, Chengwei Qin, et al. Task-adaptive parameter-efficient fine-tuning for weather foundation models. In *ICLR*, 2026.

[13] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022.

[14] Fanxu Meng, Zhaohui Wang, and Muhan Zhang. PiSSA: Principal singular values and singular vectors adaptation of large language models. In *NeurIPS*, 2024.

[15] Yifeng Xiong and Xiaohui Xie. OPLoRA: Orthogonal projection LoRA prevents catastrophic forgetting during parameter-efficient fine-tuning. In *AAAI*, 2026.

[16] Stephan Rasp, Stephan Hoyer, Alexander Merose, Ian Langmore, Peter Battaglia, Tyler Russell, Alvaro Sanchez-Gonzalez, Vivian Yang, Rob Carver, Shreya Agrawal, et al. WeatherBench 2: A benchmark for the next generation of data-driven global weather models. *J. Adv. Model. Earth Syst.*, 2024.

[17] Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. DoRA: Weight-decomposed low-rank adaptation. In *ICML*, 2024.

[18] Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *ICLR*, 2023.

[19] Hans Hersbach, Bill Bell, Paul Berrisford, Shoji Hirahara, András Horányi, Joaquín Muñoz-Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Dinand Schepers, et al. The ERA5 global reanalysis. *Q. J. R. Meteorol. Soc.*, 2020.

[20] Martin Ridal, Eric Bazile, Patrick Le Moigne, Roger Randriamampianina, Semjon Schimanke, Ulf Andrae, Lars Berggren, Pierre Brousseau, Per Dahlgren, Lisette Edvinsson, et al. CERRA, the Copernicus European Regional Reanalysis system. *Q. J. R. Meteorol. Soc.*, 2024.

[21] Jaideep Pathak, Shashank Subramanian, Peter Harrington, Sanjeev Raja, Ashesh Chattopadhyay, Morteza Mardani, Thorsten Kurth, David Hall, Zongyi Li, Kamyar Azizzadenesheli, et al. FourCastNet: A global data-driven high-resolution weather model using adaptive Fourier neural operators. *arXiv preprint arXiv:2202.11214*, 2022.

[22] Ryan Keisler. Forecasting global weather with graph neural networks. *arXiv preprint arXiv:2202.07575*, 2022.

[23] Lei Chen, Xiaohui Zhong, Feng Zhang, Yuan Cheng, Yinghui Xu, Yuan Qi, and Hao Li. FuXi: A cascade machine learning forecasting system for 15-day global weather forecast. *npj Clim. Atmos. Sci.*, 2023.

[24] Kang Chen, Tao Han, Fenghua Ling, Junchao Gong, Lei Bai, Xinyu Wang, Jing-Jia Luo, Ben Fei, Wenlong Zhang, Xi Chen, et al. The operational medium-range deterministic weather forecasting can be extended beyond a 10-day lead time. *Commun. Earth Environ.*, 2025.

[25] Tung Nguyen, Rohan Shah, Hritik Bansal, Troy Arcomano, Romit Maulik, Veerabhadra Kotamarthi, Ian Foster, Sandeep Madireddy, and Aditya Grover. Scaling transformer neural networks for skillful and reliable medium-range weather forecasting. In *NeurIPS*, 2024.

[26] 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 *ICML*, 2023.

[27] Dmitrii Kochkov, Janni Yuval, Ian Langmore, Peter Norgaard, Jamie Smith, Griffin Mooers, Milan Klöwer, James Lottes, Stephan Rasp, Peter Düben, et al. Neural general circulation models for weather and climate. *Nature*, 2024.

[28] Ilan Price, Alvaro Sanchez-Gonzalez, Ferran Alet, Tom R. Andersson, Andrew El-Kadi, Dominic Masters, Timo Ewalds, Jacklynn Stott, Shakir Mohamed, Peter Battaglia, et al. Probabilistic weather forecasting with machine learning. *Nature*, 2025.

[29] Carlos Alberto Gomez Gonzalez. DL4DS—deep learning for empirical downscaling. *Environmental Data Sci.*, 2023.

[30] Fang Wang, Di Tian, Lisa Lowe, Latif Kalin, and John Lehrter. Deep learning for daily precipitation and temperature downscaling. *Water Resour. Res.*, 2021.

[31] Neelesh Rampal, Peter B. Gibson, Abha Sood, Stephen Stuart, Nicolas C. Fauchereau, Chris Brandolino, Ben Noll, and Tristan Meyers. High-resolution downscaling with interpretable deep learning: Rainfall extremes over New Zealand. *Weather Clim. Extremes*, 2022.

[32] Lucy Harris, Andrew T. T. McRae, Matthew Chantry, Peter D. Dueben, and Tim N. Palmer. A generative deep learning approach to stochastic downscaling of precipitation forecasts. *J. Adv. Model. Earth Syst.*, 2022.

[33] Ilan Price and Stephan Rasp. Increasing the accuracy and resolution of precipitation forecasts using deep generative models. In *AISTATS*, 2022.

[34] Jussi Leinonen, Daniele Nerini, and Alexis Berne. Stochastic super-resolution for downscaling time-evolving atmospheric fields with a generative adversarial network. *IEEE Trans. Geosci. Remote Sens.*, 2021.

[35] Luca Glawion, Julius Polz, Harald Kunstmann, Benjamin Fersch, and Christian Chwala. spate-GAN: Spatio-temporal downscaling of rainfall fields using a cGAN approach. *Earth Space Sci.*, 2023.

[36] Tung Nguyen, Johannes Brandstetter, Ashish Kapoor, Jayesh K. Gupta, and Aditya Grover. ClimaX: A foundation model for weather and climate. In *ICML*, 2023.

[37] Johannes Schmude, Sujit Roy, Will Trojak, Johannes Jakubik, Daniel Salles Civitarese, Shraddha Singh, Julian Kuehnert, Kumar Ankur, Aman Gupta, Christopher E. Phillips, et al. Prithvi WxC: Foundation model for weather and climate. *arXiv preprint arXiv:2409.13598*, 2024.

[38] Shengchao Chen, Guodong Long, Jing Jiang, Dikai Liu, and Chengqi Zhang. Foundation models for weather and climate data understanding: A comprehensive survey. *arXiv preprint arXiv:2312.03014*, 2023.

[39] Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. Parameter-efficient fine-tuning for large models: A comprehensive survey. *Transactions on Machine Learning Research*, 2024.

[40] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In *NeurIPS*, 2023.

[41] Mojtaba Valipour, Mehdi Rezagholizadeh, Ivan Kobyzev, and Ali Ghodsi. DyLoRA: Parameter-efficient tuning of pre-trained models using dynamic search-free low-rank adaptation. In *EACL*, 2023.

[42] Dawid J. Kopiczko, Tijmen Blankevoort, and Yuki M. Asano. VeRA: Vector-based random matrix adaptation. In *ICLR*, 2024.

[43] Soufiane Hayou, Nikhil Ghosh, and Bin Yu. LoRA+: Efficient low rank adaptation of large models. In *ICML*, 2024.

[44] Yixiao Li, Yifan Yu, Chen Liang, Pengcheng He, Nikos Karampatziakis, Weizhu Chen, and Tuo Zhao. LoftQ: LoRA-fine-tuning-aware quantization for large language models. In *ICLR*, 2024.

[45] Yibo Yang, Xiaojie Li, Zhongzhu Zhou, Shuaiwen L. Song, Jianlong Wu, Liqiang Nie, and Bernard Ghanem. CorDA: Context-oriented decomposition adaptation of large language models for task-aware parameter-efficient fine-tuning. In *NeurIPS*, 2024.

[46] Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. Orthogonal subspace learning for language model continual learning. In *EMNLP Findings*, 2023.

[47] Zhongqi Wang, Jia Dai, Kai Li, Xu Li, and Yanmeng Guo. Complementary subspace low-rank adaptation of vision-language models for few-shot classification. In *ICASSP*, 2026.

[48] Yiding Wang, Fanxu Meng, Xuefeng Zhang, Fan Jiang, Pingzhi Tang, and Muhan Zhang. HD-PiSSA: High-rank distributed orthogonal adaptation. In *EMNLP*, 2025.

[49] Fei Wu, Jia Hu, Geyong Min, and Shiqiang Wang. Efficient orthogonal fine-tuning with principal subspace adaptation. In *ICLR*, 2026.

[50] Lanxin Zhao, Bamdev Mishra, Pratik Jawanpuria, Lequan Lin, Dai Shi, Junbin Gao, and Andi Han. LOFT: Low-rank orthogonal fine-tuning via task-aware support selection. *arXiv preprint arXiv:2605.11872*, 2026.

[51] Lixian Chen and Jianhong Tan. HiP-LoRA: Budgeted spectral plasticity for robust low-rank adaptation. *arXiv preprint arXiv:2604.17751*, 2026.

[52] Guillaume Couairon, Renu Singh, Anastase Charantonis, Christian Lessig, and Claire Monteleoni. ArchesWeatherGen: Skillful and compute-efficient probabilistic weather forecasting with machine learning. *Sci. Adv.*, 2026.

[53] Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL `https://qwen.ai/blog?id=qwen3.5`.

[54] Jinyang Li, Binyuan Hui, Ge Qu, Jiaxi Yang, Binhua Li, Bowen Li, Bailin Wang, Bowen Qin, Ruiying Geng, Nan Huo, et al. Can LLM already serve as a database interface? A BIg bench for large-scale database grounded text-to-SQLs. In *NeurIPS*, 2023.

[55] Mark J. Rodwell, David S. Richardson, Tim D. Hewson, and Thomas Haiden. A new equitable score suitable for verifying precipitation in numerical weather prediction. *Q. J. R. Meteorol. Soc.*, 2010.

[56] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

[57] Noam Shazeer. GLU variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

The appendix is organized into the following sections.

- Appendix A: Additional Analysis of Update Geometry
- Appendix B: Details of Gradient-Guided Scaling of Optimizer Updates
- Appendix C: Detailed Experimental Setup
- Appendix D: Evaluation Metrics
- Appendix E: Implementation Details
- Appendix F: PROLA and Baseline Configurations
- Appendix G: Additional Quantitative Results
- Appendix H: Qualitative Results
- Appendix I: Additional Ablations
- Appendix J: Limitations and Future Work
- Appendix K: Broader Impacts

A Additional Analysis of Update Geometry

This appendix provides the reference subspaces, block definitions, and energy aggregation used in Section 5.3.

Blockwise diagnostic. We compare the final checkpoints of Full FT, PiSSA, and PROLA on the same set $\mathcal{A}$ of 96 attention weight matrices (QKV and output projections). For each $\ell \in \mathcal{A}$, we analyze the effective update

$$\Delta W_\ell^{\text{eff}} = W_\ell^{\text{eff}} - W_{\ell,0}. \tag{14}$$

For this diagnostic, we evaluate the decomposition in Eq. (10) using the same top-r reference subspaces for all methods, with $r = 8$. Let $U_{r,\ell}$ and $V_{r,\ell}$ contain the top-r left and right singular vectors of $W_{\ell,0}$, respectively. The projectors are

$$P_\ell = U_{r,\ell}U_{r,\ell}^\top, \qquad Q_\ell = V_{r,\ell}V_{r,\ell}^\top, \qquad P_{\perp,\ell} = I - P_\ell, \qquad Q_{\perp,\ell} = I - Q_\ell. \tag{15}$$

The principal, cross, and orthogonal blocks are

$$\Delta W_\ell^{\text{prin}} = P_\ell \Delta W_\ell^{\text{eff}} Q_\ell, \tag{16}$$

$$\Delta W_\ell^{\text{cross}} = P_\ell \Delta W_\ell^{\text{eff}} Q_{\perp,\ell} + P_{\perp,\ell} \Delta W_\ell^{\text{eff}} Q_\ell, \tag{17}$$

$$\Delta W_\ell^{\text{orth}} = P_{\perp,\ell} \Delta W_\ell^{\text{eff}} Q_{\perp,\ell}. \tag{18}$$

PROLA constructs its orthogonal branch using the top-r_p directions $U_{p,\ell}, V_{p,\ell}$, with $r_p = 4$:

$$\Delta W_{o,\ell} = s_{o,\ell}(I - U_{p,\ell}U_{p,\ell}^\top)B_{o,\ell}A_{o,\ell}(I - V_{p,\ell}V_{p,\ell}^\top). \tag{19}$$

The branch therefore satisfies

$$U_{p,\ell}^\top \Delta W_{o,\ell} = 0, \qquad \Delta W_{o,\ell} V_{p,\ell} = 0. \tag{20}$$

These identities concern the top-r_p construction subspaces. Under the top-r diagnostic partition, this branch may also contribute to the principal and cross blocks. The freely trainable principal branch may contribute to the orthogonal block. Figure 5 measures blocks of the combined effective update.

Normalized block energy. For each block $b \in \{\text{prin}, \text{cross}, \text{orth}\}$, we sum its squared Frobenius norm across adapted matrices and normalize by the total update energy:

$$\bar{e}_b = \frac{\sum_{\ell \in \mathcal{A}} \|\Delta W_\ell^b\|_F^2}{\sum_{\ell \in \mathcal{A}} \|\Delta W_\ell^{\text{eff}}\|_F^2}. \tag{21}$$

B Details of Gradient-Guided Scaling of Optimizer Updates

This appendix provides additional details on the gradient-guided scaling of optimizer updates introduced in Section 3.2. The scaling is applied to the trainable low-rank factors of each adapted layer $\ell \in \mathcal{A}$. Let B_ℓ^g and A_ℓ^g denote the low-rank factors of branch $g \in \{p, o\}$ at layer ℓ, where p and o denote the principal and orthogonal branches, respectively. The k-th rank-1 component is represented by the factor pair $(\mathbf{b}_{\ell,k}^g, \mathbf{a}_{\ell,k}^g)$, where $\mathbf{b}_{\ell,k}^g$ is the k-th column of B_ℓ^g and $(\mathbf{a}_{\ell,k}^g)^\top$ is the k-th row of A_ℓ^g. For the orthogonal branch, this component is defined before applying its fixed projectors, as specified in Appendix F.

The component signal and branch signal are

$$u_{\ell,k}^g = \left\| \nabla_{\mathbf{b}_{\ell,k}^g} \mathcal{L}_{ST} \right\|_2 + \left\| \nabla_{\mathbf{a}_{\ell,k}^g} \mathcal{L}_{ST} \right\|_2, \qquad \mu_\ell^g = \frac{1}{r_g} \sum_{k=1}^{r_g} u_{\ell,k}^g. \tag{22}$$

They are normalized into the component score and branch score:

$$\pi_{\ell,k}^g = \frac{u_{\ell,k}^g + \epsilon}{\sum_{j=1}^{r_g} (u_{\ell,j}^g + \epsilon)}, \qquad \rho_\ell^g = \frac{\mu_\ell^g + \epsilon}{\sum_{g' \in \{p,o\}} (\mu_\ell^{g'} + \epsilon)}. \tag{23}$$

The rank-1 update scale is

$$\alpha_{\ell,k}^g = r \times \mathrm{sg}\left(\rho_\ell^g \pi_{\ell,k}^g \right), \qquad r = r_p + r_o, \tag{24}$$

where $\epsilon > 0$ is a numerical constant and $\mathrm{sg}(\cdot)$ denotes stop-gradient. The stop-gradient operation treats the scores as measurements of the current training signal and prevents gradients from being propagated through the score computation. For single-branch variants, the branch sums and normalizations are restricted to the existing branch, for which $\rho_\ell^g = 1$.

We apply $\alpha_{\ell,k}^g$ after the base optimizer has formed its update. Let $\widetilde{\Delta}\mathbf{b}_{\ell,k}^g$ and $\widetilde{\Delta}\mathbf{a}_{\ell,k}^g$ denote the optimizer updates before scaling. The applied updates are

$$\Delta\mathbf{b}_{\ell,k}^g = \alpha_{\ell,k}^g \widetilde{\Delta}\mathbf{b}_{\ell,k}^g, \qquad \Delta\mathbf{a}_{\ell,k}^g = \alpha_{\ell,k}^g \widetilde{\Delta}\mathbf{a}_{\ell,k}^g. \tag{25}$$

Thus, gradient-guided scaling rescales the applied parameter updates after the base optimizer computes them, without changing the base optimizer's raw-update computation.

Mean-preserving property. Since $\sum_{k=1}^{r_g} \pi_{\ell,k}^g = 1$ and $\sum_{g \in \{p,o\}} \rho_\ell^g = 1$, we have

$$\frac{1}{r} \sum_{g \in \{p,o\}} \sum_{k=1}^{r_g} \alpha_{\ell,k}^g = \frac{1}{r} \sum_{g \in \{p,o\}} \sum_{k=1}^{r_g} r \rho_\ell^g \pi_{\ell,k}^g = 1. \tag{26}$$

Therefore, the multiplicative scales have unit mean across the r rank-1 components, while their relative values redistribute update strength.

For the scaling variants in Table 5, ρ-only scaling uses uniform component allocation within each branch, giving $\alpha_{\ell,k}^g = (r/r_g)\,\mathrm{sg}(\rho_\ell^g)$. The π-only variant fixes branch allocation at r_g/r, giving $\alpha_{\ell,k}^g = r_g\,\mathrm{sg}(\pi_{\ell,k}^g)$. Both variants retain the two branches with $r_p = r_o = 4$; disabling scaling sets all $\alpha_{\ell,k}^g$ to one.

Component concentration diagnostic. We examine how gradient-guided scaling distributes optimizer updates across rank-1 components within each layer. Using the mean-preserving property, we define the normalized update-scale distribution as

$$\omega_{\ell,k}^g = \frac{\alpha_{\ell,k}^g}{r}. \tag{27}$$

The normalized entropy is

$$H_\ell = -\frac{1}{\log r} \sum_{g \in \{p,o\}} \sum_{k=1}^{r_g} \omega_{\ell,k}^g \log \omega_{\ell,k}^g. \tag{28}$$

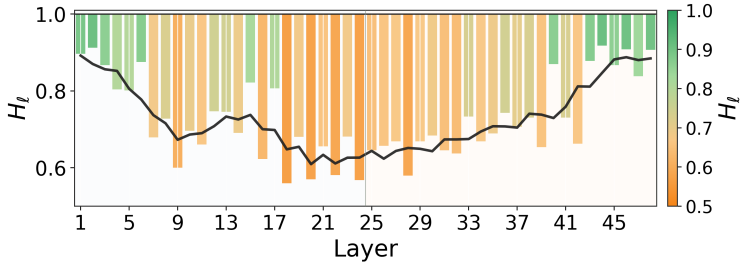


Figure 6: **Layer-wise concentration of rank-1 update scales.** Each bar extends from uniform scaling ($H_\ell = 1$) to the mean normalized entropy of one attention block. Blocks are ordered from encoder to decoder.

At a fixed trained checkpoint, we compute H_ℓ separately for each validation batch and average over batches. Within each attention block, we then average the entropies of the QKV and output projections.

Figure 6 shows that mean entropy falls below the uniform value of 1 in every block, with the largest departures in the middle layers. Even the lowest block average, approximately 0.56, remains well above the single-component limit of 0. These averages indicate that scaling departs from uniform allocation while remaining distributed across multiple rank-1 components, including in the middle layers.

C Detailed Experimental Setup

For the main global PSWD experiments, we use Aurora [4] as the pretrained weather backbone, starting from the same pretrained checkpoint[2] for all PEFT variants. The experiments use the WeatherBench 2 dataset[3] [16] at two resolutions: $5.625°$ data available at 6-hour intervals for low-resolution inputs, and $1.5°$ data available at 1-hour intervals for high-resolution targets. The input lags are $\mathcal{H}_c = \{0, \delta_c\}$ with $\delta_c = 6\text{h}$. The target leads are $\mathcal{H}_f = \{1\text{h}, \ldots, 6\text{h}\}$, with $\delta_f = 1\text{h}$ and $h_{\text{end}} = 6\text{h}$.

The $5.625°$ inputs use a 32×64 grid. We crop the raw 121×240 target grid to 120×240 to accommodate the backbone's patch size of 4. We train on 2007-2016, validate on 2017, and test on 2018. We use 68 weather variables in total. The surface variables are $T2M$, $U10$, and $V10$, and the atmospheric variables are geopotential Z, temperature T, specific humidity Q, 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. This gives $5 \times 13 = 65$ atmospheric channels and 3 surface channels.

ERA5-to-CERRA downscaling. We additionally use $5.625°$ ERA5 inputs to predict CERRA fields over a 128×128 crop of Central Europe at approximately 5.5 km spacing [19, 20]. All months of 2012–2016 are used for training, 2017 for validation, and 2018 for testing. Inputs at $t - \delta_c$ and t, with $\delta_c = 6\text{h}$, are paired with targets at $t + h$ for $h \in \mathcal{H}_f = \{3\text{h}, 6\text{h}\}$ ($\delta_f = 3\text{h}$). All required times lie within the corresponding split. ERA5 inputs are bilinearly interpolated to the geographic coordinates of the CERRA crop. The 13 channels comprise $T2M$, $U10$, and $V10$, together with Z, T, Q, U, V at 500 and 850 hPa. Wind fields are converted to geographic components where needed, and relative humidity is converted to specific humidity.

Regional precipitation forecasting. We evaluate regional precipitation forecasting using a $1.5°$, 40×40 ERA5 grid covering $0°$–$58.5°$N and $75°$–$133.5°$E. We train on 2007–2016, validate on 2017, and test on 2018, keeping input and target frames within each split. Inputs at $t - 12\text{h}$ and t are used to predict 6-hour accumulated precipitation at 12h, 24h, and 36h lead times. Each target represents the accumulation over the six hours ending at its valid time. Following the regional precipitation protocol [12], the inputs include surface temperature, horizontal winds, mean sea-level pressure, and precipitation, together with upper-air temperature, horizontal winds, relative humidity, and geopotential at the 13 pressure levels listed above.

[2]Copyright (c) 2024 Microsoft Corporation. Licensed under MIT license
[3]Copyright (c) 2023 Google LLC. Licensed under Apache-2.0 license

ArchesWeather backbone. We also evaluate the scaffold-and-correct pipeline with pretrained ArchesWeather-M [52] at the same 5.625° input and 1.5° target resolutions as the main benchmark. To match the backbone's forecasting interval, the inputs are at $t - \delta_c$ and t with $\delta_c = 24\text{h}$. The target leads are $\mathcal{H}_f = \{6\text{h}, 12\text{h}, 18\text{h}, 24\text{h}\}$, with $\delta_f = 6\text{h}$ and $h_{\text{end}} = 24\text{h}$. We train on 2007–2016 and evaluate on 2018.

Additional resolution gap. We evaluate an additional 5.625°-to-2.8125° setting, with a 64×128 target grid obtained by conservative remapping of the fine-resolution reference fields. Inputs are at $t - \delta_c$ and t with $\delta_c = 6\text{h}$. The target leads are $\mathcal{H}_f = \{3\text{h}, 6\text{h}\}$ with $\delta_f = 3\text{h}$.

Text-to-SQL adaptation. We evaluate Qwen3.5-9B [53] on BIRD [54] to assess transfer to schema-conditioned SQL generation. We use database-disjoint training, validation, and evaluation splits, with the corrected development set used for evaluation.

D Evaluation Metrics

For the main global PSWD benchmark, we report latitude-weighted root mean squared error (RMSE) and anomaly correlation coefficient (ACC). Let v denote a reported variable, $h \in \mathcal{H}_f$ a target lead time, and t_n the anchor time of sample n. At latitude index i and longitude index j, $\hat{y}^{(v,h)}_{n,i,j}$ and $y^{(v,h)}_{n,i,j}$ denote the denormalized values of variable v in $\hat{\mathbf{y}}^{ST}_{t_n+h}$ and $\mathbf{y}^{\text{HR}}_{t_n+h}$, respectively. Here $(H, W) = (H_{\text{HR}}, W_{\text{HR}})$. We use latitude weights

$$w_i = \frac{\cos(\varphi_i)}{\frac{1}{H}\sum_{i'=1}^{H} \cos(\varphi_{i'})}, \tag{29}$$

where φ_i is the latitude at row i. Following [12], we report the mean of sample-wise latitude-weighted RMSE values:

$$\text{RMSE}_{v,h} = \frac{1}{N}\sum_{n=1}^{N} \sqrt{\frac{1}{HW}\sum_{i=1}^{H}\sum_{j=1}^{W} w_i \left(\hat{y}^{(v,h)}_{n,i,j} - y^{(v,h)}_{n,i,j}\right)^2}. \tag{30}$$

For ACC, we subtract a temporal mean field from both predictions and targets. Let $c^{(v)}_{i,j}$ denote the temporal mean of the denormalized high-resolution ground-truth fields for variable v:

$$c^{(v)}_{i,j} = \frac{1}{|\mathcal{T}_{\text{clim}}|}\sum_{\tau \in \mathcal{T}_{\text{clim}}} y^{(v)}_{\tau,i,j}, \tag{31}$$

where $\mathcal{T}_{\text{clim}}$ contains the training-period target valid times from 2007–2016. We compute sample-wise anomaly correlations and then average over samples:

$$\text{ACC}_{v,h} = \frac{1}{N}\sum_{n=1}^{N} \frac{\sum_{i=1}^{H}\sum_{j=1}^{W} w_i \left(\hat{y}^{(v,h)}_{n,i,j} - c^{(v)}_{i,j}\right)\left(y^{(v,h)}_{n,i,j} - c^{(v)}_{i,j}\right)}{\max\left\{\sqrt{\sum_{i=1}^{H}\sum_{j=1}^{W} w_i \left(\hat{y}^{(v,h)}_{n,i,j} - c^{(v)}_{i,j}\right)^2}\sqrt{\sum_{i=1}^{H}\sum_{j=1}^{W} w_i \left(y^{(v,h)}_{n,i,j} - c^{(v)}_{i,j}\right)^2}, \epsilon\right\}}. \tag{32}$$

Here, $\epsilon = 10^{-8}$. Weather fields are evaluated after denormalization: RMSE is in the corresponding physical units, and ACC is dimensionless. For CERRA, we use the same sample-wise RMSE aggregation on the regional target grid.

Seasonal and spatial-tail evaluation. Seasonal results group test cases into DJF, MAM, JJA, and SON according to the target valid time. For spatial-tail evaluation, we identify the upper and lower 1% of grid points within each target field using its 99th and 1st percentiles. Tail RMSE uses latitude weighting over the target-selected grid points. For the Brier skill score (BSS), forecast and observed event indicators use the same target-derived threshold, and the reference assigns event probability 0.01 to every grid point. We report $1 - \text{BS}/\text{BS}_{\text{ref}}$, with latitude weighting for both scores.

Regional precipitation metrics. We report RMSE, ACC, and the Stable Equitable Error in Probability Space (SEEPS) [55] for 6-hour accumulated precipitation at each evaluated lead time. Precipitation RMSE is the square root of the latitude-weighted mean squared error over all evaluated times and grid points, and is reported in cm. ACC is averaged over samples using a training-period climatology

indexed by day of year and hour of day. SEEPS divides precipitation into dry, light, and heavy categories using a dry threshold of 0.1 mm and the local two-thirds quantile of wet precipitation. We report latitude-weighted mean SEEPS over locations with climatological dry fractions between 0.1 and 0.85. Lower SEEPS indicates better performance.

Spherical harmonic spectra. Following spectral evaluation in global weather forecasting [28, 52], we decompose each predicted and target field into orthonormal spherical harmonics. For fixed v and h, write $\hat{y}_n$ and y_n for the predicted and target fields of sample n at valid time $t_n + h$. For a field y_n with coefficients $a_{n,\ell,m}$, the power at degree ℓ is

$$\mathcal{P}_\ell(y_n) = \frac{1}{4\pi} \sum_{m=-\ell}^{\ell} |a_{n,\ell,m}|^2. \tag{33}$$

We compute power separately for each case and plot the ratio of mean prediction power to mean target power,

$$\frac{N^{-1} \sum_{n=1}^{N} \mathcal{P}_\ell(\hat{y}_n)}{N^{-1} \sum_{n=1}^{N} \mathcal{P}_\ell(y_n)}. \tag{34}$$

A ratio of one denotes matching power at that spatial scale. Figure 3 uses T2M at 6h from the seed-0 models over all 1,458 test cases in 2018. We evaluate degrees $\ell = 1, \ldots, 59$ on the native 120×240 grid and convert degree to wavelength as $2\pi R/\sqrt{\ell(\ell+1)}$, with Earth's radius $R = 6371$ km.

Text-to-SQL evaluation. For BIRD [54], we report execution accuracy (EX), the percentage of predicted SQL queries whose execution results match those of the reference queries on the corresponding databases. We compute EX using a custom SQLite evaluator with unordered result-set comparison.

Aggregation across seeds. For the main comparisons, we compute each test metric separately for seeds 0, 1, and 2, and report the mean and sample standard deviation across these three values. The standard deviation uses denominator $n - 1$.

E Implementation Details

The main Aurora training configuration uses two NVIDIA RTX 3090 GPUs. For PEFT and Head only experiments, we use per-GPU batch size 4 with gradient accumulation over 4 steps, giving an effective global batch size of 32. For Full FT, we use per-GPU batch size 2 with gradient accumulation over 8 steps, giving the same effective global batch size of 32. Models in this benchmark are trained for 10 epochs. We use AdamW [56] with $(\beta_1, \beta_2) = (0.9, 0.999)$, $\epsilon = 10^{-8}$, and weight decay 0.05. The learning rate follows a step-level cosine schedule with linear warmup. For PEFT and Head only experiments, the base learning rate is 5×10^{-4}, the warmup learning rate is 10^{-6}, the minimum learning rate is 10^{-6}, and warmup lasts for one epoch. For Full FT, the backbone learning rate is 5×10^{-5}, with warmup and minimum learning rates 10^{-7}. The correction heads in Full FT use the learning rate 5×10^{-4}. Gradient norm clipping is applied with a maximum norm of 1.0. The main comparisons use seeds 0, 1, and 2, with the same seed set for every method. We evaluate the final checkpoint after the prescribed training budget.

Training objective. For the main Aurora benchmark, we use a weighted L1 loss on fields normalized with Aurora's variable-specific statistics [4]. At each lead time, the surface term sums the variable-weighted spatial mean absolute errors and is multiplied by $1/4$. The upper-air term averages spatial mean absolute errors over pressure levels and then takes a weighted sum across variables. The combined loss is divided by the number of variable types, $3 + 5 = 8$. The surface weights for $(T2M, U10, V10)$ are $(3.0, 0.77, 0.66)$, and the upper-air weights for (Z, T, Q, U, V) are $(3.0, 1.5, 0.6, 0.77, 0.54)$, shared across pressure levels. We average $\mathcal{L}_{ST}$ in Eq. (5) over training examples. Its six lead terms supervise the final predictions $\hat{\mathbf{y}}^{ST}_{t+h}$ for $h \in \mathcal{H}_f$. The spatial endpoint $\hat{\mathbf{y}}^{S}_{t+h_{\text{end}}}$ receives supervision through the temporal stage.

Regional weather experiments. ERA5-to-CERRA adaptation uses 10 epochs, an effective global batch size of 8, and a learning rate of 5×10^{-4}. Regional precipitation forecasting uses 10 epochs, an effective global batch size of 32, and a learning rate of 3.75×10^{-4}. Both settings use AdamW with $(\beta_1, \beta_2) = (0.9, 0.999)$, weight decay 0.05, and a cosine learning-rate schedule with one epoch of linear warmup. These comparisons use seed 0 and the final checkpoint after 10 epochs. Within

each task, LoRA, PiSSA, OPLoRA, and PROLA adapt the same attention projections and train the same task-specific heads.

ArchesWeather experiments. We fine-tune the pretrained ArchesWeather-M backbone [52] for one epoch, corresponding to 7,300 optimizer steps, with an effective global batch size of 2 and a learning rate of 3×10^{-4}. We use AdamW with $(\beta_1, \beta_2) = (0.9, 0.98)$, weight decay 0.05, and 250 warmup steps followed by cosine decay. All compared methods adapt the attention projections and train the correction heads, using seed 0 and the final checkpoint.

Text-to-SQL adaptation. We compare the unadapted model with LoRA, PiSSA, OPLoRA, and PROLA. All adapters target attention and MLP projections with rank 16 and 40.1M trainable parameters; PROLA uses $r_p = r_o = 8$. Each adapter is trained for one epoch with a learning rate of 2×10^{-5} and seed 42.

F PROLA and Baseline Configurations

For the main Aurora comparison, PROLA and all LoRA-style PEFT baselines fine-tune the linear projections in the backbone's self-attention operations. All other pretrained backbone parameters remain frozen unless the method is Full FT. The spatial correction head C_s and temporal correction head C_t are trained in all methods in this comparison except where explicitly ablated. Thus, PEFT baseline comparisons use the same prediction pipeline and differ only in the backbone adaptation strategy.

The default main comparison uses rank $r = 8$ for all fixed-rank PEFT baselines. LoRA [13] uses the standard zero-output initialization. Unless otherwise stated, all adapters use no adapter dropout and do not train bias terms. DoRA [17] uses the same rank, target modules, dropout, and bias setting as LoRA, with the additional magnitude parameter trained following the original parameterization. PiSSA [14] initializes the low-rank factors from the top-r singular components of each pretrained weight and residualizes the frozen base weight to preserve the pretrained function at initialization. OPLoRA [15] uses fixed projectors onto the complements of the top-8 left and right singular subspaces of each pretrained weight, with the same trainable rank, target modules, dropout, and bias setting as LoRA.

For PROLA, we use the same total low-rank budget $r = 8$, split evenly into $r_p = 4$ for the principal branch and $r_o = 4$ for the orthogonal branch. The reference singular directions and orthogonal projectors are computed from the pretrained weights and kept fixed during adaptation, while the principal factors A_p, B_p remain trainable. We residualize the frozen base weight as in Eq. (9), set $B_{o,0} = 0$, and sample the entries of $A_{o,0}$ from $\mathcal{N}(0, \sigma_o^2)$ with $\sigma_o = 0.02$, preserving the pretrained function. The complete PROLA configuration uses gradient-guided update scaling (Appendix B). With the principal branch removed ($r_p = 0$, $r_o = 8$), the orthogonal-only ablation retains fixed projectors onto the complements of each pretrained weight's top-8 left and right singular subspaces. We set $s_g = \alpha / r_g$ for each PROLA branch $g \in \{p, o\}$ and $s = \alpha / r$ for LoRA, DoRA, PiSSA, and OPLoRA, with $\alpha = 1$ for all five methods.

AdaLoRA [18] uses $\alpha = 16$, $r_{\text{init}} = 12$, $r_{\text{target}} = 8$, mask updates every 10 optimizer steps, importance EMA coefficients $\beta_1 = \beta_2 = 0.85$, and orthogonal regularization weight 10^{-3}. Initial and final warmups last one and two epochs of optimizer steps, respectively. Its target modules, dropout, and bias settings match the fixed-rank baselines.

Head only trains C_s and C_t with the backbone frozen; Full FT also trains the backbone, using a smaller learning rate for its pretrained weights. Table 6 reports the trainable-parameter counts for the main Aurora configuration. For PROLA, these comprise $2{,}850{,}816$ backbone PEFT parameters (ϕ_{PEFT}), $831{,}856$ temporal correction-head parameters (ϕ_t), and $236{,}828$ spatial correction-head parameters (ϕ_s).

Correction heads and numerical scaffolds. The correction heads C_s and C_t use separate surface and upper-air branches with hidden widths 32 and 512, respectively. Both use a single depthwise 3×3 convolution followed by pointwise SwiGLU [57] projection, zero-initialized output projections, and no dropout. The spatial head C_s predicts a terminal residual from the backbone forecast and endpoint conditioning fields. The temporal head C_t combines endpoint-conditioned residual basis maps using lead-dependent softmax gates. It uses 4 basis maps for surface variables and 12 for upper-air variables, with gate hidden widths 64 and 256 and a 16-dimensional lead-time embedding.

Table 6: Trainable parameter counts for the main Aurora configuration. Counts include backbone adaptation and correction-head parameters.

Method	Trainable parameters
Head only	1,068,684
LoRA	3,919,500
DoRA	4,157,068
AdaLoRA	5,346,060
PiSSA	3,919,500
OPLoRA	3,919,500
PROLA	3,919,500
Full FT	1,257,336,076

The fixed spatial operator U_s is bilinear interpolation. It upsamples both surface fields and upper-air fields frame-wise from the 32×64 low-resolution grid to the 120×240 target grid.

The temporal scaffold in Eq. (4) uses linear interpolation:

$$\tilde{\mathbf{y}}_{t+h} = U_t\big(\tilde{\mathbf{x}}_t, \hat{\mathbf{y}}^S_{t+h_{\mathrm{end}}}; h\big) = \tilde{\mathbf{x}}_t + \frac{h}{h_{\mathrm{end}}}\left(\hat{\mathbf{y}}^S_{t+h_{\mathrm{end}}} - \tilde{\mathbf{x}}_t\right), \qquad h \in \mathcal{H}_f. \tag{35}$$

For the main Aurora benchmark, $h_{\mathrm{end}} = 6\mathrm{h}$ and $\delta_f = 1\mathrm{h}$. This operator provides a non-learnable temporal scaffold, while C_t learns the residual sub-step deviations that the linear scaffold cannot capture.

Baseline fairness. The PEFT methods in Table 1 share the scaffold-and-correct pipeline, correction heads, and adapted attention projections. The adapted modules are the QKV and output projections in 48 Aurora attention blocks (encoder depths $(6, 10, 8)$ and decoder depths $(8, 10, 6)$). All non-adapted pretrained backbone parameters are frozen for PEFT methods.

G Additional Quantitative Results

Full main results. Tables 7 and 8 report complete per-lead RMSE and ACC results over the target lead times 1h–6h, respectively. Each entry reports the mean and sample standard deviation over seeds 0, 1, and 2, following the latitude-weighted evaluation protocol in Appendix D. These results extend the representative 3h and 6h comparisons to the full forecast trajectory.

Regional weather prediction. Table 9 provides the full results for the regional tasks in Section 4.3. For ERA5-to-CERRA downscaling, PROLA achieves the lowest RMSE across all five reported variables at both 3h and 6h. For regional precipitation forecasting, it achieves the lowest SEEPS and RMSE and the highest ACC at 12h, 24h, and 36h. These comparisons use seed 0 and the 10-epoch training settings described in Appendices C and E.

Transfer of the adaptation design. We adapt Qwen3.5-9B [53] to BIRD [54] to test whether reserving capacity for orthogonal corrections alongside the principal branch also benefits schema-conditioned query generation. Table 10 shows that at rank 16, PROLA achieves 52.87% execution accuracy, exceeding PiSSA and LoRA by 0.91 and 3.33 percentage points, respectively. See Appendices C, D, and E for the data split, evaluation metric, and training settings.

Transfer to ArchesWeather. Table 11 evaluates the same adaptation methods with the ArchesWeather-M backbone at 6h, 12h, 18h, and 24h. PROLA uses $r_p = r_o = 4$, and all methods use the same attention projections and correction heads. PROLA achieves the lowest Z500 and T2M RMSE at all four lead times, extending the comparison beyond the Aurora backbone. The one-epoch training configuration is given in Appendix E.

Additional resolution gap. Table 12 compares the four PEFT methods for 5.625°-to-2.8125° downscaling with targets at 3h and 6h. PROLA achieves the lowest Z500 and T2M RMSE at both lead times.

Seasonal and spatial-tail robustness. Tables 13 and 14 evaluate Z500 forecasts at 6h across seasons and within the upper and lower spatial tails of the target fields. PROLA has the lowest RMSE in

Table 7: **Full RMSE (↓) results on WeatherBench 2 [16].** We report latitude-weighted RMSE at all lead times from 1h to 6h as mean ± sample standard deviation over three seeds (0, 1, and 2). Best displayed mean results among PEFT methods are boldfaced, including ties after rounding.

Method	Trainable Params	Variable	1h	2h	3h	4h	5h	6h
Full FT	1.26B	Z500	48.22 ± 0.67	49.44 ± 1.39	60.13 ± 1.61	72.88 ± 1.82	85.63 ± 2.27	106.68 ± 2.63
		T850	0.903 ± 0.011	0.834 ± 0.004	0.813 ± 0.002	0.874 ± 0.002	0.966 ± 0.003	1.114 ± 0.008
		T2M	1.472 ± 0.003	1.225 ± 0.004	1.064 ± 0.009	0.955 ± 0.015	1.044 ± 0.019	1.323 ± 0.022
		U10	1.239 ± 0.013	1.052 ± 0.008	0.902 ± 0.004	0.873 ± 0.003	0.941 ± 0.007	1.135 ± 0.012
		V10	1.310 ± 0.024	1.094 ± 0.017	0.925 ± 0.009	0.894 ± 0.003	0.984 ± 0.003	1.209 ± 0.010
Head only	1.07M	Z500	68.16 ± 0.72	78.45 ± 0.40	94.93 ± 0.49	112.98 ± 0.68	128.19 ± 0.97	144.03 ± 1.47
		T850	1.016 ± 0.004	1.013 ± 0.006	1.049 ± 0.006	1.135 ± 0.005	1.218 ± 0.005	1.315 ± 0.008
		T2M	1.704 ± 0.006	1.931 ± 0.007	2.083 ± 0.009	2.297 ± 0.012	2.553 ± 0.017	2.656 ± 0.021
		U10	1.540 ± 0.014	1.578 ± 0.016	1.648 ± 0.018	1.752 ± 0.020	1.847 ± 0.022	1.947 ± 0.024
		V10	1.423 ± 0.022	1.476 ± 0.019	1.563 ± 0.018	1.685 ± 0.017	1.799 ± 0.018	1.921 ± 0.018
LoRA	3.92M	Z500	55.67 ± 0.42	51.19 ± 0.44	54.72 ± 0.36	60.22 ± 0.29	64.76 ± 0.19	79.17 ± 0.33
		T850	0.943 ± 0.004	0.913 ± 0.003	0.898 ± 0.002	0.923 ± 0.001	0.944 ± 0.001	0.994 ± 0.003
		T2M	1.399 ± 0.022	1.266 ± 0.009	1.199 ± 0.007	1.140 ± 0.008	1.182 ± 0.010	1.280 ± 0.012
		U10	1.294 ± 0.010	1.248 ± 0.009	1.219 ± 0.008	1.236 ± 0.005	1.255 ± 0.005	1.314 ± 0.005
		V10	1.221 ± 0.006	1.191 ± 0.005	1.181 ± 0.002	1.214 ± 0.001	1.242 ± 0.001	1.306 ± 0.002
DoRA	4.16M	Z500	55.45 ± 0.46	50.83 ± 0.43	54.32 ± 0.35	59.73 ± 0.31	64.10 ± 0.20	78.40 ± 0.30
		T850	0.941 ± 0.003	0.909 ± 0.003	0.893 ± 0.002	0.918 ± 0.001	0.939 ± 0.001	0.989 ± 0.003
		T2M	1.392 ± 0.022	1.258 ± 0.009	1.191 ± 0.006	1.131 ± 0.007	1.174 ± 0.010	1.272 ± 0.013
		U10	1.286 ± 0.010	1.239 ± 0.009	1.209 ± 0.008	1.226 ± 0.005	1.245 ± 0.005	1.306 ± 0.006
		V10	1.215 ± 0.005	1.184 ± 0.003	1.173 ± 0.001	1.205 ± 0.002	1.233 ± 0.001	1.298 ± 0.001
AdaLoRA	5.35M	Z500	54.37 ± 0.47	49.74 ± 0.21	53.40 ± 0.27	58.93 ± 0.32	63.20 ± 0.11	77.51 ± 0.94
		T850	0.934 ± 0.011	0.901 ± 0.008	0.884 ± 0.006	0.910 ± 0.004	0.931 ± 0.005	0.983 ± 0.006
		T2M	1.373 ± 0.010	1.241 ± 0.002	1.176 ± 0.005	1.115 ± 0.005	1.152 ± 0.002	1.251 ± 0.002
		U10	1.256 ± 0.009	1.206 ± 0.006	1.175 ± 0.005	1.193 ± 0.005	1.216 ± 0.005	1.282 ± 0.004
		V10	1.213 ± 0.009	1.175 ± 0.006	1.158 ± 0.002	1.187 ± 0.001	1.214 ± 0.001	1.279 ± 0.002
PiSSA	3.92M	Z500	55.65 ± 0.28	51.28 ± 0.95	54.55 ± 0.89	59.60 ± 0.57	63.79 ± 0.44	78.43 ± 0.93
		T850	0.942 ± 0.003	0.910 ± 0.002	0.893 ± 0.002	0.918 ± 0.000	0.938 ± 0.002	0.989 ± 0.004
		T2M	1.382 ± 0.020	1.261 ± 0.008	1.198 ± 0.005	1.131 ± 0.005	1.158 ± 0.007	1.251 ± 0.009
		U10	1.291 ± 0.006	1.245 ± 0.005	1.215 ± 0.005	1.228 ± 0.003	1.243 ± 0.003	1.298 ± 0.004
		V10	1.218 ± 0.009	1.187 ± 0.007	1.175 ± 0.003	1.204 ± 0.001	1.228 ± 0.001	1.289 ± 0.003
OPLoRA	3.92M	Z500	55.77 ± 0.42	51.37 ± 0.43	54.94 ± 0.39	60.48 ± 0.33	65.11 ± 0.23	79.60 ± 0.38
		T850	0.944 ± 0.004	0.914 ± 0.003	0.899 ± 0.003	0.925 ± 0.001	0.946 ± 0.001	0.996 ± 0.003
		T2M	1.400 ± 0.026	1.270 ± 0.011	1.204 ± 0.007	1.145 ± 0.006	1.184 ± 0.010	1.281 ± 0.012
		U10	1.297 ± 0.010	1.251 ± 0.008	1.223 ± 0.007	1.240 ± 0.005	1.259 ± 0.004	1.319 ± 0.005
		V10	1.224 ± 0.008	1.194 ± 0.006	1.185 ± 0.003	1.217 ± 0.001	1.246 ± 0.001	1.311 ± 0.002
PROLA (Ours)	3.92M	Z500	**52.98** ± 0.22	**47.49** ± 0.49	**50.58** ± 0.42	**55.44** ± 0.22	**58.62** ± 0.15	**72.35** ± 0.64
		T850	**0.927** ± 0.006	**0.887** ± 0.004	**0.864** ± 0.003	**0.885** ± 0.002	**0.904** ± 0.002	**0.956** ± 0.002
		T2M	**1.372** ± 0.022	**1.205** ± 0.004	**1.129** ± 0.002	**1.062** ± 0.002	**1.109** ± 0.006	**1.212** ± 0.006
		U10	**1.236** ± 0.009	**1.181** ± 0.006	**1.143** ± 0.003	**1.156** ± 0.002	**1.175** ± 0.002	**1.240** ± 0.005
		V10	**1.178** ± 0.004	**1.137** ± 0.002	**1.120** ± 0.001	**1.148** ± 0.001	**1.171** ± 0.001	**1.234** ± 0.002

Table 8: **Full ACC (↑) results on WeatherBench 2 [16].** We report latitude-weighted ACC at all lead times from 1h to 6h as mean ± sample standard deviation over three seeds (0, 1, and 2). Best displayed mean results among PEFT methods are boldfaced, including ties after rounding.

Method	Trainable Params	Variable	1h	2h	3h	4h	5h	6h
Full FT	1.26B	Z500	0.999 ± 0.000	0.999 ± 0.000	0.998 ± 0.000	0.997 ± 0.000	0.996 ± 0.000	0.995 ± 0.000
		T850	0.984 ± 0.000	0.986 ± 0.000	0.987 ± 0.000	0.985 ± 0.000	0.981 ± 0.000	0.976 ± 0.000
		T2M	0.958 ± 0.001	0.972 ± 0.000	0.979 ± 0.000	0.983 ± 0.000	0.980 ± 0.001	0.970 ± 0.001
		U10	0.952 ± 0.001	0.966 ± 0.000	0.975 ± 0.000	0.976 ± 0.000	0.973 ± 0.000	0.961 ± 0.001
		V10	0.947 ± 0.002	0.964 ± 0.001	0.974 ± 0.001	0.976 ± 0.000	0.971 ± 0.000	0.956 ± 0.001
Head only	1.07M	Z500	0.998 ± 0.000	0.997 ± 0.000	0.996 ± 0.000	0.994 ± 0.000	0.992 ± 0.000	0.990 ± 0.000
		T850	0.979 ± 0.000	0.979 ± 0.000	0.978 ± 0.000	0.974 ± 0.000	0.970 ± 0.000	0.965 ± 0.001
		T2M	0.944 ± 0.001	0.927 ± 0.001	0.915 ± 0.001	0.894 ± 0.002	0.868 ± 0.001	0.856 ± 0.001
		U10	0.925 ± 0.001	0.922 ± 0.001	0.914 ± 0.001	0.902 ± 0.002	0.890 ± 0.002	0.877 ± 0.002
		V10	0.938 ± 0.002	0.933 ± 0.002	0.924 ± 0.002	0.910 ± 0.002	0.898 ± 0.003	0.883 ± 0.003
LoRA	3.92M	Z500	0.998 ± 0.000	**0.999** ± 0.000	**0.999** ± 0.000	0.998 ± 0.000	**0.998** ± 0.000	**0.997** ± 0.000
		T850	0.982 ± 0.000	0.983 ± 0.000	0.984 ± 0.000	0.983 ± 0.000	0.982 ± 0.000	0.980 ± 0.000
		T2M	0.963 ± 0.001	0.970 ± 0.000	0.973 ± 0.000	0.976 ± 0.000	0.974 ± 0.000	0.969 ± 0.001
		U10	0.948 ± 0.001	0.952 ± 0.000	0.954 ± 0.000	0.952 ± 0.000	0.951 ± 0.000	0.946 ± 0.000
		V10	0.954 ± 0.000	0.957 ± 0.000	0.957 ± 0.000	0.955 ± 0.000	0.953 ± 0.000	0.948 ± 0.000
DoRA	4.16M	Z500	**0.999** ± 0.000	**0.999** ± 0.000	**0.999** ± 0.000	0.998 ± 0.000	**0.998** ± 0.000	**0.997** ± 0.000
		T850	0.982 ± 0.000	0.983 ± 0.000	0.984 ± 0.000	0.983 ± 0.000	0.982 ± 0.000	0.980 ± 0.000
		T2M	0.963 ± 0.001	0.970 ± 0.000	0.973 ± 0.000	0.976 ± 0.000	0.974 ± 0.000	0.970 ± 0.001
		U10	0.948 ± 0.001	0.952 ± 0.000	0.955 ± 0.000	0.953 ± 0.000	0.952 ± 0.000	0.947 ± 0.000
		V10	0.955 ± 0.000	0.957 ± 0.000	0.958 ± 0.000	0.955 ± 0.000	0.953 ± 0.000	0.948 ± 0.000
AdaLoRA	5.35M	Z500	**0.999** ± 0.000	**0.999** ± 0.000	**0.999** ± 0.000	0.998 ± 0.000	**0.998** ± 0.000	**0.997** ± 0.000
		T850	0.982 ± 0.000	**0.984** ± 0.000	0.984 ± 0.000	0.983 ± 0.000	0.983 ± 0.000	0.981 ± 0.000
		T2M	**0.964** ± 0.001	0.971 ± 0.000	0.974 ± 0.000	0.977 ± 0.000	0.975 ± 0.000	0.971 ± 0.001
		U10	0.951 ± 0.000	0.955 ± 0.000	0.957 ± 0.000	0.956 ± 0.000	0.954 ± 0.000	0.949 ± 0.000
		V10	0.955 ± 0.000	0.958 ± 0.000	0.959 ± 0.000	0.957 ± 0.000	0.955 ± 0.000	0.950 ± 0.000
PiSSA	3.92M	Z500	**0.999** ± 0.000	**0.999** ± 0.000	**0.999** ± 0.000	0.998 ± 0.000	**0.998** ± 0.000	**0.997** ± 0.000
		T850	0.982 ± 0.000	0.983 ± 0.000	0.984 ± 0.000	0.983 ± 0.000	0.982 ± 0.000	0.980 ± 0.000
		T2M	0.963 ± 0.001	0.970 ± 0.001	0.973 ± 0.000	0.976 ± 0.000	0.975 ± 0.000	0.971 ± 0.001
		U10	0.948 ± 0.000	0.952 ± 0.000	0.954 ± 0.000	0.953 ± 0.000	0.952 ± 0.000	0.948 ± 0.000
		V10	0.955 ± 0.000	0.957 ± 0.000	0.958 ± 0.000	0.955 ± 0.000	0.954 ± 0.000	0.949 ± 0.000
OPLoRA	3.92M	Z500	0.998 ± 0.000	**0.999** ± 0.000	**0.999** ± 0.000	0.998 ± 0.000	**0.998** ± 0.000	**0.997** ± 0.000
		T850	0.982 ± 0.000	0.983 ± 0.000	0.984 ± 0.000	0.983 ± 0.000	0.982 ± 0.000	0.980 ± 0.000
		T2M	0.962 ± 0.001	0.970 ± 0.000	0.973 ± 0.000	0.975 ± 0.000	0.974 ± 0.000	0.969 ± 0.000
		U10	0.948 ± 0.001	0.951 ± 0.000	0.954 ± 0.000	0.952 ± 0.000	0.951 ± 0.000	0.946 ± 0.000
		V10	0.954 ± 0.000	0.957 ± 0.000	0.957 ± 0.000	0.954 ± 0.000	0.952 ± 0.000	0.947 ± 0.000
PROLA (Ours)	3.92M	Z500	**0.999** ± 0.000	**0.999** ± 0.000	**0.999** ± 0.000	**0.999** ± 0.000	**0.998** ± 0.000	**0.997** ± 0.000
		T850	**0.983** ± 0.000	**0.984** ± 0.000	**0.985** ± 0.000	**0.984** ± 0.000	**0.984** ± 0.000	**0.982** ± 0.000
		T2M	**0.964** ± 0.001	**0.973** ± 0.001	**0.976** ± 0.000	**0.979** ± 0.000	**0.977** ± 0.001	**0.973** ± 0.001
		U10	**0.953** ± 0.000	**0.957** ± 0.000	**0.960** ± 0.000	**0.958** ± 0.000	**0.957** ± 0.000	**0.952** ± 0.000
		V10	**0.958** ± 0.000	**0.961** ± 0.000	**0.962** ± 0.000	**0.959** ± 0.000	**0.958** ± 0.000	**0.953** ± 0.000

Table 9: **Generalization to regional weather prediction.** ERA5-to-CERRA results report latitude-weighted RMSE (↓) for five variables at 3h and 6h. Regional precipitation results report SEEPS (↓), RMSE in cm (↓), and ACC (↑) at 12h, 24h, and 36h. All methods are trained for 10 epochs and evaluated on the 2018 test split. Best displayed results within each block are boldfaced.

ERA5 → CERRA: regional spatiotemporal downscaling

	3h lead					*6h lead*				
Method	Z500	T850	T2M	U10	V10	Z500	T850	T2M	U10	V10
LoRA	72.14	1.076	2.150	1.370	1.327	86.33	1.092	2.147	1.373	1.340
PiSSA	71.80	1.072	2.126	1.354	1.311	85.72	1.084	2.114	1.352	1.323
OPLoRA	72.56	1.081	2.132	1.371	1.332	86.13	1.099	2.146	1.375	1.348
PROLA	**68.44**	**1.028**	**2.021**	**1.316**	**1.274**	**81.41**	**1.036**	**2.019**	**1.319**	**1.283**

Regional precipitation forecasting: future 6h accumulated precipitation

	12h lead			*24h lead*			*36h lead*		
Method	SEEPS ↓	RMSE ↓	ACC ↑	SEEPS ↓	RMSE ↓	ACC ↑	SEEPS ↓	RMSE ↓	ACC ↑
LoRA	0.608	0.212	0.660	0.706	0.228	0.586	0.784	0.238	0.537
PiSSA	0.612	0.213	0.656	0.711	0.229	0.583	0.788	0.238	0.537
OPLoRA	0.605	0.212	0.661	0.704	0.228	0.587	0.780	0.238	0.538
PROLA	**0.590**	**0.208**	**0.676**	**0.692**	**0.225**	**0.600**	**0.751**	**0.233**	**0.562**

Table 10: **Text-to-SQL transfer.** Execution accuracy (EX) on BIRD for Qwen3.5-9B.

	Pretrained	LoRA	PiSSA	OPLoRA	PROLA
EX (%) ↑	45.50	49.54	51.96	49.41	**52.87**

all four seasons and both tails, and the highest BSS for both tail events. The evaluation protocol is defined in Appendix D.

Spatial spectral fidelity. Table 15 summarizes spherical spectral fidelity for the five representative variables at 3h and 6h, using all 1,458 test cases in 2018 and the seed-0 models. For each variable and lead time, let q_ℓ be the ratio of mean prediction power to mean target power defined in Appendix D. We report the mean absolute log-power ratio, where lower values indicate closer spectral agreement. PROLA has the lowest distance in all ten reported variable–lead combinations. For T2M at 6h, its distance is 0.062, compared with 0.079 for PiSSA, a reduction of 21.5%. Figures 7 and 8 show the corresponding power-ratio curves for eight variables.

Reconstruction and physical diagnostics. Table 16 reports additional diagnostics at 6h on the same 1,458 cases. Variance-normalized reconstruction SNR is computed for each case as $10\log_{10}\big(\mathrm{Var}_w(y)/\,\mathrm{MSE}_w(\hat{y},y)\big)$ and then averaged, where w denotes latitude weighting. For each variable, tendency error is the latitude-weighted RMSE between the predicted and observed changes from $t + h_{\mathrm{end}} - \delta_f$ to $t + h_{\mathrm{end}}$, corresponding to the 5h and 6h leads on this benchmark. We also report RMSE of horizontal divergence and relative vorticity computed from the 850-hPa wind components using spherical finite differences, with periodic longitude and evaluation restricted to $|\varphi| < 89°$. For tendency and wind-derivative errors, squared errors are averaged over space and cases before taking the square root. PROLA improves all six reported diagnostics, including Z500 and T2M reconstruction, their hourly evolution, and wind-field derivatives.

Long-horizon rollouts. Figure 9 compares Z500 and T2M RMSE at lead times from 6h to 240h, normalized by Full FT at each horizon. PROLA has the lowest relative RMSE among the compared PEFT methods at all reported lead times for both variables.

Full fine-tuning learning-rate sweep. Table 17 compares three Full FT learning rates under the same training budget. PROLA outperforms every tested Full FT setting on Z500, T850, and T2M, while Full FT achieves lower U10 and V10 errors.

Training cost measurement. Table 18 reports wall-clock training time per epoch and peak per-GPU memory for the main Aurora PEFT configuration on two NVIDIA RTX 3090 GPUs. The time

Table 11: **Backbone transfer to ArchesWeather.** Latitude-weighted RMSE (↓) on the 2018 test split after one training epoch and total trainable adapter rank 8. PROLA uses $r_p = r_o = 4$. Best displayed results within each lead time are boldfaced.

	6h lead					*12h lead*				
Method	Z500	T850	T2M	U10	V10	Z500	T850	T2M	U10	V10
LoRA	138.22	1.241	1.967	2.055	2.079	123.23	1.291	1.757	**2.368**	3.175
PiSSA	134.41	1.205	1.904	2.556	2.112	121.61	1.247	2.062	2.457	**2.272**
OPLoRA	139.20	1.237	1.839	2.093	2.097	124.38	1.296	1.760	2.519	3.102
PROLA	**133.03**	**1.192**	**1.531**	**2.031**	**2.077**	**119.97**	**1.246**	**1.614**	2.749	2.838
	18h lead					*24h lead*				
Method	Z500	T850	T2M	U10	V10	Z500	T850	T2M	U10	V10
LoRA	144.02	1.261	2.058	**2.344**	2.967	148.49	1.347	1.755	**2.729**	3.768
PiSSA	137.46	1.237	2.036	2.679	**2.141**	148.09	1.334	1.910	3.296	**2.415**
OPLoRA	145.96	1.271	2.052	2.400	2.979	150.05	1.357	1.733	2.875	3.708
PROLA	**132.14**	**1.228**	**1.862**	2.610	2.928	**142.66**	**1.316**	**1.604**	3.191	3.459

Table 12: **Additional resolution gap.** Latitude-weighted RMSE (↓) on the 2018 test split for $5.625°$-to-$2.8125°$ downscaling. Best displayed values are boldfaced.

	3h lead		*6h lead*	
Method	Z500	T2M	Z500	T2M
LoRA	52.27	1.148	76.14	1.141
PiSSA	54.62	1.171	82.54	1.159
OPLoRA	52.08	1.145	76.25	1.134
PROLA	**48.01**	**1.054**	**69.58**	**0.998**

Table 13: **Seasonal robustness.** Latitude-weighted Z500 RMSE (↓) at 6h on the 2018 test split. Best displayed values are boldfaced.

Method	DJF	MAM	JJA	SON
LoRA	82.78	79.28	75.29	79.70
PiSSA	81.70	78.54	74.37	79.25
OPLoRA	83.28	79.88	75.70	79.99
PROLA	**75.99**	**73.22**	**69.73**	**73.65**

is computed from the recorded training-only runtime divided by the number of training epochs, excluding validation and separate test evaluation. PROLA's runtime and peak GPU memory usage fall within the range observed for the PEFT baselines.

H Qualitative Results

Global WeatherBench 2 forecasts. We provide qualitative comparisons on the 2018 test split of WeatherBench 2 [16]. Predictions within each comparison use the same forecast initialization; the global examples shown here are initialized at 18:00 UTC on 10 January 2018. For each map, all predictions and targets are shown after denormalization to physical units. Prediction and target panels for each variable and lead time share the same color scale. The absolute-error difference map is defined as

$$\Delta_{\mathrm{AE}}(\mathbf{s}) = |\hat{y}_{\mathrm{Ours}}(\mathbf{s}) - y(\mathbf{s})| - |\hat{y}_{\mathrm{Base}}(\mathbf{s}) - y(\mathbf{s})|\,, \tag{36}$$

where $\mathbf{s}$ is a grid point. For the displayed variable and lead h, $\hat{y}_{\mathrm{Ours}}(\mathbf{s})$ and $y(\mathbf{s})$ are the corresponding values of $\hat{\mathbf{y}}^{ST}_{t+h}$ and $\mathbf{y}^{\mathrm{HR}}_{t+h}$ at $\mathbf{s}$. Here, Base denotes the comparison method shown in the corresponding panel or column. Thus, negative values indicate locations where PROLA has a smaller absolute error

Table 14: **Spatial-tail performance.** Z500 at 6h on the 2018 test split. Upper and lower tails are defined within each target field as in Appendix D. Best displayed values are boldfaced.

Method	Upper RMSE ↓	Upper BSS ↑	Lower RMSE ↓	Lower BSS ↑
LoRA	56.87	0.514	129.19	0.780
PiSSA	55.50	0.513	126.28	0.781
OPLoRA	56.63	0.512	127.55	0.776
PROLA	**50.80**	**0.545**	**116.36**	**0.795**

Table 15: **Spherical spectral distance.** Mean absolute log-ratio of prediction to target power (↓), evaluated over degrees 1–59 using all 1,458 test cases in 2018. Best displayed values are boldfaced, including rounded ties.

	3h lead					*6h lead*				
Method	Z500	T850	T2M	U10	V10	Z500	T850	T2M	U10	V10
LoRA	0.301	0.627	0.272	0.396	0.372	0.077	0.349	0.081	0.286	0.260
PiSSA	0.304	0.619	0.277	0.392	0.369	0.082	0.331	0.079	0.291	0.263
OPLoRA	0.302	0.628	0.277	0.402	0.378	0.077	0.350	0.087	0.290	0.264
PROLA	**0.294**	**0.589**	**0.229**	**0.329**	**0.312**	**0.072**	**0.293**	**0.062**	**0.248**	**0.204**

than the compared baseline, while positive values indicate locations where the baseline has a smaller absolute error.

Figures 10, 11, and 12 show qualitative prediction results at 2h, 4h, and 6h lead times across the five reported variables, comparing PROLA, the pretrained Aurora [4] backbone, and Full FT against the ground-truth target. The pretrained Aurora baseline spatially upsamples the current coarse state and the pretrained 6h forecast, then interpolates between them to obtain the intermediate 2h and 4h fields, without learned correction heads. Figures 13, 14, and 15 show pairwise T2M comparisons at 2h lead time between PROLA and three PEFT baselines (LoRA [13], PiSSA [14], and OPLoRA [15]), each accompanied by absolute-error maps and pointwise absolute-error difference maps. Figures 16, 17, and 18 show the analogous Z500 comparisons at 2h lead time between PROLA and the same three PEFT baselines.

Regional ERA5-to-CERRA forecasts. Figure 4 in the main paper compares LoRA and PROLA on a T2M forecast initialized at 06:00 UTC on 4 February 2018 and valid six hours later. Figure 19 provides an additional 10 m wind-speed comparison initialized at 00:00 UTC on 3 February 2018 and valid at 06:00 UTC. Wind speed is computed as $\sqrt{u_{10}^2 + v_{10}^2}$ from the predicted or target wind components. The LoRA, PROLA, and CERRA target panels share the same geographic extent, color scale, and display smoothing within each comparison. Quantitative errors are computed on the native fields before display smoothing.

I Additional Ablations

Full pipeline ablations. Table 19 extends the Direct prediction comparison in Table 4 to all lead times from 1h to 6h for LoRA, PiSSA, OPLoRA, and PROLA. Table 20 provides the full per-lead pipeline ablation with PROLA. The compared variants are Direct prediction, the scaffold-only model without C_s and C_t, the model without C_t, the model without C_s, and the full scaffold-and-correct pipeline. The full pipeline achieves the lowest RMSE across all five variables at 3h, with consistent gains on Z500 and T850 across lead times. Unless otherwise stated, the ablations use seed 0 and the same training schedule as the main comparison.

Full dual-branch ablations. Tables 21 and 22 extend the PROLA design ablation in Table 5 to all lead times from 1h to 6h. The former compares principal-only, orthogonal-only, and dual-branch configurations with and without update scaling. The latter compares the $(r_p, r_o) = (4, 4), (6, 2), (2, 6)$ rank splits and the ρ-only and π-only scaling variants, whose definitions are given in Appendix B. All configurations use a total rank of eight. The complete $4 + 4$ configuration achieves the lowest RMSE across all five variables at every reported lead time from 2h to 6h.

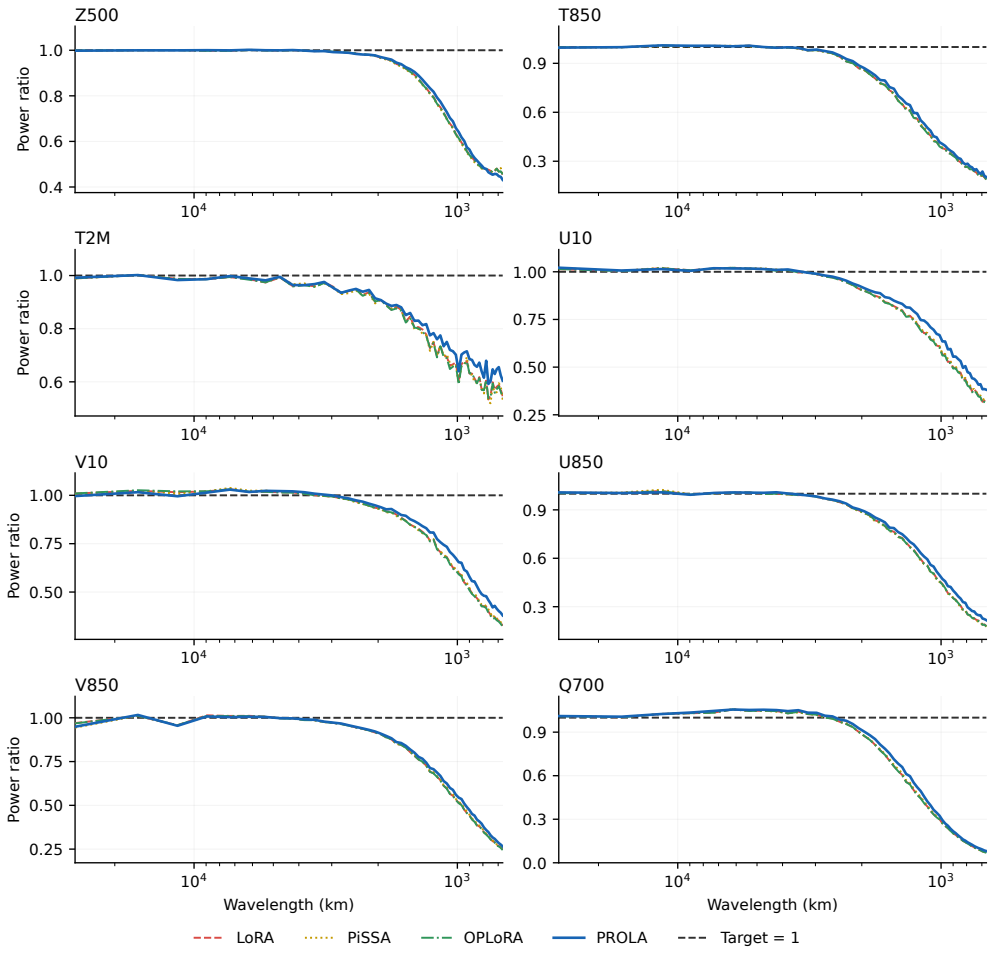


Figure 7: **Spherical power ratios at 3h.** Ratios of mean prediction power to mean target power for eight variables, evaluated on all 1,458 test cases in 2018. The dashed horizontal line marks the target ratio of one.

Table 16: **Reconstruction and physical diagnostics at 6h.** Results use all 1,458 test cases in 2018. SNR denotes variance-normalized reconstruction SNR (dB); tendency RMSE is in $\mathrm{m}^2\,\mathrm{s}^{-2}$ for Z500 and K for T2M; wind-derivative RMSE is in $10^{-6}\,\mathrm{s}^{-1}$. Best displayed values are boldfaced, including rounded ties.

Method	SNR ↑		Tendency RMSE ↓		850-hPa wind RMSE ↓	
	Z500	T2M	Z500	T2M	Divergence	Vorticity
LoRA	30.78	21.46	40.48	0.418	8.70	15.53
PiSSA	30.87	21.62	40.26	0.417	8.61	15.45
OPLoRA	30.73	21.45	40.53	0.418	8.71	15.55
PROLA	**31.48**	**21.87**	**39.89**	**0.415**	**8.46**	**15.18**

Effect of the total rank. Table 23 compares PiSSA, OPLoRA, and PROLA at total ranks 8 and 128, with PROLA using $(r_p, r_o) = (4, 4)$ and $(64, 64)$, respectively. PROLA has the lowest RMSE across all five variables at both reported lead times within each rank budget. Increasing PROLA's rank reduces Z500 RMSE from 50.30 to 48.19 at 3h and from 73.07 to 68.21 at 6h, with improvements also observed for the other four variables.

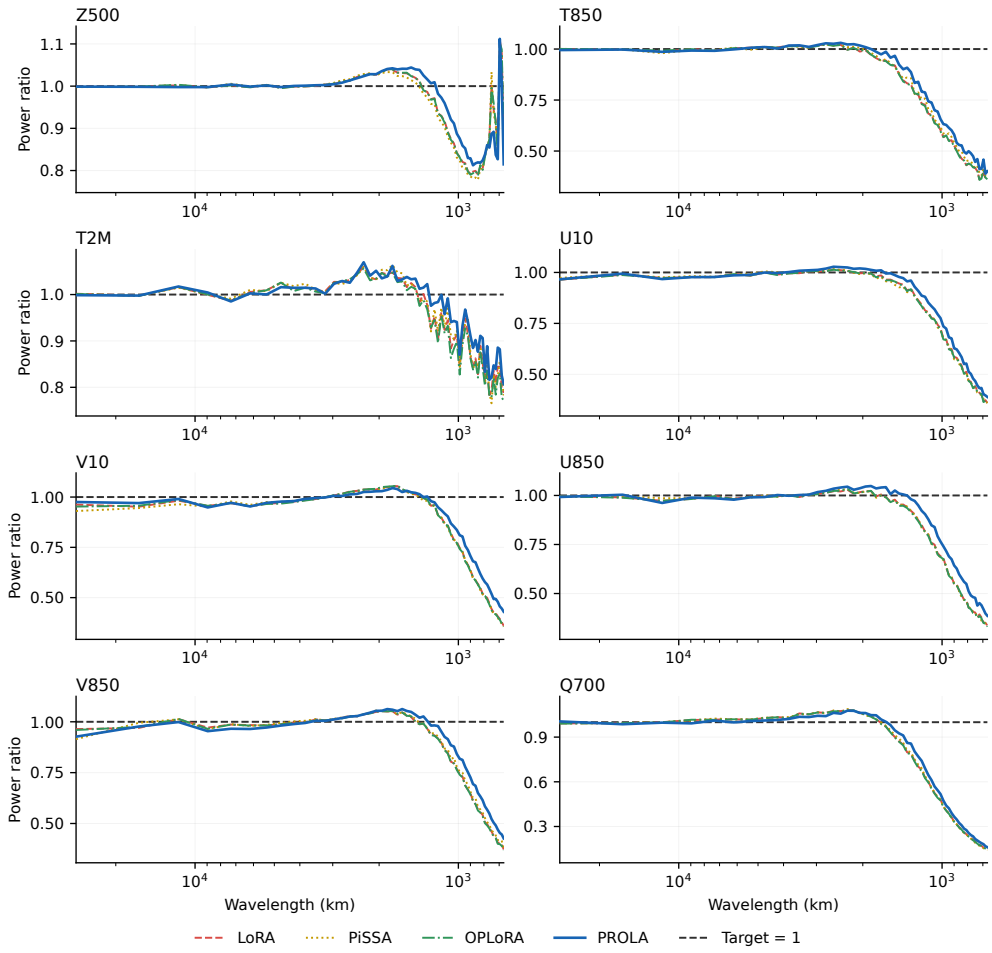


Figure 8: **Spherical power ratios at 6h.** Ratios of mean prediction power to mean target power for eight variables, evaluated on all 1,458 test cases in 2018. The dashed horizontal line marks the target ratio of one.

Table 17: **Full fine-tuning learning-rate sweep.** Latitude-weighted RMSE (↓) at 6h on the 2018 test split. The main Full FT configuration uses a backbone learning rate of 5×10^{-5}. Best displayed values are boldfaced, including rounded ties.

Method / backbone LR	Z500	T850	T2M	U10	V10
Full FT, 2.5×10^{-5}	116.94	1.116	1.306	1.178	1.235
Full FT, 5×10^{-5}	108.90	1.105	1.304	1.122	1.203
Full FT, 10^{-4}	106.06	1.096	1.324	**1.093**	**1.165**
PROLA	**73.07**	**0.958**	**1.211**	1.236	1.234

J Limitations and Future Work

PSWD is constrained by atmospheric chaos and the limited fine-scale information available in coarse inputs. Our weather evaluation covers a global reanalysis benchmark and regional prediction tasks, including ERA5-to-CERRA transfer to a distinct target dataset. Future work could extend this evaluation to broader regions and target sources, and incorporate physical constraints and probabilistic prediction to improve forecast consistency and reliability.

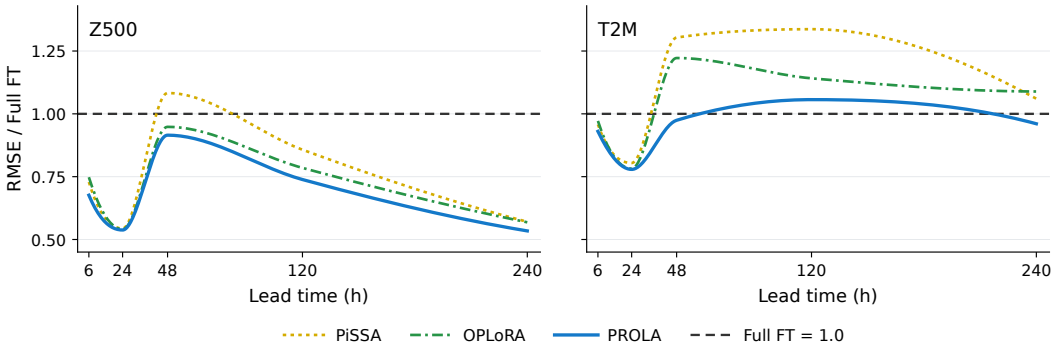


Figure 9: **Relative RMSE over long rollouts.** Z500 (left) and T2M (right) RMSE normalized by Full FT at each lead time. The dashed horizontal line denotes Full FT at 1.0; lower values indicate lower RMSE.

Table 18: **Training cost of the PEFT baselines and PROLA.** We report the average training time per epoch and the peak per-GPU memory recorded during training.

Method	Training Time / Epoch	Peak GPU Memory
LoRA	54m 38s	15.56 GiB
DoRA	71m 19s	15.56 GiB
AdaLoRA	58m 45s	15.63 GiB
PiSSA	55m 52s	15.56 GiB
OPLoRA	60m 56s	15.57 GiB
PROLA	67m 18s	15.56 GiB

K Broader Impacts

This work aims to improve the efficiency and accuracy of high-resolution weather prediction from coarse global states. More accessible spatiotemporal downscaling may benefit applications such as renewable-energy planning, flood-risk assessment, aviation safety, and local weather-risk analysis, where fine spatial and temporal information is important. However, inaccurate or overconfident forecasts could negatively affect downstream decisions if used without proper uncertainty estimates or expert oversight. For real-world use, such models should be integrated with uncertainty estimates, operational validation protocols, and expert oversight to ensure reliable and responsible decision support.

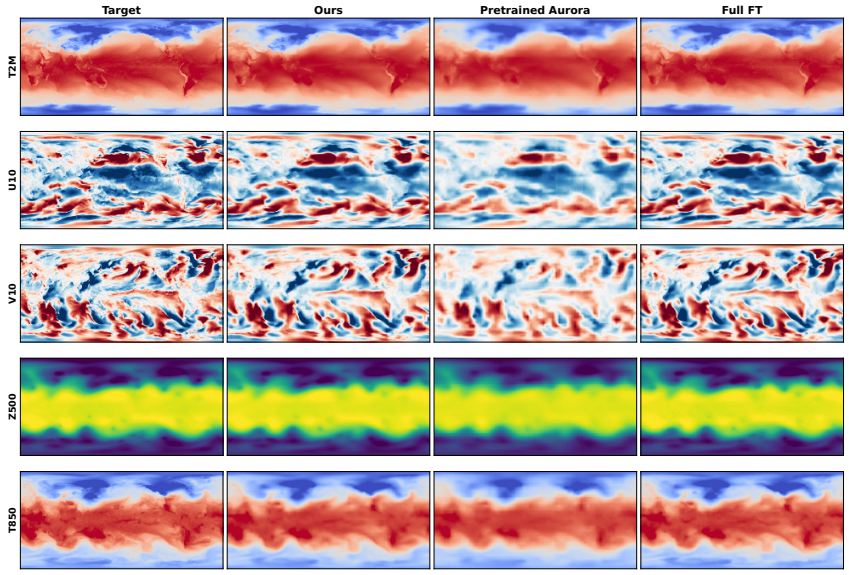


Figure 10: Qualitative comparison at 2h lead time for multiple variables, showing the target, PROLA, pretrained Aurora, and Full FT predictions.

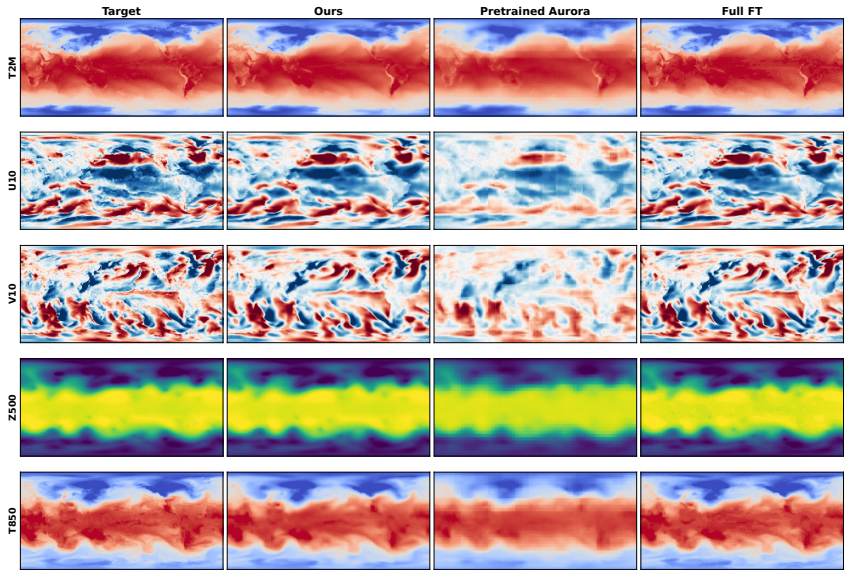


Figure 11: Qualitative comparison at 4h lead time for multiple variables, showing the target, PROLA, pretrained Aurora, and Full FT predictions.

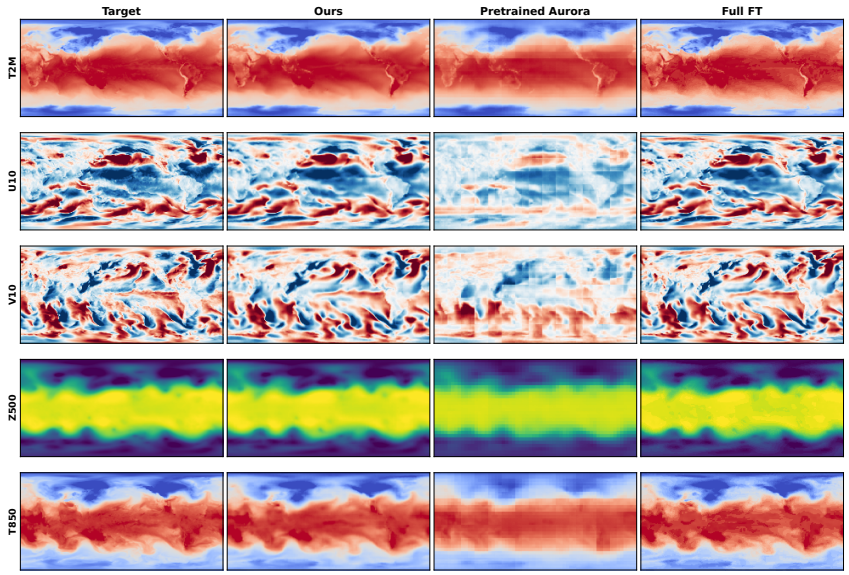


Figure 12: Qualitative comparison at 6h lead time for multiple variables, showing the target, PROLA, pretrained Aurora, and Full FT predictions.

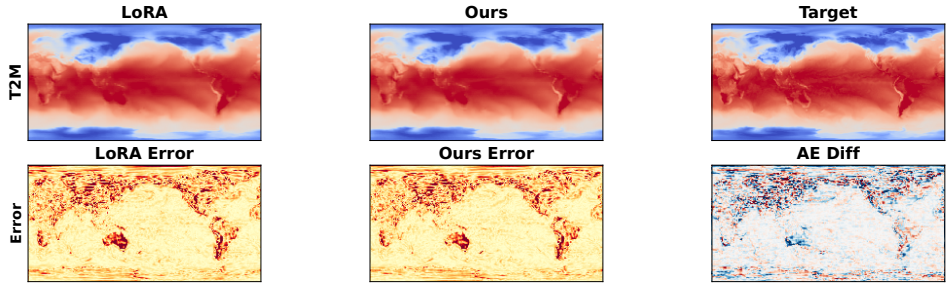


Figure 13: T2M comparison at 2h lead time between LoRA and PROLA, with absolute-error maps and the pointwise absolute-error difference map.

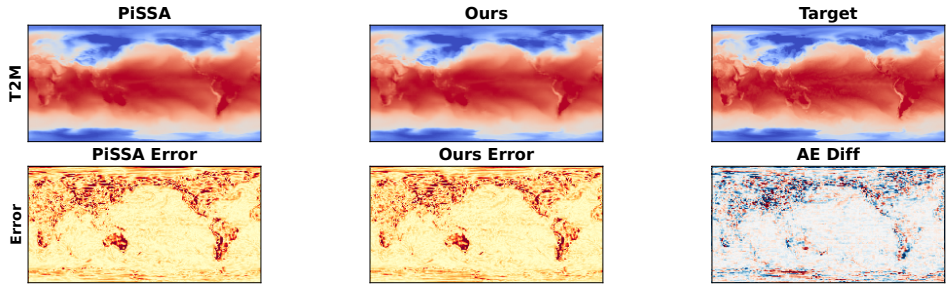


Figure 14: T2M comparison at 2h lead time between PiSSA and PROLA, with absolute-error maps and the pointwise absolute-error difference map.

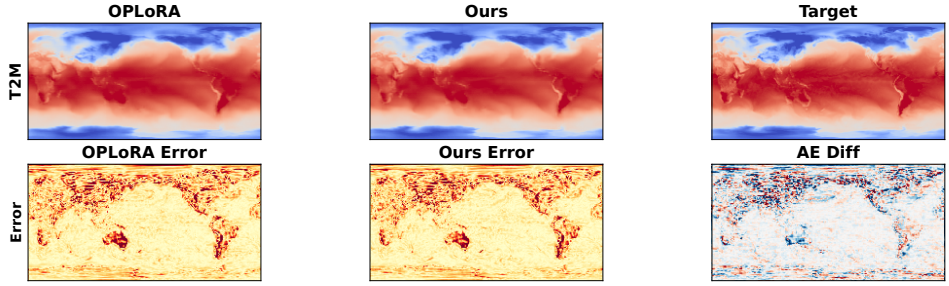


Figure 15: T2M comparison at 2h lead time between OPLoRA and PROLA, with absolute-error maps and the pointwise absolute-error difference map.

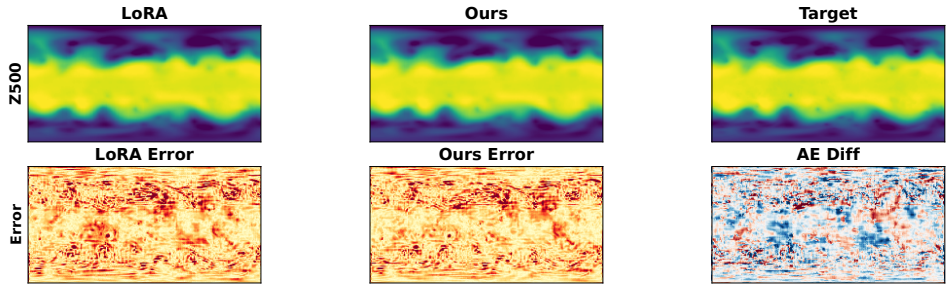


Figure 16: Z500 comparison at 2h lead time between LoRA and PROLA, with absolute-error maps and the pointwise absolute-error difference map.

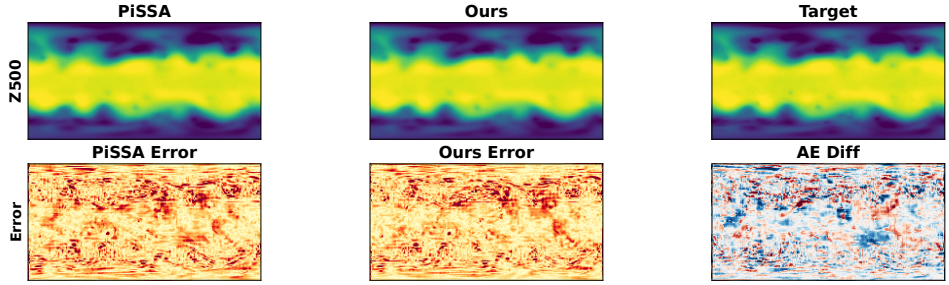


Figure 17: Z500 comparison at 2h lead time between PiSSA and PROLA, with absolute-error maps and the pointwise absolute-error difference map.

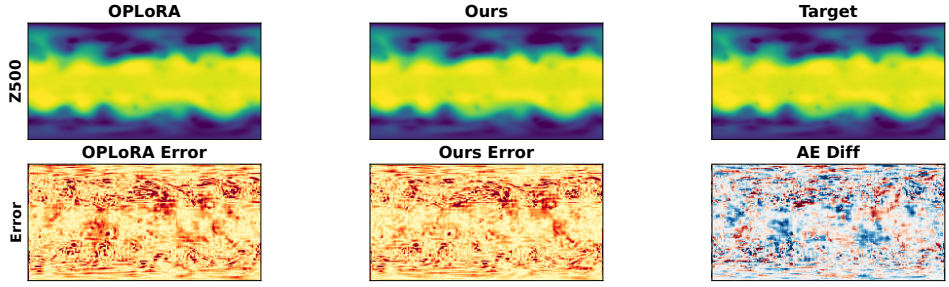


Figure 18: Z500 comparison at 2h lead time between OPLoRA and PROLA, with absolute-error maps and the pointwise absolute-error difference map.

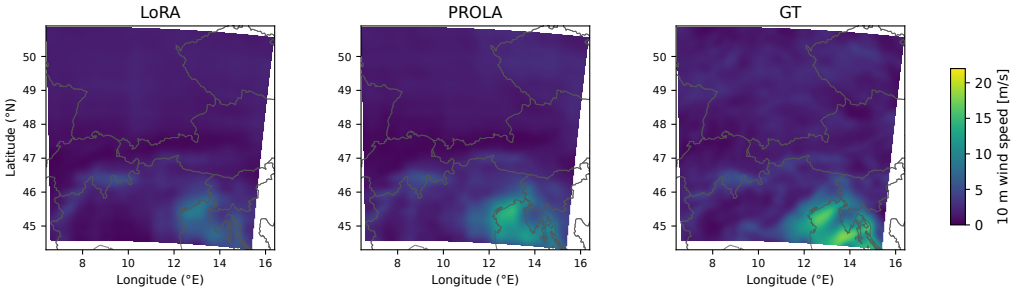


Figure 19: **Qualitative ERA5-to-CERRA wind prediction at 6h.** 10 m wind speed from LoRA, PROLA, and the CERRA target, valid at 06:00 UTC on 3 February 2018. All panels share the same color scale and display smoothing.

Table 19: **Direct prediction across PEFT methods.** We report latitude-weighted RMSE (↓) at 1–6h. Best displayed values, including ties, are boldfaced.

Method	Variable	RMSE ↓ 1h	2h	3h	4h	5h	6h
LoRA	Z500	**101.90**	72.91	58.91	72.14	98.30	129.60
	T850	1.009	0.973	0.979	1.044	1.118	1.214
	T2M	1.419	1.303	1.256	1.230	1.294	1.394
	U10	1.426	1.438	1.481	1.567	1.652	1.755
	V10	1.376	1.417	1.484	1.585	1.677	1.781
PiSSA	Z500	104.90	73.76	56.47	68.45	95.47	128.00
	T850	1.009	0.955	0.942	0.992	1.058	1.152
	T2M	**1.416**	1.261	1.191	1.134	1.185	1.292
	U10	1.332	1.292	1.272	1.298	1.327	1.392
	V10	1.272	1.243	1.239	1.280	1.324	1.405
OPLoRA	Z500	103.29	73.17	58.15	71.78	98.97	131.16
	T850	1.013	0.977	0.983	1.047	1.122	1.219
	T2M	1.426	1.304	1.254	1.224	1.288	1.388
	U10	1.409	1.420	1.463	1.551	1.637	1.741
	V10	1.361	1.396	1.457	1.551	1.636	1.735
PROLA	Z500	104.37	**72.09**	**53.13**	**64.82**	**92.24**	**125.07**
	T850	**0.991**	**0.924**	**0.898**	**0.941**	**1.002**	**1.095**
	T2M	1.432	**1.245**	**1.165**	**1.109**	**1.174**	**1.289**
	U10	**1.308**	**1.247**	**1.211**	**1.233**	**1.271**	**1.358**
	V10	**1.245**	**1.202**	**1.177**	**1.203**	**1.244**	**1.344**

Table 20: **Full ablation results on scaffold-and-correct pipeline design.** We report latitude-weighted RMSE (↓) and ACC (↑) at 1–6h. Best displayed values, including ties, are boldfaced.

Method / Variant	Variable	RMSE ↓						ACC ↑					
		1h	2h	3h	4h	5h	6h	1h	2h	3h	4h	5h	6h
Direct (PROLA)	Z500	104.37	72.09	53.13	64.82	92.24	125.07	0.995	0.997	**0.999**	0.998	0.996	0.992
	T850	0.991	0.924	0.898	0.941	1.002	1.095	0.980	0.983	0.984	0.982	0.980	0.976
	T2M	1.432	1.245	1.165	1.109	1.174	1.289	0.960	0.970	0.974	0.976	0.974	0.968
	U10	1.308	1.247	1.211	1.233	1.271	1.358	0.946	0.951	0.954	0.952	0.949	0.942
	V10	1.245	1.202	1.177	1.203	1.244	1.344	0.953	0.956	0.958	0.956	0.953	0.945
w/o C_s, C_t	Z500	155.56	120.51	87.73	65.77	71.87	111.87	0.989	0.993	0.996	0.998	**0.998**	0.994
	T850	1.226	1.076	0.959	0.922	0.935	1.037	0.970	0.977	0.982	0.983	0.982	0.979
	T2M	1.745	1.527	1.307	1.127	1.088	1.289	0.940	0.955	0.967	0.976	0.977	0.971
	U10	1.665	1.455	1.277	1.189	1.194	1.339	0.915	0.936	0.950	0.956	0.956	0.948
	V10	1.694	1.464	1.270	1.175	1.196	1.371	0.913	0.936	0.952	0.958	0.957	0.946
w/o C_t	Z500	154.94	118.93	84.53	60.31	66.47	108.80	0.989	0.994	0.997	0.998	**0.998**	0.995
	T850	1.219	1.062	0.938	0.896	0.909	1.016	0.971	0.978	0.982	**0.984**	**0.983**	0.980
	T2M	1.740	1.515	1.290	1.107	**1.070**	1.281	0.941	0.955	0.968	0.976	**0.978**	0.971
	U10	1.659	1.442	1.259	1.169	1.177	1.332	0.915	0.937	0.952	0.957	**0.957**	0.949
	V10	1.688	1.452	1.254	1.159	1.183	1.365	0.914	0.938	0.954	0.959	**0.958**	0.947
w/o C_s	Z500	55.84	50.74	53.35	58.16	62.84	78.25	**0.999**	**0.999**	**0.999**	0.998	**0.998**	**0.997**
	T850	0.942	0.904	0.880	0.900	0.920	0.974	0.982	**0.984**	0.984	**0.984**	**0.983**	0.981
	T2M	1.374	1.213	1.138	1.069	1.109	**1.209**	**0.963**	**0.972**	0.975	**0.978**	0.976	**0.972**
	U10	1.250	1.193	1.155	1.165	1.178	**1.233**	0.951	0.955	0.958	0.957	0.956	**0.952**
	V10	1.199	1.155	1.133	1.157	1.179	1.237	0.956	0.960	0.961	0.959	**0.958**	0.953
Full pipeline	Z500	**52.75**	**47.15**	**50.30**	**55.23**	**58.78**	**73.07**	**0.999**	**0.999**	**0.999**	**0.999**	**0.998**	**0.997**
	T850	**0.924**	**0.887**	**0.866**	**0.888**	**0.907**	**0.958**	**0.983**	**0.984**	**0.985**	**0.984**	**0.983**	**0.982**
	T2M	**1.373**	**1.205**	**1.128**	**1.060**	1.107	1.211	**0.963**	**0.972**	**0.976**	**0.978**	0.976	**0.972**
	U10	**1.226**	**1.174**	**1.140**	**1.154**	**1.173**	1.236	**0.953**	**0.957**	**0.959**	**0.958**	**0.957**	**0.952**
	V10	**1.176**	**1.136**	**1.119**	**1.146**	**1.171**	**1.234**	**0.958**	**0.961**	**0.962**	**0.960**	**0.958**	**0.954**

Table 21: **Full ablation results on PROLA branch composition and update scaling.** We report latitude-weighted RMSE (↓) and ACC (↑) at 1–6h. Best displayed values, including ties, are boldfaced. All configurations use $r_p + r_o = 8$; checkmarks indicate update scaling.

Configuration	r_p	r_o	Scaling	Variable	RMSE ↓						ACC ↑					
					1h	2h	3h	4h	5h	6h	1h	2h	3h	4h	5h	6h
Principal	8	0	–	Z500	55.57	50.45	53.66	58.98	63.55	78.55	**0.999**	**0.999**	**0.999**	0.998	**0.998**	**0.997**
				T850	0.941	0.910	0.894	0.918	0.937	0.987	0.982	0.983	0.984	0.983	0.982	0.980
				T2M	1.383	1.258	1.193	1.125	1.152	1.248	0.963	0.969	0.973	0.976	0.975	0.970
				U10	1.288	1.240	1.209	1.225	1.241	1.294	0.948	0.952	0.954	0.953	0.952	0.947
				V10	1.227	1.194	1.178	1.204	1.228	1.290	0.954	0.957	0.958	0.956	0.954	0.949
Principal	8	0	✓	Z500	54.94	49.80	53.06	58.35	62.77	77.71	**0.999**	**0.999**	**0.999**	0.998	**0.998**	**0.997**
				T850	0.935	0.904	0.887	0.911	0.930	0.979	0.982	**0.984**	0.984	0.983	**0.983**	0.981
				T2M	1.378	1.251	1.184	1.116	1.144	1.241	0.963	0.970	0.973	0.976	0.975	0.970
				U10	1.275	1.228	1.198	1.214	1.232	1.289	0.949	0.953	0.955	0.953	0.952	0.948
				V10	1.213	1.181	1.166	1.194	1.221	1.285	0.955	0.958	0.959	0.956	0.954	0.950
Orthogonal	0	8	–	Z500	55.59	51.05	54.54	60.12	64.98	79.83	**0.999**	**0.999**	**0.999**	0.998	**0.998**	**0.997**
				T850	0.942	0.913	0.899	0.924	0.944	0.994	0.982	0.983	0.984	0.983	0.982	0.980
				T2M	1.384	1.261	1.197	1.138	1.174	1.272	0.963	0.969	0.972	0.975	0.974	0.969
				U10	1.290	1.243	1.216	1.235	1.255	1.314	0.948	0.952	0.954	0.952	0.950	0.946
				V10	1.233	1.201	1.188	1.217	1.246	1.314	0.954	0.956	0.957	0.955	0.953	0.947
Orthogonal	0	8	✓	Z500	54.63	49.55	52.81	58.11	62.32	76.67	**0.999**	**0.999**	**0.999**	0.998	**0.998**	**0.997**
				T850	0.934	0.903	0.886	0.910	0.929	0.977	0.982	**0.984**	0.984	0.983	**0.983**	0.981
				T2M	1.409	1.254	1.175	1.108	1.155	1.260	0.961	0.970	0.973	0.976	0.974	0.970
				U10	1.275	1.223	1.189	1.204	1.223	1.284	0.949	0.953	0.956	0.954	0.953	0.948
				V10	1.203	1.169	1.157	1.188	1.216	1.278	0.956	0.959	0.960	0.957	0.955	0.950
Both branches	4	4	–	Z500	53.55	48.22	51.36	56.31	60.04	74.38	**0.999**	**0.999**	**0.999**	0.998	**0.998**	**0.997**
				T850	0.930	0.894	0.875	0.898	0.917	0.967	**0.983**	**0.984**	**0.985**	**0.984**	**0.983**	0.981
				T2M	**1.347**	1.207	1.143	1.080	1.120	1.221	**0.965**	**0.972**	0.975	**0.978**	**0.976**	0.971
				U10	1.248	1.198	1.165	1.179	1.196	1.256	0.951	0.955	0.958	0.956	0.955	0.950
				V10	1.187	1.149	1.133	1.163	1.188	1.250	0.957	0.960	0.961	0.959	0.957	0.952
Complete	4	4	✓	Z500	**52.75**	**47.15**	**50.30**	**55.23**	**58.78**	**73.07**	**0.999**	**0.999**	**0.999**	**0.999**	**0.998**	**0.997**
				T850	**0.924**	**0.887**	**0.866**	**0.888**	**0.907**	**0.958**	**0.983**	**0.984**	**0.985**	**0.984**	**0.983**	**0.982**
				T2M	1.373	**1.205**	**1.128**	**1.060**	**1.107**	**1.211**	0.963	**0.972**	**0.976**	**0.978**	**0.976**	**0.972**
				U10	**1.226**	**1.174**	**1.140**	**1.154**	**1.173**	**1.236**	**0.953**	**0.957**	**0.959**	**0.958**	**0.957**	**0.952**
				V10	**1.176**	**1.136**	**1.119**	**1.146**	**1.171**	**1.234**	**0.958**	**0.961**	**0.962**	**0.960**	**0.958**	**0.954**

Table 22: **Full ablation results on rank allocation and scaling components.** We report latitude-weighted RMSE (↓) and ACC (↑) at 1–6h. Best displayed values, including ties, are boldfaced. All configurations use $r_p + r_o = 8$; checkmarks indicate update scaling.

Configuration	r_p	r_o	Scaling	Variable	RMSE ↓						ACC ↑					
					1h	2h	3h	4h	5h	6h	1h	2h	3h	4h	5h	6h
Principal-heavy	6	2	✓	Z500	53.90	48.22	51.46	56.54	60.16	74.19	**0.999**	**0.999**	**0.999**	0.998	**0.998**	**0.997**
				T850	0.933	0.897	0.876	0.899	0.919	0.970	0.982	**0.984**	**0.985**	**0.984**	**0.983**	0.981
				T2M	1.385	1.229	1.153	1.086	1.127	1.225	0.962	0.971	0.974	0.977	**0.976**	0.971
				U10	1.245	1.193	1.161	1.176	1.193	1.249	0.951	0.955	0.958	0.956	0.955	0.951
				V10	1.189	1.152	1.136	1.165	1.189	1.244	0.957	0.960	0.961	0.958	0.957	0.953
Orthogonal-heavy	2	6	✓	Z500	54.52	48.95	52.52	58.10	62.32	76.35	**0.999**	**0.999**	**0.999**	0.998	**0.998**	**0.997**
				T850	0.938	0.906	0.890	0.915	0.934	0.983	0.982	0.983	0.984	0.983	0.982	0.981
				T2M	1.403	1.250	1.176	1.113	1.161	1.263	0.961	0.970	0.973	0.976	0.974	0.969
				U10	1.276	1.229	1.200	1.216	1.231	1.285	0.949	0.953	0.955	0.953	0.952	0.948
				V10	1.205	1.172	1.160	1.192	1.220	1.282	0.956	0.958	0.959	0.956	0.955	0.950
ρ only	4	4	✓	Z500	54.17	48.87	52.16	57.44	61.58	76.06	**0.999**	**0.999**	**0.999**	0.998	**0.998**	**0.997**
				T850	0.937	0.903	0.885	0.909	0.928	0.978	0.982	**0.984**	0.984	0.983	**0.983**	0.981
				T2M	1.374	1.233	1.168	1.106	1.144	1.240	0.963	0.971	0.974	0.976	0.975	0.971
				U10	1.265	1.219	1.188	1.201	1.217	1.273	0.950	0.953	0.956	0.954	0.953	0.949
				V10	1.202	1.165	1.151	1.180	1.204	1.264	0.956	0.959	0.960	0.957	0.956	0.951
π only	4	4	✓	Z500	54.08	48.76	51.94	57.06	61.05	75.45	**0.999**	**0.999**	**0.999**	0.998	**0.998**	**0.997**
				T850	0.934	0.900	0.882	0.906	0.925	0.976	0.982	**0.984**	0.984	**0.984**	**0.983**	0.981
				T2M	**1.358**	1.219	1.155	1.093	1.137	1.235	**0.964**	0.971	0.974	0.977	0.975	0.971
				U10	1.256	1.208	1.176	1.191	1.208	1.267	0.950	0.954	0.957	0.955	0.954	0.950
				V10	1.196	1.159	1.144	1.175	1.200	1.261	0.957	0.959	0.960	0.958	0.956	0.952
Complete	4	4	✓	Z500	**52.75**	**47.15**	**50.30**	**55.23**	**58.78**	**73.07**	**0.999**	**0.999**	**0.999**	**0.999**	**0.998**	**0.997**
				T850	**0.924**	**0.887**	**0.866**	**0.888**	**0.907**	**0.958**	**0.983**	**0.984**	**0.985**	**0.984**	**0.983**	**0.982**
				T2M	1.373	**1.205**	**1.128**	**1.060**	**1.107**	**1.211**	0.963	**0.972**	**0.976**	**0.978**	**0.976**	**0.972**
				U10	**1.226**	**1.174**	**1.140**	**1.154**	**1.173**	**1.236**	**0.953**	**0.957**	**0.959**	**0.958**	**0.957**	**0.952**
				V10	**1.176**	**1.136**	**1.119**	**1.146**	**1.171**	**1.234**	**0.958**	**0.961**	**0.962**	**0.960**	**0.958**	**0.954**

Table 23: **Effect of the total trainable rank.** Latitude-weighted RMSE (↓) at 3h and 6h. PROLA uses equal principal and orthogonal ranks. Best displayed values within each rank budget are boldfaced.

Method	*3h lead*					*6h lead*				
	Z500	T850	T2M	U10	V10	Z500	T850	T2M	U10	V10
Total rank $r = 8$										
PiSSA	53.66	0.894	1.193	1.209	1.178	78.55	0.987	1.248	1.294	1.290
OPLoRA	54.54	0.899	1.197	1.216	1.188	79.83	0.994	1.272	1.314	1.314
PROLA (4+4)	**50.30**	**0.866**	**1.128**	**1.140**	**1.119**	**73.07**	**0.958**	**1.211**	**1.236**	**1.234**
Total rank $r = 128$										
PiSSA	51.35	0.872	1.124	1.137	1.112	74.56	0.962	1.194	1.248	1.239
OPLoRA	57.33	0.938	1.258	1.272	1.233	85.79	1.031	1.332	1.372	1.373
PROLA (64+64)	**48.19**	**0.833**	**1.072**	**1.077**	**1.063**	**68.21**	**0.933**	**1.126**	**1.182**	**1.190**

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: In Abstract and Introduction.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: In the appendix.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: In the appendix.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: In Experiments and appendix.

   Guidelines:

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

5. **Open access to data and code**

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

Answer: [Yes]

Justification: We release the code.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: In Experiments and appendix.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: We report mean and standard deviation over three seeds for the main comparisons.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: In the appendix.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We follow NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: In the appendix.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: We do not release any datasets.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: In the appendix.

    Guidelines:

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

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: We provide code in the supplementary material.

    Guidelines:

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

14. **Crowdsourcing and research with human subjects**

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

    Answer: [N/A]

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

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

Answer: [Yes]

Justification: We evaluate Qwen3.5-9B on BIRD in appendix.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.