ANCHORED ADAPTIVE PROBES WITH SCALE-AWARE RENDERING FOR ATMOSPHERIC SUPER-RESOLUTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Arbitrary-scale spatial super-resolution of atmospheric fields aims to alleviate limitations in acquiring high-resolution data by enabling a single model to meet resolution requirements, including those at unseen scales, without scale-specific training. However, reconstruction performance may be constrained if models do not adequately account for region-specific detail recovery requirements and variations in the target scale. Jointly modeling the spatial allocation and scale-adaptive use of reference information can align the distribution of feature support with regional refinement needs while adapting its spatial range of influence to the target scale. We propose A^{2}PSR, a reconstruction model that combines anchored adaptive probes with scale-aware rendering. The model treats input grid nodes as stable anchors that preserve basic information and generates probes carrying auxiliary information within grid cells on demand, forming the reference information needed for reconstruction. Scale-aware rendering then adaptively regulates the influence of this reference information to enable continuous-scale reconstruction. Experiments on ERA5 show that A^{2}PSR performs well within the training scale range of 1–$2\times$ and outperforms all evaluated arbitrary-scale methods at the tested extrapolation scales, including $22.5\times$, demonstrating good generalization across scales.

1 INTRODUCTION

High-resolution atmospheric fields provide rich spatial information and serve as an important basis for accurately characterizing weather processes Roberts et al. (2025); Jin et al. (2023). However, obtaining such high-resolution data is constrained by both observational limitations Hersbach et al. (2020); Gelaro et al. (2017) and the high computational and storage costs of numerical simulations Acosta et al. (2024). Deep learning-based arbitrary-scale atmospheric spatial super-resolution methods can reduce reliance on direct high-resolution observations and reconstruct high-resolution atmospheric fields with low computational and storage overhead Chen et al. (2026).

In recent years, arbitrary-scale super-resolution methods have primarily constructed continuous fields using coordinate-based implicit representations and learnable Gaussian primitives. Coordinate-based implicit representations Chen et al. (2021); Jiang et al. (2025); Chen et al. (2026) predict field values at arbitrary target locations by querying local latent features on a regular grid, whereas Gaussian-splatting methods Chen et al. (2025); Peng et al. (2025) generate target fields at different scales by rendering Gaussian primitives with learnable positions and shapes. These methods use local features at fixed grid locations or learnable primitives to carry reference information for reconstruction, and perform feature querying or primitive rendering according to the target locations and scale to generate target fields at different scales. Arbitrary-scale reconstruction of atmospheric fields also requires models to account for region-specific detail recovery while adaptively adjusting the spatial extent and contribution of reference information across target scales. This need is evident in the reconstruction of atmospheric fields across scales. As shown in Figure 1, we illustrate the super-resolution reconstruction process at different target scales using complex and smooth regions of ground-truth 2 m temperature fields. Different target scales correspond to different query-point configurations, and gains in detail recovery vary significantly across regions as spatial resolution increases. However, coordinate-based implicit methods that primarily query predefined grid neighborhoods are constrained by the fixed grid structure in their choice of reference locations, making it difficult to flexibly incorporate auxiliary reference information according to local detail requirements. Although Gaussian-splatting methods can flexibly represent local structures through adaptive

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

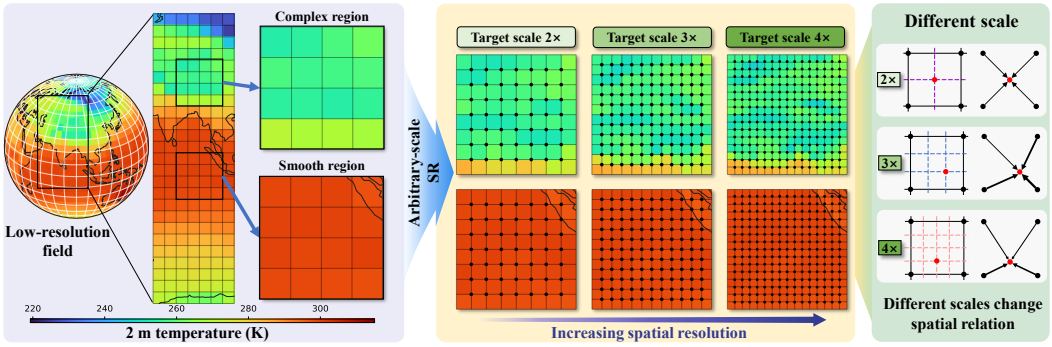

Figure 1: Illustration of spatially varying detail gains and scale-dependent spatial relationships using ground-truth 2 m temperature fields at different resolutions. Different target scales correspond to different query-point configurations relative to coarse-grid reference points, while gains in resolved detail vary significantly across regions as spatial resolution increases.

Gaussian primitives, this representational flexibility alone does not guarantee that the primitives' spatial ranges of influence are adapted to the target scale.

From a grid perspective, the detail gains brought by super-resolution can be understood as previously unresolved structures within the original grid cells becoming clearer through refinement. The details revealed in different cells vary, and using only the sampling locations of the low-resolution input as references may be insufficient to satisfy the refinement needs of some regions. Introducing auxiliary reference locations beyond the original sampling locations can therefore provide support for refinement within grid cells. However, the support provided by these auxiliary locations may vary across target scales, and keeping their spatial ranges of influence fixed may limit the effective use of auxiliary information. The target scale can thus be incorporated into the adjustment of these ranges, allowing auxiliary information to contribute effectively over appropriate spatial extents. We therefore argue that arbitrary-scale reconstruction of atmospheric fields requires joint modeling of region-specific detail recovery and scale-adaptive utilization of reference information.

Building on the above analysis, we propose A^{2}PSR, which combines anchored adaptive probes with scale-aware rendering for arbitrary-scale atmospheric super-resolution. The model uses coarse-grid points from low-resolution data as stable anchors to preserve basic information and jointly considers regional structures, cross-region dependencies, and the target scale to adaptively allocate probes within coarse-grid cells, providing latent feature support for region-specific detail recovery. On this basis, the model adjusts the spatial support range of information carried by anchors and probes according to the target scale and performs weighted aggregation of reference information within the effective support range at each target location to reconstruct the target field. By combining content-adaptive probe representations with scale-aware information aggregation, A^{2}PSR jointly incorporates regional refinement demands and the need for scale-adaptive use of reference information into the reconstruction process, supporting region-specific detail recovery across different target scales.

Our main contributions are summarized as follows:

1. We analyze arbitrary-scale atmospheric field reconstruction from the perspective of refinement within coarse-grid cells, link region-specific detail recovery requirements to the use of reference information at different target scales, and propose a modeling approach that accounts for both regional differences and scale variations.
2. We propose A^{2}PSR, which uses coarse-grid nodes as stable anchors, adaptively allocates probes based on regional structures, cross-region dependencies, and the target scale, and adjusts the spatial support range of information through scale-aware rendering to adaptively aggregate information.
3. We conduct arbitrary-scale atmospheric spatial super-resolution experiments on ERA5. The results show that the proposed method performs well within the training scale range and outperforms the compared arbitrary-scale methods at extrapolation scales, demonstrating good generalization across scales.

2 Related Work

2.1 Atmospheric Spatial Super-Resolution

Atmospheric spatial super-resolution aims to recover fine-scale spatial structures from coarse-resolution atmospheric fields. DeepRU and related methods Höhlein et al. (2020); Sha et al. (2020) employ U-Net-based convolutional networks for meteorological field downscaling. GeoFAR Xu et al. (2026) enhances high-frequency detail recovery through frequency-aware convolution and geographic encoding, with a representation module that can be integrated with different super-resolution backbones. Swin Transformer-based downscaling methods Shiraishi et al. (2023); Pérez et al. (2024); Li et al. (2025) employ shifted-window self-attention Liu et al. (2021) to model spatial dependencies for high-resolution reconstruction of meteorological fields. However, models with scale-specific reconstruction modules are typically trained for predefined source–target resolution pairs and often require architectural adjustments or retraining when the target scale changes. To move beyond fixed output grids, MINet Chen et al. (2026) constructs multi-scale implicit features and predicts atmospheric field values through coordinate-based feature retrieval and a coordinate-conditioned mixture-of-experts decoder, enabling arbitrary-scale reconstruction using only fixed-scale training data. GSSA-ViT Han et al. (2026) constructs continuous atmospheric field representations using learnable Gaussian primitives and reconstructs atmospheric fields at arbitrary resolutions through differentiable rendering. Although these methods remove the constraint of a fixed output grid, regional variations in atmospheric structures and changes in the target scale impose different requirements on local detail recovery. Further investigation is needed into how spatial representations and reconstruction mechanisms can be adapted to better meet these requirements in arbitrary-scale atmospheric downscaling.

2.2 Spatial Representation and Rendering for Arbitrary-Scale Reconstruction

To accommodate reconstruction requirements across regional content and target scales, arbitrary-scale reconstruction requires a spatial representation that can be evaluated at target coordinates and a rendering mechanism that converts this representation into values on the target grid. MetaSR Hu et al. (2019) retains feature representations on a regular grid and generates reconstruction filters conditioned on the target scale and relative position. LIIF Chen et al. (2021) instead predicts values at arbitrary locations from local latent features on a regular grid through a coordinate-conditioned implicit decoder. SRNO Wei & Zhang (2023), CiaoSR Cao et al. (2023), and HIIF Jiang et al. (2025) further introduce neural-operator-based non-local modeling, attention-based local and non-local feature aggregation, and hierarchical positional encoding, respectively. Another line of work constructs continuous spatial representations using Gaussian primitives. GaussianSR Hu et al. (2025), GSASR Chen et al. (2025), and ContinuousSR Peng et al. (2025) adopt Gaussian-based continuous representations and achieve arbitrary-scale image reconstruction through spatial aggregation of Gaussian kernels. However, the regular-grid implicit methods described above still tie feature-bearing locations to a predefined grid, while Gaussian representation methods determine spatial contributions primarily through primitive positions, shapes, and rasterization mechanisms. The allocation of representation locations and the adjustment of their spatial ranges of influence remain constrained by the established spatial modeling schemes. To address these limitations, A^{2}PSR accounts for both the demands that regional structures place on spatial representations and the influence of target scale on rendering. It combines anchored adaptive probes with scale-aware rendering to reconstruct atmospheric fields at arbitrary scales.

3 Method

3.1 Problem Definition

Atmospheric spatial super-resolution reconstructs an atmospheric field on a requested target grid from a coarse source field. Let $\mathbf{X} \in \mathbb{R}^{C\times H_s\times W_s}$ and $\mathbf{Y} \in \mathbb{R}^{C\times H_t\times W_t}$ denote the source and target fields containing the same C atmospheric variables. The goal is to learn a single model f_θ that predicts the atmospheric field at any requested target resolution (H_t, W_t):

$$\widehat{\mathbf{Y}} = f_\theta(\mathbf{X}, H_t, W_t) \in \mathbb{R}^{C\times H_t\times W_t}. \tag{1}$$

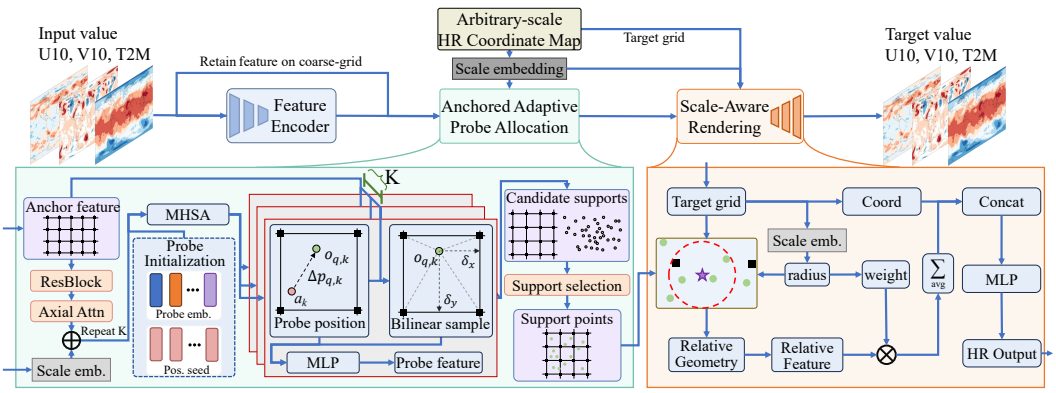

Figure 2: Overview of A^{2}PSR. The model constructs spatial support using coarse-grid anchors and adaptive probes, and aggregates features through scale-aware rendering to reconstruct atmospheric fields at arbitrary target resolutions.

We train A^{2}PSR by minimizing the mean absolute reconstruction error for each pair $(\mathbf{X}, \mathbf{Y})$:

$$\mathcal{L}_{\text{rec}} = \frac{1}{CH_tW_t} \sum_{c=1}^{C} \sum_{h=1}^{H_t} \sum_{w=1}^{W_t} \left| \widehat{Y}_{c,h,w} - Y_{c,h,w} \right|. \tag{2}$$

During training, different resolutions are sampled within the training range, allowing one parameter set to learn reconstruction across scales rather than a mapping tied to a single resolution pair.

3.2 Overall Framework

As illustrated in Fig. 2, A^{2}PSR consists of atmospheric feature encoding, anchored adaptive probe allocation, and scale-aware rendering. Given a source atmospheric field, the encoder first jointly extracts multivariate features and incorporates latitude–longitude coordinates to construct shared spatial representations. These representations are combined with global regional context and the requested scale to estimate where additional latent support is needed. A fixed probe budget is adaptively allocated to candidates generated inside source cells, while the observed source positions are retained as anchored support. Finally, the rendering module aggregates features from anchors and probes at target locations using a scale-dependent support range.

3.3 Anchored Adaptive Probe Allocation

Atmospheric fields exhibit spatially nonuniform gains in resolved detail as resolution increases, indicating that the demand for information-bearing locations varies across regions. Accordingly, we design an adaptive probe allocation module with anchored support. The module retains coarse-grid points as stable anchors and jointly considers local structures, cross-region dependencies, and the target scale to adaptively allocate additional probes within coarse-grid cells, providing information-bearing locations that account for spatial heterogeneity in local detail recovery.

Specifically, given a coarse-resolution input field $\mathbf{X} \in \mathbb{R}^{C \times H_s \times W_s}$, we first employ an EDSR Lim et al. (2017) backbone without upsampling to jointly extract multivariate features. We then incorporate normalized latitude–longitude coordinates for spatial encoding, producing a shared spatial feature map $\mathbf{F} \in \mathbb{R}^{D \times H_s \times W_s}$, where C is the number of input atmospheric variables and D is the number of feature channels. We then concatenate the shared spatial features $\mathbf{F}$ and the original input $\mathbf{X}$ along the channel dimension and map the result to anchor features, retaining all $H_s \times W_s$ grid points as anchors. Four adjacent anchors enclose a grid cell q, within which candidate probes are generated. With periodic connectivity along the longitudinal direction, the source grid contains $Q = (H_s - 1)W_s$ such cells. To represent the target scale shared by all regions, we define a two-dimensional scale vector from the latitudinal and longitudinal log-resolution ratios:

$$\boldsymbol{\ell} = \left[\log_2 \frac{H_t - 1}{H_s - 1}, \log_2 \frac{W_t}{W_s} \right]. \tag{3}$$

To characterize the refinement demand of each cell at the target scale, the module concatenates the shared spatial features at its four corners and encodes them into a local region representation $\mathbf{r}_q$. It

then models cross-region dependencies using axial self-attention, fuses global context, and adds the target-scale encoding to obtain the region representation $\mathbf{z}_q$:

$$\mathbf{r}_q = \mathrm{MLP}_{\mathrm{reg}}\left(\left[\mathbf{F}_q^{\mathrm{tl}}, \mathbf{F}_q^{\mathrm{tr}}, \mathbf{F}_q^{\mathrm{bl}}, \mathbf{F}_q^{\mathrm{br}}\right]\right), \qquad \{\mathbf{z}_q\}_{q=1}^{Q} = \mathrm{AxialAttn}\left(\{\mathbf{r}_q\}_{q=1}^{Q}\right) + \mathrm{MLP}_{\mathrm{sc}}(\boldsymbol{\ell}). \quad (4)$$

Here, $[\cdot]$ denotes feature concatenation, and $\mathbf{F}_q^{\mathrm{tl}}$, $\mathbf{F}_q^{\mathrm{tr}}$, $\mathbf{F}_q^{\mathrm{bl}}$, and $\mathbf{F}_q^{\mathrm{br}}$ are the shared spatial features at the top-left, top-right, bottom-left, and bottom-right corners of cell q, respectively. $\mathrm{MLP}_{\mathrm{reg}}$ maps the concatenated corner features to a local region representation, $\mathrm{AxialAttn}$ denotes cross-region encoding that combines axial self-attention with global context fusion, and $\mathrm{MLP}_{\mathrm{sc}}$ maps the scale vector $\boldsymbol{\ell}$ to the region feature space. Accordingly, $\mathbf{r}_q$ is the local feature of cell q, and $\mathbf{z}_q$ is its region representation enriched with cross-region dependencies and target-scale information. To adapt probes to spatially varying refinement demands, the module projects $\mathbf{z}_q$ into the candidate feature space and replicates it K times, once for each candidate in the cell. Here, $\mathbf{e}_k$ and $\mathbf{a}_k$ are the learnable probe bias and initial positional parameter of the k-th candidate, respectively. These parameters are randomly initialized separately for each candidate, providing distinct initial representations for the K candidates. The module then adds $\mathbf{e}_k$ and the encoding of $\mathbf{a}_k$ to each copy to obtain the candidate representations:

$$\mathbf{T}_q = \left[W_z \mathbf{z}_q + \mathbf{e}_k + W_a \sigma(\mathbf{a}_k)\right]_{k=1}^{K}, \qquad \mathbf{T}_q \leftarrow \mathbf{T}_q + \mathrm{MHSA}\left(\mathrm{LN}(\mathbf{T}_q)\right). \quad (5)$$

Here, W_z and W_a map the region representation and the initial relative position into the candidate feature space, respectively, and $\mathbf{T}_q$ collects the K candidate representations in cell q. σ denotes the sigmoid function. The module then models relations among candidates within the same grid cell using multi-head self-attention and updates $\mathbf{T}_q$ through a feed-forward network. The module separately scores the region representation $\mathbf{z}_q$ and the candidate representation $\mathbf{T}_{q,k}$, centers the scores across regions and across candidates within each region, respectively, and sums the two terms to obtain the allocation score $s_{q,k}$ for candidate selection:

$$s_{q,k} = \mathcal{S}_{\mathrm{reg}}(\mathbf{z}_q) - \frac{1}{Q}\sum_{i=1}^{Q} \mathcal{S}_{\mathrm{reg}}(\mathbf{z}_i) + \mathcal{S}_{\mathrm{cand}}(\mathbf{T}_{q,k}) - \frac{1}{K}\sum_{j=1}^{K} \mathcal{S}_{\mathrm{cand}}(\mathbf{T}_{q,j}). \quad (6)$$

Here, $\mathcal{S}_{\mathrm{reg}}$ and $\mathcal{S}_{\mathrm{cand}}$ are lightweight fully connected networks that output scalar region and candidate scores, respectively. The score is also used to update $\mathbf{T}_{q,k}$ for predicting the candidate's normalized relative coordinate within cell q:

$$\Delta\mathbf{p}_{q,k} = \mathrm{MLP}_{\mathrm{pos}}(\mathbf{T}_{q,k}), \qquad \mathbf{o}_{q,k} = \sigma\left(\mathbf{a}_k + \Delta\mathbf{p}_{q,k}\right). \quad (7)$$

Here, $\Delta\mathbf{p}_{q,k}$ denotes the predicted positional offset, and $\mathbf{o}_{q,k}$ denotes the normalized relative coordinate of the candidate probe within cell q. Through this representation update path, the scoring module can be optimized using the reconstruction loss even with discrete candidate selection. The module maps this relative coordinate to the cell's spatial extent to obtain the probe coordinate and bilinearly samples the shared spatial features $\mathbf{F}$ and the original input $\mathbf{X}$ there. It then fuses the sampled values with the updated candidate representation $\mathbf{T}_{q,k}$ and the relative coordinate $\mathbf{o}_{q,k}$ to generate the candidate probe feature. Implementation details of candidate representation updates, positional constraints, and candidate selection are provided in Appendix B. When the budget permits, the highest-scoring candidate in each grid cell is retained first, after which the remaining probes are selected from all unselected candidates in descending score order until the budget is reached. If the budget is insufficient for one candidate per cell, probes are selected from all candidates by descending score up to the budget.

Overall, the module preserves a stable observational basis through regular anchors and uses additional adaptive probes to carry supplementary information for local detail recovery. By concentrating additional probes in regions with greater refinement demands, it provides spatial support that combines stable coverage with content adaptivity for subsequent rendering.

3.4 Scale-Aware Rendering

As the target scale changes, the sampling locations on the target grid and their spatial relations to anchors and probes also change. A fixed spatial support range may therefore limit the effective use of information carried by these support points. To this end, we design a scale-aware rendering module. The module adjusts the shared spatial support range of anchors and probes according to the

target scale and aggregates information using relative positions, support confidence, and distance decay to produce regular-grid features for target-field reconstruction.

The module applies elementwise bounded normalization to the scale vector $\boldsymbol{\ell}$ defined above and uses the result to predict a rendering radius ρ shared by all anchors and probes at the current target scale:

$$\rho = \rho_0 \exp\left(\mathcal{R}\left(\frac{\boldsymbol{\ell}}{1+|\boldsymbol{\ell}|}\right)\right). \tag{8}$$

Here, $\mathcal{R}$ is a learnable radius predictor implemented as a multilayer perceptron, ρ_0 is the reference rendering radius in the normalized coordinate space, and the exponential mapping ensures a positive predicted rendering radius. For each target location j, the module uses the spatial distances d_{ij} to select support points i within the rendering radius ρ from all anchors and retained probes, forming the effective support set $\mathcal{N}(j)$. The feature of support point i is denoted by $\mathbf{h}_i$. For each $i \in \mathcal{N}(j)$, the module normalizes the displacement from support point i to target location j by the latitudinal and longitudinal source-grid spacings to obtain $\Delta_{ij}^{\mathrm{lat}}$ and $\Delta_{ij}^{\mathrm{lon}}$. The module linearly maps the feature $\mathbf{h}_i$ of support point i into four feature components, $\mathbf{v}_i^{\mathrm{base}}$, $\mathbf{v}_i^{\mathrm{lat}}$, $\mathbf{v}_i^{\mathrm{lon}}$, and $\mathbf{v}_i^{\mathrm{rad}}$, and combines them with the relative displacement to construct the relation-aware message sent to target location j:

$$\mathbf{m}_{ij} = \mathbf{v}_i^{\mathrm{base}} + \Delta_{ij}^{\mathrm{lat}}\mathbf{v}_i^{\mathrm{lat}} + \Delta_{ij}^{\mathrm{lon}}\mathbf{v}_i^{\mathrm{lon}} + \left[\left(\Delta_{ij}^{\mathrm{lat}}\right)^2 + \left(\Delta_{ij}^{\mathrm{lon}}\right)^2\right]\mathbf{v}_i^{\mathrm{rad}}. \tag{9}$$

Here, $\mathbf{m}_{ij}$ denotes the relation-aware message sent from support point i to target location j. The base term provides information independent of relative displacement, while the latitudinal and longitudinal terms capture the effects of displacement along their respective directions. The radial term further incorporates the effect of the squared normalized relative distance. Accordingly, the same support point can transmit different information according to its spatial relations to different target locations. To account for variations in both support-point reliability and spatial distance, the module combines support confidence with distance decay to compute normalized contribution weights within the effective support set:

$$c_i = \begin{cases} 1, & i \text{ is an anchor,} \\ 2\sigma(\mathcal{C}(\mathbf{h}_i)), & i \text{ is an adaptive probe,} \end{cases} \qquad \alpha_{ij} = \frac{c_i\left(1 - d_{ij}^2/\rho^2\right)^2}{\sum_{k\in\mathcal{N}(j)} c_k\left(1 - d_{kj}^2/\rho^2\right)^2}. \tag{10}$$

Here, anchor confidence is fixed to 1, whereas the confidence of an adaptive probe is predicted from its feature representation $\mathbf{h}_i$. The function $\mathcal{C}$ is a multilayer perceptron with a scalar output, σ denotes the sigmoid function, and α_{ij} is the normalized contribution weight of support point i to target location j. The relation-aware messages are then aggregated using these weights to obtain the rendered feature at target location j:

$$\mathbf{f}_j^{\mathrm{ren}} = \sum_{i\in\mathcal{N}(j)} \alpha_{ij}\mathbf{m}_{ij}. \tag{11}$$

Finally, the rendered feature is passed to a pointwise decoder with the target coordinate, target-cell size, and support-coverage state to predict the atmospheric variables at each target location. All predictions are assembled into a regular grid atmospheric field at the requested target resolution.

Overall, the module adjusts the spatial support range of information carried by anchors and probes according to the target scale and modulates their contributions based on relative positions and support confidence. By aggregating information within the effective support range at each target location, it reconstructs atmospheric fields at different resolutions.

4 Experiments

4.1 Experimental Setup

Dataset and scales. We conduct experiments on the ERA5 reanalysis dataset Hersbach et al. (2020) using 2-m temperature (T2M) and 10-m zonal and meridional winds (U10 and V10). Hourly data from 2008–2018, 2019, and 2020 are used for training, validation, and testing, respectively. Following WeatherBench 2 Rasp et al. (2024), we generate source and target fields from native 0.25° ERA5 via first-order conservative spherical regridding:

$$Z_{c,i}^{H,W} = \frac{\sum_j a_{ij} Z_{c,j}^0}{\sum_j a_{ij}}, \tag{12}$$

Table 1: Lat-RMSE on ERA5 at $2\times$, $4\times$, $8\times$, and $22.5\times$. Lower is better. Bold and underline mark the best and second-best arbitrary-scale results. Per-scale denotes separate training per scale. U-Net is a reference. † denotes an adapted configuration described in Appendix E.

Method	Per-scale	T2M				U10				V10			
		$2\times$	$4\times$	$8\times$	$22.5\times$	$2\times$	$4\times$	$8\times$	$22.5\times$	$2\times$	$4\times$	$8\times$	$22.5\times$
Bicubic	No	1.566	2.169	2.477	2.685	1.370	1.833	2.057	2.195	1.476	2.053	2.333	2.477
Bilinear	No	1.698	2.254	2.537	2.728	1.528	1.941	2.140	2.262	1.586	2.093	2.340	2.494
U-Net (2015)	Yes	0.574	0.744	0.856	1.032	0.700	0.889	0.990	1.114	0.644	0.842	0.946	1.056
Meta-SR (2019)	No	0.513	0.969	1.285	1.527	0.625	0.949	1.163	1.306	0.591	0.912	1.117	1.258
LIIF (2021)	No	0.531	0.964	1.297	1.550	0.635	0.980	1.261	1.471	0.598	0.922	1.152	1.320
SRNO (2023)	No	0.538	0.906	1.201	1.445	0.642	0.934	1.134	1.279	0.607	0.899	1.101	1.251
CiaoSR (2023)	No	0.531	0.924	1.169	<u>1.391</u>	0.640	0.969	<u>1.131</u>	1.280	0.605	0.916	<u>1.084</u>	1.225
HIIF (2025)	No	0.512	1.120	1.532	1.815	0.627	1.060	1.353	1.542	0.592	1.126	1.488	1.721
GSASR (2025)	No	0.513	<u>0.869</u>	<u>1.067</u>	1.417	0.635	<u>0.921</u>	1.171	<u>1.276</u>	0.594	<u>0.875</u>	1.125	<u>1.209</u>
MINet (2026)	No	**0.472**	0.893	1.253	1.526	<u>0.615</u>	<u>0.921</u>	1.170	1.352	<u>0.586</u>	0.898	1.180	1.413
GSSA-ViT† (2026)	No	1.364	1.914	2.180	2.339	1.156	1.659	1.886	2.007	1.104	1.628	1.885	2.030
A^{2}PSR (ours)	No	<u>0.495</u>	**0.814**	**1.045**	**1.248**	**0.611**	**0.884**	**1.064**	**1.196**	**0.574**	**0.844**	**1.019**	**1.153**

where Z^0 denotes the native field, $Z^{H,W}$ its regridded counterpart, and a_{ij} the spherical overlap area between target cell i and native cell j for channel c. The source grid is fixed at 33×64 (approximately $5.625°$ spacing), with training target resolutions spanning $1\times$–$2\times$ scales. We evaluate at $2\times$, $4\times$, $8\times$, and $22.5\times$ scales. The latter three assess extrapolation to unseen resolutions, with $22.5\times$ recovering the native ERA5 grid. Data normalization is detailed in Appendix A.

Baselines and metrics. We compare A^{2}PSR with bilinear interpolation, bicubic interpolation, and the end-to-end fixed-scale baseline U-Net Ronneberger et al. (2015). We also compare with eight arbitrary-scale baselines (Table 1). U-Net is trained separately at each evaluation resolution as a fixed-scale reference. Comparisons of generalization to unseen scales are restricted to arbitrary-scale methods. We report latitude-weighted root mean square error (Lat-RMSE) for each atmospheric variable in physical units:

$$\omega_h = \frac{\sin\phi_h^+ - \sin\phi_h^-}{\frac{1}{H_t}\sum_{r=1}^{H_t}(\sin\phi_r^+ - \sin\phi_r^-)}, \quad \text{Lat-RMSE}_c = \frac{1}{N}\sum_{n=1}^{N}\left[\frac{1}{H_t W_t}\sum_{h=1}^{H_t}\sum_{w=1}^{W_t}\omega_h\left(e_{c,h,w}^{(n)}\right)^2\right]^{1/2}. \quad (13)$$

Here, N is the number of test fields, $e_{c,h,w}^{(n)}$ is the prediction error, and ϕ_h^- and ϕ_h^+ are the latitude bounds of row h. The latitude weight ω_h is proportional to the spherical grid-cell area, accounting for area differences across latitudes. Supplementary evaluation using MAE, SSIM, and wind-speed FSS is provided in Appendix F, along with metric definitions and evaluation settings.

Implementation details. All arbitrary-scale methods uniformly sample target resolutions over $1\times$–$2\times$ during training and use a single model to directly evaluate all scales. U-Net is trained separately at each evaluation resolution. All learnable models are trained for 20 epochs with a global batch size of 8 and converge within this training budget. All comparisons and ablations use the same random seed (3407). The additional probe budget for A^{2}PSR is set to $M = 2{,}500$. Model configurations, implementation details, and training settings (optimizer, loss function, and learning-rate schedule) are provided in Appendices E and B. All experiments use PyTorch. Training stability and computational costs are reported in Appendices D and H.

4.2 Comparison with Other Methods

Table 1 reports reconstruction errors at $2\times$, covered during training, and at unseen scales of $4\times$, $8\times$, and $22.5\times$. U-Net is trained independently at each target scale and included only as a reference. At $2\times$, A^{2}PSR achieves the best U10 and V10 results and the second-best T2M result among arbitrary-scale methods. At the remaining scales, A^{2}PSR yields lower errors for every variable than all other arbitrary-scale methods. Its wind-field errors at $4\times$ are comparable to those of U-Net supervised

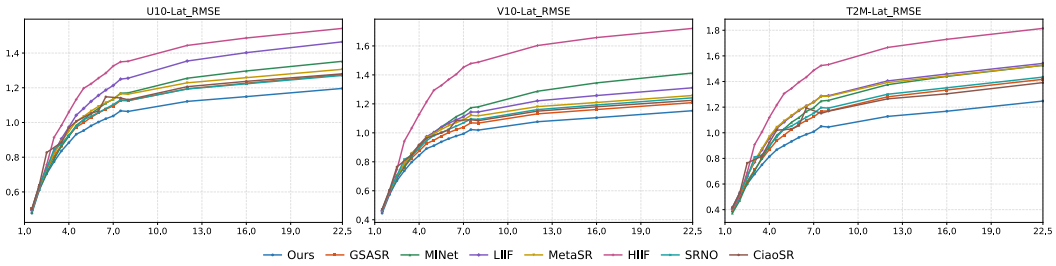

Figure 3: Lat-RMSE trends on ERA5 from $1.5\times$ to $22.5\times$.

Table 2: Ablation results for stable anchors, adaptive probes, and scale-aware rendering. Lat-RMSE is reported at each evaluation scale. Lower is better. Bold and underline indicate the best and second-best results in each column, respectively.

Anchors	Probes	Rendering	T2M				U10				V10			
			$2\times$	$4\times$	$8\times$	$22.5\times$	$2\times$	$4\times$	$8\times$	$22.5\times$	$2\times$	$4\times$	$8\times$	$22.5\times$
Yes	None	Scale-aware	0.521	0.860	1.127	1.365	0.626	0.919	1.127	1.291	0.589	0.874	1.068	1.217
Yes	None	Fixed	0.597	0.984	1.195	1.350	0.681	1.001	1.166	1.266	0.640	0.951	1.110	1.208
Yes	None	Local implicit	0.522	0.921	1.215	1.441	0.619	0.936	1.165	1.324	0.585	0.893	1.100	1.241
Yes	Random	Scale-aware	0.559	0.905	1.158	1.380	0.648	0.944	1.135	1.279	0.612	0.901	1.085	1.225
Yes	Adaptive	Local implicit	0.546	0.891	1.115	<u>1.295</u>	0.634	0.902	1.075	1.200	0.591	0.860	1.028	<u>1.149</u>
Yes	Adaptive	Fixed	<u>0.508</u>	<u>0.850</u>	<u>1.099</u>	1.316	<u>0.617</u>	0.908	1.101	1.243	<u>0.578</u>	0.857	1.031	1.157
No	Adaptive	Scale-aware	0.525	0.860	1.102	1.318	0.619	<u>0.892</u>	<u>1.066</u>	**1.192**	0.583	<u>0.849</u>	**1.013**	**1.132**
Yes	**Adaptive**	**Scale-aware**	**0.495**	**0.814**	**1.045**	**1.248**	**0.611**	**0.884**	**1.064**	<u>1.196</u>	**0.574**	**0.844**	<u>1.019</u>	1.153

at that scale, demonstrating strong generalization to unseen scales. Figure 3 compares Lat-RMSE trends across arbitrary-scale super-resolution methods from $1.5\times$ to $8\times$ in increments of $0.5\times$, and at $12\times$, $16\times$, and $22.5\times$, illustrating reconstruction performance across target scales.

4.3 Ablation Studies

Table 2 examines the roles of stable anchors, adaptive probes, and scale-aware rendering. All configurations with additional probes use a fixed budget of $2{,}500$ probes. We remove anchors, omit probes, or replace adaptive probes with random probes, and replace the proposed rendering approach with fixed-radius rendering or LIIF-based local implicit decoding Chen et al. (2021). The fixed-radius setting uses the reference radius ρ_0 to evaluate the contribution of scale-adaptive support ranges. Implementation details of local implicit decoding are provided in Appendix C. The results show that combining stable anchors, adaptive probes, and scale-aware rendering improves reconstruction. The full model performs best for most variable-scale combinations, while some ablations perform slightly better on wind variables at higher scales.

4.4 Effect of Probe Number

We vary only the adaptive-probe number M to evaluate each atmospheric variable's reconstruction accuracy. We evaluate five probe budgets, $M \in \{1{,}000, 2{,}048, 2{,}500, 5{,}000, 10{,}000\}$, at the training-boundary scale of $2\times$ and the unseen extrapolation scales of $4\times$, $8\times$, and $22.5\times$. As shown in Table 3, increasing the probe number from $1{,}000$ to $2{,}500$ clearly reduces reconstruction errors across all four evaluated scales, indicating that a sufficient number of adaptive probes can effectively supplement the representational capacity of stable anchors. For $M \geq 2{,}500$, the results across probe budgets are generally close. Further increasing the probe number does not yield consistent performance gains. Under the experimental settings, a moderate number of probes already achieves near-optimal reconstruction accuracy among the evaluated budgets.

Table 3: Effect of the adaptive-probe number on Lat-RMSE for T2M, U10, and V10 at each evaluation scale. Lower is better. Bold and underline indicate the best and second-best results in each column.

Number of probes	T2M				U10				V10			
	2×	4×	8×	22.5×	2×	4×	8×	22.5×	2×	4×	8×	22.5×
1,000	0.553	0.900	1.144	1.362	0.653	0.945	1.128	1.266	0.614	0.899	1.074	1.205
2,048	<u>0.511</u>	<u>0.837</u>	1.070	1.270	0.622	0.899	1.084	1.225	<u>0.584</u>	0.856	1.034	1.171
2,500	**0.495**	**0.814**	**1.045**	<u>1.248</u>	**0.611**	**0.884**	<u>1.064</u>	<u>1.196</u>	**0.574**	**0.844**	<u>1.019</u>	1.153
5,000	0.516	0.841	1.063	1.251	<u>0.620</u>	0.897	1.073	1.201	<u>0.584</u>	<u>0.854</u>	<u>1.019</u>	<u>1.138</u>
10,000	0.519	0.840	<u>1.058</u>	**1.240**	0.624	<u>0.894</u>	**1.063**	**1.179**	0.589	0.855	**1.016**	**1.130**

4.5 Case Study

As shown in Figure 4, we use Hurricane Teddy (2020) to illustrate probe allocation during LR-to-HR reconstruction. The model adaptively allocates auxiliary probes to provide additional reference information in the storm region and areas with complex wind-field structures, supporting the recovery of local details. On the 2020 test set, detail demand, measured by the mean area-weighted interpolation MSE across three standardized variables, has a mean per-sample Spearman correlation of 0.1396 with probe counts. Excluding guaranteed probes, the highest demand quartile receives 10.02 times as many probes per cell as the lowest, supporting this allocation tendency. Details are provided in the Appendix G. We further compare wind-speed reconstructions derived from U10 and V10 for Typhoon Haishen. Our method more closely matches the ground truth (GT) in the morphology of the eye and the surrounding high-wind regions. Additionally, to evaluate spatial matching of strong-wind regions under extreme-cyclone conditions, we compute FSS on 2020 IBTrACS track samples. Detailed settings and results are provided in Appendix F.2.

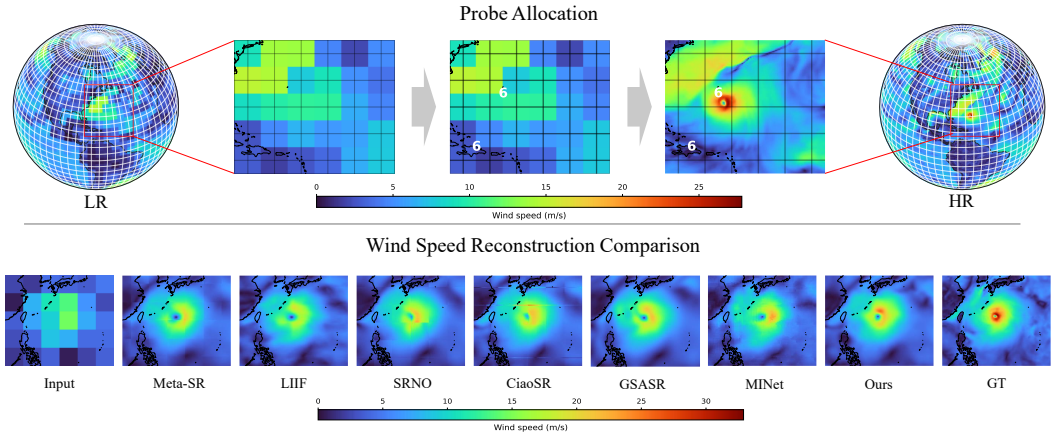

Figure 4: Tropical cyclone cases at 22.5×. The top row shows probe allocation for Hurricane Teddy (2020-09-21 04:00 UTC), with HR denoting ground truth. Numbers indicate auxiliary probes per cell. Unmarked cells contain at most one. The bottom row compares wind-speed reconstructions for Typhoon Haishen (2020-09-05 00:00 UTC).

5 Conclusion

We propose A^{2}PSR, which combines stable anchors, adaptive probes, and scale-aware rendering for regionally adaptive atmospheric spatial super-resolution at arbitrary scales. Experiments on ERA5 show competitive performance within the training scale range and superior performance over all evaluated arbitrary-scale baselines at the tested extrapolation scales. Future work will extend evaluation to more variables and regional high-resolution data.

AI USE STATEMENT

In this work, we used generative AI tools solely for translating and polishing the manuscript. These tools were not used for any other research tasks. The authors have reviewed all AI-assisted text for accuracy and consistency with the intended meaning. The authors take full responsibility for the final content of this work.

REFERENCES

Mario C. Acosta, Sergi Palomas, Stella V. Paronuzzi Ticco, Gladys Utrera, Joachim Biercamp, Pierre-Antoine Bretonniere, Reinhard Budich, Miguel Castrillo, Arnaud Caubel, Francisco Doblas-Reyes, et al. The Computational and Energy Cost of Simulation and Storage for Climate Science: Lessons from CMIP6. *Geoscientific Model Development*, 17:3081–3098, 2024. doi: 10.5194/gmd-17-3081-2024.

Jiezhang Cao, Qin Wang, Yongqin Xian, Yawei Li, Bingbing Ni, Zhiming Pi, Kai Zhang, Yulun Zhang, Radu Timofte, and Luc Van Gool. CiaoSR: Continuous Implicit Attention-in-Attention Network for Arbitrary-Scale Image Super-Resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1796–1807, June 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Cao_CiaoSR_Continuous_Implicit_Attention-in-Attention_Network_for_Arbitrary-Scale_Image_Super-Resolution_CVPR_2023_paper.html.

Du Chen, Liyi Chen, Zhengqiang Zhang, and Lei Zhang. Generalized and Efficient 2D Gaussian Splatting for Arbitrary-Scale Super-Resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 26435–26445, October 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Chen_Generalized_and_Efficient_2D_Gaussian_Splatting_for_Arbitrary-scale_Super-Resolution_ICCV_2025_paper.html.

Teng-Yue Chen, Jie-Lan Xie, Wei Zhou, Jian-Fang Hu, Peng-Qin Yao, Tian-Ming Liang, Wei-Shi Zheng, and Pak-Wai Chan. Arbitrary-Scale Atmospheric Downscaling with Mixture of Implicit Neural Networks Trained on Fixed-Scale Data. *Pattern Recognition*, 173:112802, 2026. doi: 10.1016/j.patcog.2025.112802.

Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning Continuous Image Representation with Local Implicit Image Function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8628–8638, June 2021. URL https://openaccess.thecvf.com/content/CVPR2021/html/Chen_Learning_Continuous_Image_Representation_With_Local_Implicit_Image_Function_CVPR_2021_paper.html.

Ronald Gelaro, Will McCarty, Max J. Suárez, Ricardo Todling, Andrea Molod, Lawrence Takacs, Cynthia A. Randles, Anton Darmenov, Michael G. Bosilovich, Rolf Reichle, et al. The Modern-Era Retrospective Analysis for Research and Applications, Version 2 (MERRA-2). *Journal of Climate*, 30(14):5419–5454, 2017. doi: 10.1175/JCLI-D-16-0758.1.

Tao Han, Zhibin Wen, Zhenghao Chen, Fenghua Lin, Junyu Gao, Song Guo, and Lei Bai. Generative 3D Gaussian Splatting for Arbitrary-Resolution Atmospheric Downscaling and Forecasting. *arXiv preprint arXiv:2604.07928*, 2026. doi: 10.48550/arXiv.2604.07928. URL https://arxiv.org/abs/2604.07928.

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. *Quarterly Journal of the Royal Meteorological Society*, 146(730):1999–2049, 2020. doi: 10.1002/qj.3803.

Kevin Höhlein, Michael Kern, Timothy Hewson, and Rüdiger Westermann. A Comparative Study of Convolutional Neural Network Models for Wind Field Downscaling. *Meteorological Applications*, 27(6):e1961, 2020. doi: 10.1002/met.1961.

Jintong Hu, Bin Xia, Bin Chen, Wenming Yang, and Lei Zhang. GaussianSR: High Fidelity 2D Gaussian Splatting for Arbitrary-Scale Image Super-Resolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 3554–3562, 2025. doi: 10.1609/aaai.v39i4.32369. URL https://ojs.aaai.org/index.php/AAAI/article/view/32369.

Xuecai Hu, Haoyuan Mu, Xiangyu Zhang, Zilei Wang, Tieniu Tan, and Jian Sun. Meta-SR: A Magnification-Arbitrary Network for Super-Resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1575–1584, June 2019. URL https://openaccess.thecvf.com/content_CVPR_2019/html/Hu_Meta-SR_A_Magnification-Arbitrary_Network_for_Super-Resolution_CVPR_2019_paper.html.

Yuxuan Jiang, Ho Man Kwan, Tianhao Peng, Ge Gao, Fan Zhang, Xiaoqing Zhu, Joel Sole, and David Bull. HIIF: Hierarchical Encoding based Implicit Image Function for Continuous Super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2289–2299, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Jiang_HIIF_Hierarchical_Encoding_based_Implicit_Image_Function_for_Continuous_Super-resolution_CVPR_2025_paper.html.

Zhengrui Jin, Fei Ge, Quanliang Chen, and Zhiye Lin. To What Extent Horizontal Resolution Improves the Simulation of Precipitation in CMIP6 HighResMIP Models over Southwest China? *Frontiers in Earth Science*, 10:1003748, 2023. doi: 10.3389/feart.2022.1003748.

Bo Li, Zijian Zhu, Xiaohui Zhong, Ruxin Tan, Yegui Wang, Weiren Lan, and Hao Li. One-kilometer resolution forecasts of hourly precipitation over China using machine learning models. *Atmospheric Science Letters*, 26(3):e1297, 2025. doi: 10.1002/asl.1297.

Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced Deep Residual Networks for Single Image Super-Resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 136–144, 2017. URL https://openaccess.thecvf.com/content_cvpr_2017_workshops/w12/html/Lim_Enhanced_Deep_Residual_CVPR_2017_paper.html.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10012–10022, October 2021. URL https://openaccess.thecvf.com/content/ICCV2021/html/Liu_Swin_Transformer_Hierarchical_Vision_Transformer_Using_Shifted_Windows_ICCV_2021_paper.html.

Long Peng, Anran Wu, Wenbo Li, Peizhe Xia, Xueyuan Dai, Xinjie Zhang, Xin Di, Haoze Sun, Renjing Pei, Yang Wang, Yang Cao, and Zheng-Jun Zha. Pixel to Gaussian: Ultra-Fast Continuous Super-Resolution with 2D Gaussian Modeling. *arXiv preprint arXiv:2503.06617*, 2025. URL https://arxiv.org/abs/2503.06617.

Antonio Pérez, Mario Santa Cruz, Daniel San Martín, and José Manuel Gutiérrez. Transformer based super-resolution downscaling for regional reanalysis: Full domain vs tiling approaches. *arXiv preprint arXiv:2410.12728*, 2024. doi: 10.48550/arXiv.2410.12728. URL https://arxiv.org/abs/2410.12728.

Stephan Rasp, Stephan Hoyer, Alexander Merose, Ian Langmore, Peter Battaglia, Tyler Russell, Alvaro Sanchez-Gonzalez, Vivian Yang, Rob Carver, Shreya Agrawal, Matthew Chantry, Zied Ben Bouallegue, Peter Dueben, Carla Bromberg, Jared Sisk, Luke Barrington, Aaron Bell, and Fei Sha. WeatherBench 2: A Benchmark for the Next Generation of Data-Driven Global Weather Models. *Journal of Advances in Modeling Earth Systems*, 16(6):e2023MS004019, 2024. doi: 10.1029/2023MS004019.

Malcolm J. Roberts, Kevin A. Reed, Qing Bao, Joseph J. Barsugli, Suzana J. Camargo, Louis-Philippe Caron, Ping Chang, Cheng-Ta Chen, Hannah M. Christensen, Gokhan Danabasoglu, et al. High-Resolution Model Intercomparison Project Phase 2 (HighResMIP2) towards CMIP7. *Geoscientific Model Development*, 18:1307–1332, 2025. doi: 10.5194/gmd-18-1307-2025.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention*, volume 9351 of *Lecture Notes in Computer Science*, pp. 234–241. Springer, 2015. doi: 10.1007/978-3-319-24574-4_28.

Yingkai Sha, David John Gagne II, Gregory West, and Roland Stull. Deep-Learning-Based Gridded Downscaling of Surface Meteorological Variables in Complex Terrain. Part I: Daily Maximum and Minimum 2-m Temperature. *Journal of Applied Meteorology and Climatology*, 59(12):2057–2073, 2020. doi: 10.1175/JAMC-D-20-0057.1.

Kenta Shiraishi, Yuka Muto, and Shunji Kotsuki. Downscaling Spatial Patterns of Precipitation Data By Deep-learning-based Super Resolution. *Artificial Intelligence and Data Science*, 4(3): 515–521, 2023. doi: 10.11532/jsceiii.4.3_515.

Min Wei and Xuesong Zhang. Super-Resolution Neural Operator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18247–18256, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Wei_Super-Resolution_Neural_Operator_CVPR_2023_paper.html.

Chang Xu, Gencer Sumbul, Li Mi, Robin Zbinden, and Devis Tuia. GeoFAR: Geography-Informed Frequency-Aware Super-Resolution for Climate Data. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=0WHpOekph0.

A Data Normalization

Each variable c is normalized as $\widetilde{Z}_c = (Z_c - \mu_c)/\sigma_c$, where the mean μ_c and standard deviation σ_c are computed from the 2008–2018 training data. The same statistics are used for source and target fields across data splits and resolutions.

B A^{2}PSR Implementation Details

After within-cell multi-head self-attention and feed-forward updates, the region representation is projected by a 1×1 convolution and the candidate representation is projected by an MLP to obtain region-level and candidate-level allocation codes. These codes are concatenated with the allocation score $s_{q,k}$ and passed through an update MLP to update the candidate representation by residual addition:

$$\mathbf{T}_{q,k} \leftarrow \mathbf{T}_{q,k} + \mathrm{MLP}_{\mathrm{upd}}([\mathrm{Conv}_{1\times1}(\mathbf{z}_q), \mathrm{MLP}_{\mathrm{cand}}(\mathbf{T}_{q,k}), s_{q,k}]) \,. \tag{14}$$

The updated candidate representation is used to predict candidate positions, features, and rendering confidence. The positional-head output is multiplied by a fixed factor of 0.5, added to the initial positional parameter $\mathbf{a}_k$, and passed through a sigmoid function. The resulting coordinate components are then mapped as

$$\mathbf{o}_{q,k} = 0.01 + 0.98\,\sigma(\mathbf{a}_k + 0.5\ \mathrm{MLP}_{\mathrm{pos}}(\mathbf{T}_{q,k})) \,. \tag{15}$$

This keeps both within-cell coordinate components in $(0.01, 0.99)$, enforcing the listed position margin. Although the discrete top-k selection and candidate deduplication do not propagate gradients through their indices, selected candidates retain the candidate-representation update path. The reconstruction loss can therefore optimize the allocation-score module without an additional probe-allocation target. The complete model configuration is summarized in Table 4.

Table 4: Model configuration of A^{2}PSR.

Component	Configuration
Input and output channels	3
EDSR backbone	16 residual blocks, 64 channels; no upsampling
Residual scaling	1
Spatial, region, anchor, and probe feature dimension	256
Candidate representation dimension	64
Region and candidate allocation-code dimension	16
Global region attention	1 axial block, 4 heads per axis
Within-cell candidate attention	1 block, 4 heads
Candidates per source cell, K	50
Additional probe budget, M	2,500
Candidate position margin	0.01 in within-cell coordinates
Candidate separation within and across cells	0.03 in source-cell units
Reference support radius, ρ_0	0.08 in normalized coordinates
Radius-predictor hidden dimension	32
Output-decoder hidden layers	2 layers, width 256
Hidden-layer activation	SiLU

C Local Implicit Decoding Ablations

The local implicit ablation follows LIIF's local query approach and reconstructs each target value by combining predictions from four support points. Specifically, we select the four support points nearest to each target location from the combined set of anchors and adaptive probes, use a shared MLP to make separate predictions, and fuse them using confidence-modulated normalized inverse-distance weights. The weights are defined as

$$w_i = \frac{c_i / \max(\|\boldsymbol{\delta}_i\|_2, 10^{-6})}{\sum_{j\in\mathcal{N}_4} c_j / \max(\|\boldsymbol{\delta}_j\|_2, 10^{-6})}. \tag{16}$$

The four predictions are fused with these weights to obtain the reconstructed value at the target location.

D TRAINING STABILITY

This supplementary experiment assesses the training stability of A^{2}PSR using three independent random seeds, with all other settings fixed. Table 5 lists the evaluation entries for T2M, U10, and V10 at $2\times$, $4\times$, $8\times$, and $22.5\times$ on the 2020 test set.

Table 5: Lat-RMSE of A^{2}PSR across three independent training runs (seeds 3407, 9708, and 42). Lower is better. Mean and Std denote the mean and sample standard deviation across runs, respectively.

Run	T2M (K)				U10 ($\mathrm{m\,s^{-1}}$)				V10 ($\mathrm{m\,s^{-1}}$)			
	$2\times$	$4\times$	$8\times$	$22.5\times$	$2\times$	$4\times$	$8\times$	$22.5\times$	$2\times$	$4\times$	$8\times$	$22.5\times$
Seed 3407	0.495	0.814	1.045	1.248	0.611	0.884	1.064	1.196	0.574	0.844	1.019	1.153
Seed 9708	0.499	0.820	1.052	1.254	0.617	0.892	1.072	1.205	0.579	0.851	1.025	1.155
Seed 42	0.498	0.817	1.047	1.247	0.617	0.889	1.062	1.188	0.578	0.847	1.014	1.136
Mean	0.497	0.817	1.048	1.250	0.615	0.888	1.066	1.196	0.577	0.848	1.019	1.148
Std	0.002	0.003	0.003	0.004	0.004	0.004	0.005	0.009	0.003	0.004	0.005	0.010

E BASELINE IMPLEMENTATIONS

All learnable baselines use AdamW with $\beta_1 = 0.9$ and $\beta_2 = 0.95$. The weight decay is set to 0.1 for U-Net and 10^{-5} for the other baselines. The learning rate is linearly warmed up from 10^{-7} to 10^{-4} over the first 100 iterations and then decayed to 10^{-6} using a cosine schedule. All models compute the reconstruction loss in normalized space, with U-Net and GSSA-ViT using MSE and the other baselines using an L_1 loss. Table 6 lists the model-specific configurations and parameter counts.

Bilinear and bicubic interpolation. Both methods resample each atmospheric variable independently on the target grid using their respective two-dimensional interpolation kernels.

U-Net. U-Net Ronneberger et al. (2015) processes the bilinearly resized input using a convolutional encoder–decoder with skip connections and adds its predicted residual to the interpolated field.

Meta-SR. Meta-SR Hu et al. (2019) combines a residual dense network (RDN) with a position-to-weight network that predicts local upsampling weights from relative coordinates and scale information.

LIIF. LIIF Chen et al. (2021) decodes local EDSR features together with relative coordinates and target-cell sizes, combining neighboring predictions through local ensembling.

SRNO. SRNO Wei & Zhang (2023) combines neighboring EDSR features and coordinate information on the target grid and applies Galerkin attention operators to model interactions among query locations.

CiaoSR. CiaoSR Cao et al. (2023) aggregates features around each query through local implicit attention, with cross-scale nonlocal attention supplying additional context.

HIIF. HIIF Jiang et al. (2025) uses hierarchical coordinate encoding and residual MLPs to decode local features, together with linear attention across query locations.

GSASR. GSASR Chen et al. (2025) converts EDSR features into Gaussian representations, refines Gaussian embeddings with cross-attention and self-attention, and renders them onto the target grid. Because the atmospheric variables are standardized and may take signed values, we disable the sigmoid transformation on the Gaussian color channels during training and evaluation. The rendered fields therefore remain in normalized variable space and are de-normalized before metric computation.

MINet. MINet Chen et al. (2026) decodes multiresolution RDN features conditioned on target-cell sizes with a sparse mixture of experts, selecting two experts per query to jointly reconstruct the three variables.

Table 6: Model-specific hyperparameters and parameter counts (millions per model instance).

Method	Model-specific hyperparameters	Params (M)
Bilinear	No learnable parameters	0
Bicubic	No learnable parameters	0
U-Net	Base width: 64; channel multipliers: $(1, 2, 4, 8)$; encoder levels: 4; decoder stages: 3; two 3×3 convolutions with BatchNorm per stage; 2×2 transposed convolutions for upsampling	7.703
Meta-SR	RDN configuration A: 20 dense blocks, 6 layers per block, growth rate 32, feature width 64; position-to-weight network: one hidden layer of width 256; predicted kernel: 3×3	5.912
LIIF	EDSR: 16 residual blocks, width 128; decoder: 4 hidden layers of width 256; local ensemble, 3×3 feature unfolding, and cell decoding enabled	5.368
SRNO	EDSR: 16 residual blocks, width 64; Galerkin operator: 2 blocks, width 256, 16 heads; output projection hidden width: 256	2.015
CiaoSR	EDSR: 16 residual blocks, width 64; query, key, and value MLPs: 4 hidden layers of width 256; local size: 2; feature unfolding enabled; nonlocal attention scale: 2; local attention temperature: 1	2.649
HIIF	EDSR: 16 residual blocks, width 64; hierarchical decoder: 6 levels, branching factor 2; hidden width: 256; attention heads: 16; bilinear input residual enabled	2.955
GSASR	EDSR: 16 residual blocks, width 64; Gaussian embedding width: 180; attention heads: 6; cross-attention: 1 block with 2 layers; self-attention: 6 blocks with 6 layers each; Gaussian seeds: 144; window size: 12×12; two $2\times$ pixel-shuffle stages	20.445
MINet	RDN: 16 dense blocks, 8 layers per block, feature width and growth rate 64; three $2\times$ pixel-shuffle stages; experts: 8, top-k: 2; each expert: 3 hidden layers of width 256; local ensemble and cell decoding enabled	24.013
GSSA-ViT	Initial Gaussian grid: 33×64 (2,112 primitives); patch size and stride: 1×1; Transformer: 8 blocks, width 1,024, 16 heads, MLP ratio 4; learnable absolute positions; separate Gaussian and variable prediction heads	113.480
A^2PSR	Configuration in Table 4	7.924

GSSA-ViT. GSSA-ViT Han et al. (2026) uses a vision transformer to predict Gaussian attributes and jointly reconstructs the three atmospheric variables through differentiable Gaussian splatting. Under our experimental settings, the official Gaussian count and default hyperparameters, including a 48×96 initial Gaussian grid and a 24-layer transformer, yielded poor reconstruction performance. Among the configurations we tested, performance improved only after aligning the initial Gaussian distribution with the 33×64 low-resolution input grid and reducing the transformer depth to 8. The adapted model uses $2{,}112$ Gaussian primitives, an embedding dimension of $1{,}024$, and 16 attention heads. Despite these adjustments, we did not reproduce the performance reported for the official model; the GSSA-ViT results in this paper correspond to this adapted implementation, which differs from the official configuration.

F Supplementary Evaluation at $22.5\times$

F.1 Pixelwise MAE and SSIM

This supplementary evaluation focuses on $22.5\times$, the largest tested extrapolation scale, corresponding to the native $0.25°$ ERA5 grid. MAE and SSIM assess T2M, U10, and V10 separately. The wind-speed FSS definition, evaluation settings, and results are provided in the following subsection.

Table 7: Supplementary MAE and SSIM at $22.5\times$. MAE is reported in K for T2M and $\mathrm{m\,s^{-1}}$ for U10 and V10. SSIM is dimensionless. Arrows indicate better performance. Bold and underline mark the best and second-best results in each column.

Method	MAE ↓			SSIM ↑		
	T2M	U10	V10	T2M	U10	V10
Bicubic	1.5768	1.5632	1.7589	0.9451	0.8449	0.8290
Bilinear	1.6210	1.6150	1.7663	0.9451	0.8428	0.8289
Meta-SR	0.9498	0.8847	0.8611	0.9483	0.8709	0.8640
LIIF	0.9975	1.0465	0.9279	0.9498	0.8537	0.8581
SRNO	0.8354	0.8766	0.8665	0.9539	0.8736	0.8667
CiaoSR	<u>0.7892</u>	<u>0.8688</u>	<u>0.8458</u>	0.9491	0.8602	0.8545
HIIF	1.0346	1.0651	1.1742	0.9521	0.8682	0.8536
GSASR	0.8462	0.8780	0.8463	<u>0.9554</u>	<u>0.8779</u>	**0.8726**
MINet	0.9148	0.9326	1.0122	0.9488	0.8598	0.8343
GSSA-ViT	1.3201	1.4449	1.5107	0.9430	0.8336	0.8158
A^2PSR (ours)	**0.7548**	**0.8332**	**0.8161**	**0.9582**	**0.8792**	<u>0.8718</u>

MAE. Mean absolute error is computed after transforming predictions and targets back to physical units. For variable c, we use the per-frame, latitude-area-weighted MAE

$$\mathrm{MAE}_c = \frac{1}{N H_t} \sum_{n=1}^{N} \sum_{h=1}^{H_t} \omega_h \left[\frac{1}{W_t} \sum_{w=1}^{W_t} \left| \widehat{Y}_{c,h,w}^{(n)} - Y_{c,h,w}^{(n)} \right| \right], \tag{17}$$

where ω_h is the normalized spherical-area weight defined in the main text. The spatial average is computed for each frame and then averaged over the $N = 8{,}784$ hourly test frames; lower is better.

SSIM. For each frame and variable, both fields are normalized using the ground-truth range:

$$\widetilde{Y} = \frac{Y - m_Y}{r_Y}, \qquad \widetilde{\widehat{Y}} = \frac{\widehat{Y} - m_Y}{r_Y}, \qquad m_Y = \min_{h,w} Y_{h,w}, \quad r_Y = \max_{h,w} Y_{h,w} - m_Y, \tag{18}$$

where r_Y is set to 1 when the ground-truth field is constant. We then compute the local SSIM map using an 11×11 Gaussian window with $\sigma = 1.5$. For local means μ_x, μ_y, variances σ_x^2, σ_y^2, and covariance σ_{xy}, each map value is

$$\mathrm{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \qquad C_1 = 0.01^2, \quad C_2 = 0.03^2. \tag{19}$$

The Gaussian window is circularly padded along longitude and uses valid convolution along latitude, so the latitude aggregation covers the $H_t - 10$ valid rows. The SSIM map is averaged over longitude, weighted by the normalized spherical-area weights over these valid latitude rows, and then averaged over the $N = 8{,}784$ frames; higher is better.

F.2 Extreme-Cyclone FSS

We evaluate spatial matching of strong-wind structures in extreme-cyclone cases using the Fractions Skill Score (FSS). FSS compares predicted and observed strong-wind event fractions within spatial neighborhoods. The evaluation uses 90 storm-time cases from official 2020 IBTrACS tracks in the North Atlantic, Eastern/Central North Pacific, and Western North Pacific basins. The selected tropical cyclones have an annual peak USA_WIND of at least 64 kt and the sample times are determined from the IBTrACS tracks. Main-track records during tropical stages at 6-hour intervals form the candidate times. We select samples by cycling through cyclones and the pre-peak, near-peak, and post-peak phases, enforcing a minimum separation of 48 hours between selected times from the same cyclone. The near-peak phase extends from 12 hours before the first occurrence of peak intensity to 12 hours after its last occurrence. For each case, we retain a $20^\circ \times 20^\circ$ region centered at the best-track position, corresponding to 81×81 grid cells on the native 0.25° grid. The same cases and regions are used for every method. After restoring U10 and V10 to physical units, the 10-m wind speed is computed as

$$\mathrm{WS} = \sqrt{U_{10}^2 + V_{10}^2}. \tag{20}$$

Table 8: Aggregate FSS on the extreme-cyclone cases. The window is measured in grid cells, higher is better. Bold and underline mark the best and second-best results in each column.

Method	1×1	3×3	5×5	9×9	17×17
Bicubic	0.0104	0.0122	0.0132	0.0149	0.0217
Bilinear	0.0000	0.0000	0.0000	0.0000	0.0000
Meta-SR	0.2812	0.3272	0.3578	0.4055	0.4786
LIIF	0.2017	0.2356	0.2593	0.3001	0.3628
SRNO	0.3273	0.3845	0.4224	0.4780	0.5466
CiaoSR	<u>0.4563</u>	<u>0.5397</u>	<u>0.5905</u>	<u>0.6649</u>	<u>0.7663</u>
HIIF	0.0000	0.0000	0.0000	0.0000	0.0000
GSASR	0.1188	0.1334	0.1451	0.1667	0.1940
MINet	0.3004	0.3552	0.3957	0.4699	0.5880
GSSA-ViT	0.0405	0.0471	0.0528	0.0650	0.0933
Ours	**0.4825**	**0.5579**	**0.6073**	**0.6852**	**0.7889**

We define a strong-wind event by $\mathrm{WS} \geq 17.2\,\mathrm{m\,s^{-1}}$, the lower bound of Beaufort force 8. This threshold is applied identically to the GT and each prediction. All grid cells in each selected region are retained, including both event and non-event cells, so that missed events, false alarms, and correct non-events are all included. For each square neighborhood of $s \times s$ grid cells with $s \in \{1, 3, 5, 9, 17\}$, we compute the local event fraction with spherical grid-cell-area weighting. For every grid point in the evaluation region, the complete neighborhood is read from the global field with periodic longitude and no zero padding at the evaluation-region boundaries. Let $f_{n,s}$ and $\hat{f}_{n,s}$ denote the GT and predicted event fractions for case n at neighborhood size s. The aggregate FSS is computed by pooling the area-weighted error and reference terms over all cases.

$$\mathrm{FSS}_s = 1 - \frac{\sum_n \left\langle (\hat{f}_{n,s} - f_{n,s})^2 \right\rangle_A}{\sum_n \left\langle \hat{f}_{n,s}^2 + f_{n,s}^2 \right\rangle_A}, \tag{21}$$

where $\langle \cdot \rangle_A$ denotes normalized spherical-area weighting. A sample with a zero reference term contributes zero to both pooled terms. Higher FSS indicates better spatial matching. Table 8 reports the aggregate FSS for the evaluated methods and neighborhood windows.

G Probe Allocation and Regional Detail Demand

Table 9: Regional demand and probe allocation on the 2020 test set. Each quartile contains $4{,}497{,}408$ sample–cell observations. Single-variable demand uses $E_{q,c}$; joint standardized demand uses E_q^{joint}.

Statistic	Quartile	T2M	U10	V10	Joint standardized
Mean demand	Q1	0.1194	0.4502	0.4218	0.0154
	Q2	0.5160	1.3097	1.1576	0.0408
	Q3	2.6073	3.0082	2.5552	0.0852
	Q4	16.1939	11.8130	9.0685	0.2608
Mean n_q	Q1	1.0825	1.0602	1.0632	1.0402
	Q2	1.1538	1.1687	1.1651	1.1481
	Q3	1.2524	1.2611	1.2747	1.2914
	Q4	1.3942	1.3928	1.3798	1.4031
Mean $(n_q - 1)$	Q1	0.0825	0.0602	0.0632	0.0402
	Q2	0.1538	0.1687	0.1651	0.1481
	Q3	0.2524	0.2611	0.2747	0.2914
	Q4	0.3942	0.3928	0.3798	0.4031

Regional detail demand. We use local interpolation residual energy as a proxy for detail demand: larger values indicate more local variation left unresolved by interpolation. For each sample, we retain the method notation $\mathbf{X}$, $\mathbf{Y}$, C, and $q \in \{1, \dots, Q\}$, where $Q = (H_s - 1)W_s$. Define the per-variable interpolation error $E_{q,c}$ and joint standardized demand E_q^{joint} as

$$E_{q,c} = \frac{\sum_{(h,w)\in\Omega_q} \omega_h \left[Y_{c,h,w} - [\mathcal{I}(\mathbf{X})]_{c,h,w}\right]^2}{\sum_{(h,w)\in\Omega_q} \omega_h}, \qquad E_q^{\text{joint}} = \frac{1}{C}\sum_{c=1}^{C} \frac{E_{q,c}}{\sigma_c^2}. \tag{22}$$

Here, $\mathbf{X}$ and $\mathbf{Y}$ are expressed in physical units. We evaluate regional detail demand and probe allocation at $22.5\times$. The operator $\mathcal{I}$ maps the 33×64 source field to the 721×1440 target grid using bilinear interpolation with aligned latitude endpoints and periodic longitude. The set Ω_q contains target-grid locations (h, w) assigned to source cell q according to their point centers. Points on internal cell boundaries are assigned to the next row or column, and the final latitude endpoint is assigned to the last cell row. The latitude weight ω_h is the same area weight used for Lat-RMSE in the main text. The training-set standard deviation σ_c follows Appendix A, making E_q^{joint} the average interpolation MSE over the $C = 3$ standardized variables. The shared normalization mean cancels in the residual.

Allocation statistics. For each sample, let n_q denote the number of retained auxiliary probes in the source cell q defined in the method, excluding anchors. The count $n_q - 1$ removes the one-per-cell guarantee and isolates probes assigned through competition for the remaining budget. We compute Spearman correlation across cells within each sample and then average over samples; subtracting one probe from every cell leaves these ranks unchanged. All $8{,}784$ samples yield valid correlations for each demand measure. For each sample and demand measure ($E_{q,c}$ or E_q^{joint}), we divide the $Q = (33 - 1)64 = 2{,}048$ source cells into Q1–Q4 by demand-rank percentiles, from lowest to highest demand. Tied demand values receive average ranks and are assigned to the same group. These are sample-specific demand groups, not fixed geographic regions. The reported $4{,}497{,}408$ observations per group count sample–cell pairs pooled over $8{,}784$ samples. Table 9 reports the mean demand and probe counts pooled within each quartile. Across all $8{,}784$ samples, the mean per-sample Spearman correlations are 0.1161, 0.1282, 0.1180, and 0.1396 for T2M, U10, V10, and joint standardized demand, respectively. Average probe counts increase monotonically across demand quartiles for all four measures. For joint standardized demand, Q4 receives 10.02 times as many probes beyond the guarantee as Q1 (0.4031 versus 0.0402); including the guaranteed probe, the ratio is 1.35. These results support a tendency to allocate more probes to regions with higher interpolation residuals.

H Training and Inference Costs

The arbitrary-scale models are trained with distributed data parallelism (DDP) on 8 NVIDIA RTX A6000 GPUs using FP16 precision, a global batch size of 8, and target scales uniformly sampled over $1\times$–$2\times$. For inference, all models use 8 NVIDIA RTX A6000 GPUs with FP32 precision. We evaluate all hourly ERA5 samples from 2020, jointly reconstructing T2M, U10, and V10 from a 33×64 grid ($5.625°$) to a 721×1440 grid ($0.25°$) at $22.5\times$. Table 10 reports peak training memory per GPU, total inference time, and peak inference memory. The reported wall-clock time covers inference on the full 2020 test set across 8 GPUs after warm-up, excluding data loading, metric computation, and output writing. GPU execution is synchronized at timing boundaries. Peak inference memory is the maximum GPU memory allocated by PyTorch during inference.

Table 10: Training costs and full-year 2020 inference costs on 8 GPUs for the compared arbitrary-scale learnable methods. U-Net is omitted because it is a fixed-scale reference, and interpolation methods have no training cost. Lower is better. Bold and underline mark the lowest and second-lowest costs in each column.

Method	Training memory (GB)	Inference time (s)	Inference memory (GB)
Meta-SR	**0.956**	376	20.95
LIIF	<u>1.504</u>	705	<u>1.706</u>
SRNO	2.294	418	13.90
CiaoSR	1.722	1519	3.228
HIIF	3.508	1107	29.59
GSASR	4.882	1086	**1.224**
MINet	2.036	591	10.24
GSSA-ViT	7.448	**77**	2.406
A^2PSR (ours)	4.308	<u>350</u>	8.802