WINDINR: LATENT-STATE INR FOR FAST LOCAL WIND QUERY AND CORRECTION IN COMPLEX TERRAIN

Anonymous authors
Paper under double-blind review

ABSTRACT

Many downstream decisions in complex terrain require fast wind estimates at a small number of user-specified locations and heights for a given forecast valid time, rather than another dense forecast field on a fixed grid. We present WindINR, a latent-state implicit neural representation framework for continuous high-resolution local wind query and sparse-observation correction. WindINR maps static terrain descriptors, a low-resolution background field, and continuous query coordinates to a high-resolution wind state through a latent-conditioned decoder. To enable rapid inference-time correction, WindINR separates reusable representation learning from sample-specific latent-state correction. During training, a privileged encoder infers a reference latent state from high-resolution supervision, a deployable latent predictor estimates an initial latent state from inference-time inputs alone, and their discrepancies are summarized into a dataset-adaptive Gaussian prior over latent corrections. At inference time, within the WindINR module, network weights remain fixed and only the latent state is updated by minimizing a regularized correction objective using sparse observations and their uncertainty. We train WindINR on the large-scale FuXi-CFD dataset spanning diverse terrain and evaluate it across complementary observation settings, including controlled OSSEs, simulated helicopter trajectories, and real UAV measurements over Tromsø. Across these settings, assimilating sparse observations consistently improves local wind-field predictions, with WindINR outperforming several baselines, including WindSeer. Moreover, by performing online correction only in a low-dimensional latent space, WindINR achieves a 14$\times$ speedup over full-network fine-tuning. The resulting representation remains continuously queryable at arbitrary spatial coordinates, making WindINR a promising and efficient correction framework that provides a practical interface between kilometer-scale background products, sparse local observations, and wind queries in complex terrain.

1 INTRODUCTION

Operational weather products are usually delivered as gridded analyses and forecasts, but many decisions in complex terrain require only a small number of fast, accurate wind estimates at user-specified locations. During a helicopter approach, for example, the relevant question is often not the full wind field over a large domain, but the local gusts and potential turbulence along an approach corridor and around a landing zone. These local winds can vary strongly because of terrain-induced channeling, ridge acceleration, land–sea contrast, and boundary-layer structure. Sparse new observations from UAVs or stations may also arrive immediately before the decision. The desired operating point is therefore not simply another dense high-resolution forecast, but a local wind-state estimator that can be queried continuously and corrected quickly. Fig. 1 illustrates such a scenario.

Recent advances in AI weather forecasting and terrain-aware wind downscaling have made fast local wind estimation increasingly feasible. Large AI weather models can generate forecasts efficiently, and downscaling methods can map coarse atmospheric inputs to fine-scale terrain-dependent wind structures. However, most existing approaches still operate on fixed grids and treat downscaling as a one-shot prediction problem, which limits their ability to answer arbitrary off-grid local queries or to rapidly incorporate sparse observations after deployment. The challenge is not only one of

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.

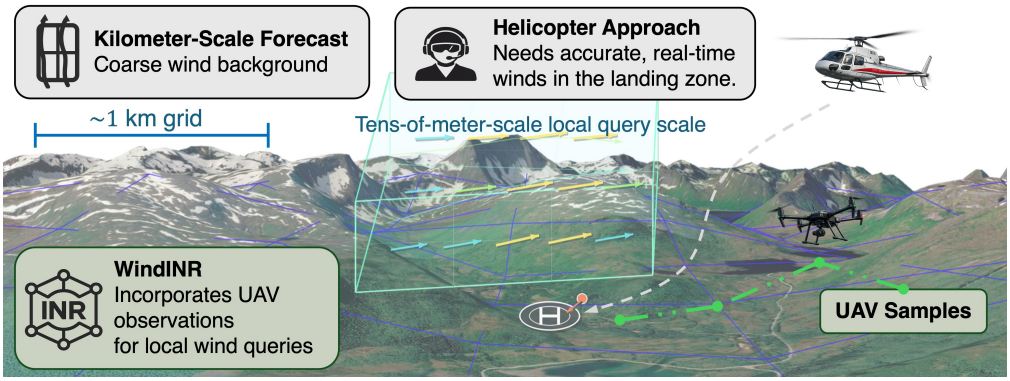


Figure 1: **Operational motivation for WindINR.** Kilometer-scale forecast products provide useful coarse background information over complex terrain, but helicopter approach and landing decisions require fast wind estimates at tens-of-meters scale and at user-specified off-grid locations. In the illustrated workflow, a UAV flying ahead of the helicopter provides sparse local wind observations along the approach corridor. WindINR uses the terrain and coarse forecast background as conditioning inputs, and rapidly updates a compact latent state with these sparse observations while preserving continuous local wind query.

spatial resolution. Fine-scale wind prediction over mountains and fjords cannot be obtained from low-resolution data alone: Multiple terrain-aligned high-resolution realizations can be consistent with the same kilometer-scale atmospheric state. Sparse but high-resolution observations, such as those of the drone of Fig. 1 are therefore necessary not merely to correct small point-wise errors but, more crucially, to fully resolve local state variables.

Conditional implicit neural representations are a natural fit for this task because they define the wind field as a continuous function of coordinates, supporting off-grid queries. The main challenge lies in fast observation-guided correction. A straightforward strategy would be to fine-tune the whole network during inference after observations arrive, but this is unattractive in the sparse-observation regime: the number of observations is often far smaller than the number of trainable parameters, making overfitting likely, while repeated weight updates also increase latency and entangle reusable representation learning with case-specific state correction. Instead, we seek a formulation in which offline learning captures reusable terrain-aware structure, while online assimilation is restricted to a compact sample-specific latent state regularized by an explicit dataset-adaptive prior.

We introduce WindINR, a latent-state conditional implicit neural representation for fast local wind query and correction. Given static terrain descriptors, a low-resolution wind background, and continuous query coordinates, WindINR predicts the wind state through a shared INR decoder conditioned on a sample-specific latent state. The decoder captures reusable terrain-aware structure, while the latent state represents unresolved local variation and provides the degree of freedom for observation-guided correction. During training, a privileged reference encoder uses high-resolution supervision to define reference latents, and a deployable latent predictor estimates initial latents from inference-time inputs alone. Their discrepancies define a dataset-adaptive Gaussian prior over latent corrections. At inference time, all network weights remain fixed: without observations, WindINR decodes the initial latent state directly; with sparse observations, it updates only the low-dimensional latent state by minimizing a regularized correction objective evaluated through direct INR queries at the observation coordinates. Thus the same arbitrary-coordinate query interface is preserved before and after correction, while avoiding full-network fine-tuning.

This paper makes two main contributions:

- We formulate WindINR, a latent-state conditional INR for terrain-aware local wind estimation with arbitrary-coordinate continuous query.
- We introduce a training and inference scheme for fast observation-guided latent correction, using training-set latent discrepancies to define a dataset-adaptive prior and updating only the latent state.

We train WindINR on the large-scale FuXi-CFD dataset spanning diverse terrain and evaluate it with controlled OSSEs, simulated helicopter trajectories, and real UAV measurements over Tromsø. In OSSEs on 1,881 held-out cases, 1,024 observations reduce mean component RMSE by 13.26%, with WindINR outperforming baselines including WindSeer. The helicopter pilot further reduces look-ahead error by 15.17%, while real UAV experiments validate its effectiveness in mountainous terrain. By updating only a low-dimensional latent space, WindINR also achieves a 14$\times$ speedup over full-network fine-tuning while retaining arbitrary-coordinate querying.

2 Related Work

This work lies at the intersection of AI weather downscaling, continuous neural representations, and observation-driven state estimation.

AI weather forecasting and downscaling. Recent AI weather models such as FourCastNet, Pangu-Weather, GraphCast, and GenCast deliver strong forecast skill and low latency, but they generally remain fixed-grid predictors (Pathak et al., 2022; Bi et al., 2023; Lam et al., 2023; Price et al., 2025). On the downscaling side, FuXi-CFD, CorrDiff, and related terrain-aware frameworks show that learned models can recover realistic fine-scale atmospheric structure from coarse inputs in complex terrain (Lin et al., 2026; Mardani et al., 2025; Hess et al., 2025). WindSeer further considers observation-conditioned reconstruction, learning to infer a complete high-resolution wind field from sparse local measurements and terrain information (Achermann et al., 2024); however, it likewise operates as a pretrained feed-forward model rather than performing online correction after deployment. Our focus is different: rather than full-field gridded prediction alone, we target continuous local query with fast observation-driven correction after deployment.

Continuous representations and implicit neural representations. Implicit Neural Representations were initially introduced for shape modeling (Park et al., 2019; Mescheder et al., 2019) and image re-synthesis (Mildenhall et al., 2020) and have since been considerably refined (Zhang et al., 2023; Chen et al., 2025). They provide a natural interface for arbitrary-coordinate and arbitrary-resolution decoding (Sitzmann et al., 2020; Chen et al., 2021). In atmospheric and geophysical applications, recent works such as KANI, HyperDS, SpLIIF, and continuous field reconstruction from sparse observations show that such representations are well matched to off-grid meteorological query and flexible downscaling (Liu et al., 2025; 2024; del Valle Inclan Redondo et al., 2025; Luo et al., 2024). However, most existing INR-based weather methods treat the model primarily as a feed-forward downscaler or reconstructor, rather than studying how a deployed conditional INR can be updated efficiently when new sparse observations arrive.

Data assimilation and AI-based assimilation. Data assimilation combines a background estimate with observations under uncertainty assumptions to produce an updated analysis state. Recent AI-based approaches explore arbitrary-resolution neural-process assimilation, latent-space variational methods, diffusion-based DA, and forecast-assimilation systems coupled to AI weather models (Chen et al., 2024; Fan et al., 2025; Xiao et al., 2025; Fan et al., 2026a; Huang et al., 2024; Sun et al., 2025; Andry et al., 2025; Xu et al., 2025; Fan et al., 2026b). While these methods advance learned observation-conditioned state estimation, they mostly target gridded analyses, fixed model outputs, or large generative reconstructions, rather than terrain-aware continuous local query on top of a conditional INR background.

Taken together, existing work has advanced terrain-aware downscaling, continuous atmospheric representations, and observation-driven state estimation largely along separate directions. Our work targets their intersection: a terrain-aware conditional INR for local wind estimation whose sample-specific latent state can be corrected using a dataset-adaptive prior and sparse observations, while preserving arbitrary-coordinate continuous query.

3 Formulation

Our goal is to estimate the local high-resolution wind field over complex terrain from terrain descriptors, a low-resolution background wind prediction, and sparse local observations when available.

We call the resulting framework WindINR: a latent-state implicit neural representation in which a learned decoder supports continuous query at arbitrary coordinates and only a low-dimensional latent state is updated at inference time. Figure 2 summarizes the full learning-and-inference pipeline and highlights the deployed WindINR boundary at inference time.

3.1 Latent-State INR for Querying Local Wind Fields

Let $\Omega \subset \mathbb{R}^3$ denote the continuous query domain, and let $p = (r_x, r_y, h) \in \Omega$ be a 3D query coordinate. For any given sample, that is, a target area with its static terrain descriptors s and sparse observations of the local conditions $\mathcal{O}$, let x^{LR} be the low-resolution grid-based wind prediction, which we assume to be given *a priori*, and let $x^{\mathrm{HR}} : \Omega \to \mathbb{R}^C$ denote the target high-resolution wind field. In our current setting, $C = 3$ because we predict wind components along three axes.

A standard conditional INR can be written as

$$\widehat{x}^{\mathrm{HR}}(p) = \mathcal{F}_\theta\big(s, x^{\mathrm{LR}}, p\big), \qquad \forall p \in \Omega, \tag{1}$$

where $\mathcal{F}_\theta$ is a decoder. This already supports arbitrary-coordinate continuous query, but the reconstruction is fully determined by the decoder weights θ and the deployable inputs (s, x^{LR}).

To provide a sample-specific and updatable degree of freedom, we introduce a low-dimensional latent state $z \in \mathbb{R}^d$ and define the latent-state INR as

$$\widehat{x}^{\mathrm{HR}}(p; z) = \mathcal{F}_\theta\big(s, x^{\mathrm{LR}}, p, z\big), \qquad p \in \Omega. \tag{2}$$

Here, θ captures reconstruction structure shared across samples, while z represents sample-specific variation unresolved by (s, x^{LR}) alone. Therefore, when new local observations become available, we refine the queried field by updating z rather than modifying θ. Because the backbone is an INR, predictions at observation locations are obtained by direct evaluation of $\mathcal{F}_\theta$ at the observed coordinates, so the same continuous query interface is preserved before and after latent correction.

3.2 Inferring the Latent State

We now describe how the latent state is inferred at training time and at inference time.

Training-time reference latent state. During training, when high-resolution supervision is available, we define a privileged reference latent state

$$z_{\mathrm{ref}} = \mathcal{E}_\alpha\big(s, x^{\mathrm{LR}}, x^{\mathrm{HR}}\big). \tag{3}$$

This latent state provides a high-quality sample-specific target in latent space during training, but it is not available at inference time.

Why not directly learn an observation-conditioned latent map? At inference time, the high-resolution target field is unavailable, so the goal is to infer a latent state that approximates the reference latent state that would have been obtained under privileged supervision. A naive strategy would be to learn a unified observation-conditioned predictor, for example $z = \mathcal{Z}_\eta\big(s, x^{\mathrm{LR}}, \mathcal{O}\big)$, that directly maps the deployable inputs and the available observations to a latent state aligned with z_{ref}. We do not adopt this fully amortized strategy because, in our setting, the observation set $\mathcal{O}$ is sparse, irregular, variable in number, and heterogeneous in quality. A single feed-forward predictor would therefore be tied to the observation layouts and noise patterns seen during training. Instead, we treat the observations at inference time as constraints that guide regularized correction of a latent estimate.

Two-step latent inference. Let $z_0 = \mathcal{Z}_\beta(s, x^{\mathrm{LR}})$ denote the no-observation initial latent state. We therefore define the latent inference map as

$$\mathcal{Z}\big(s, x^{\mathrm{LR}}, \mathcal{O}\big) = \begin{cases} z_0, & \mathcal{O} = \varnothing, \\ \mathcal{U}_\gamma(z_0, \mathcal{O}; s, x^{\mathrm{LR}}), & \mathcal{O} \neq \varnothing. \end{cases} \tag{4}$$

Here, $\mathcal{Z}_\beta$ denotes a deployable no-observation latent predictor that provides an initial latent state from (s, x^{LR}) alone, while $\mathcal{U}_\gamma$ denotes an observation-conditioned correction operator. The concrete learning objective for $\mathcal{Z}_\beta$ and the concrete form of $\mathcal{U}_\gamma$ are introduced in Section 4. When no observations are available, $\mathcal{Z}_\beta$ returns the initial deployable latent state. When observations are available, $\mathcal{U}_\gamma$ refines this initial latent state while keeping the network weights fixed.

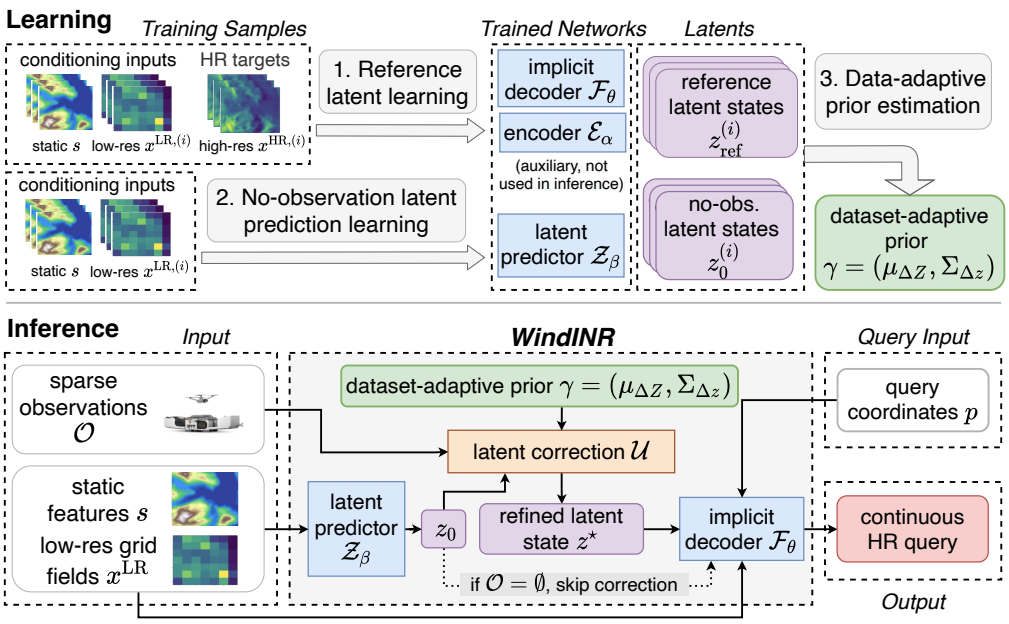


Figure 2: **Overview of WindINR.** At training time, reference latent states and no-observation latent states are learned and their discrepancies are summarized into a dataset-adaptive prior γ. At inference time, sparse observations guide a regularized latent correction from the initial latent state z_0 to a refined latent state $z^\star$, enabling continuous high-resolution query through the implicit decoder $\mathcal{F}_\theta$.

Dataset-adaptive prior for correction. In contrast to many previous correction schemes, where γ is treated as a tunable regularization parameter chosen empirically, here γ denotes the parameters of a dataset-adaptive prior for latent correction. Specifically, over the training set we consider the discrepancies between the reference codes and their matched no-observation codes, $\Delta z^{(i)} = z_{\mathrm{ref}}^{(i)} - z_0^{(i)}$, and summarize them through their empirical mean and covariance, $\gamma = (\mu_{\Delta z}, \Sigma_{\Delta z})$. Rather than imposing a generic isotropic ℓ_2 penalty with a hand-tuned weight, this formulation uses training-set latent discrepancies to define a more informative prior, thereby guiding the correction toward latent-error patterns supported by the data.

The final high-resolution field remains continuously queryable as $\widehat{x}^{\mathrm{HR}}(p; \mathcal{O}) = \mathcal{F}_\theta\big(s, x^{\mathrm{LR}}, p, \mathcal{Z}(s, x^{\mathrm{LR}}, \mathcal{O})\big)$, $p \in \Omega$. This separation defines the core operating principle of WindINR: the decoder learns a reusable terrain-aware representation, while sparse local observations refine only the latent state of the current local wind field.

4 Learning and Latent Correction in WindINR

We now instantiate the operators that define WindINR in Section 3. At training time, we first learn a privileged reference latent state and then train a deployable no-observation latent predictor. We next estimate the parameters of the dataset-adaptive prior from training-set latent discrepancies. At inference time, when sparse observations are available, WindINR refines the latent state by solving a regularized correction problem while keeping all network weights fixed.

4.1 Reference latent learning

We first learn the implicit decoder $\mathcal{F}_\theta$ together with the reference latent encoder $\mathcal{E}_\alpha$. Although Section 3.2 writes the reference latent state abstractly as $z_{\mathrm{ref}} = \mathcal{E}_\alpha(s, x^{\mathrm{LR}}, x^{\mathrm{HR}})$, in implementation the high-resolution target field enters through sampled support and query sets. For each training sample, we draw $\mathcal{P}^{\mathrm{sup}} = \{(p_i, x^{\mathrm{HR}}(p_i))\}_{i=1}^{m_{\mathrm{sup}}}$, $\mathcal{P}^{\mathrm{qry}} = \{(q_j, x^{\mathrm{HR}}(q_j))\}_{j=1}^{m_{\mathrm{qry}}}$, where $p_i, q_j \in \Omega$ are continuous coordinates sampled from the high-resolution domain. The reference latent state is inferred from the deployable inputs together with the support set: $z_{\mathrm{ref}} =$

$\mathcal{E}_\alpha\big(s, x^{\mathrm{LR}}, \mathcal{P}^{\mathrm{sup}}\big)$. Conditioned on this latent state, the decoder predicts the field at held-out query coordinates, $\widehat{x}^{\mathrm{HR}}_{\mathrm{ref}}(q_j) = \mathcal{F}_\theta\big(s, x^{\mathrm{LR}}, q_j, z_{\mathrm{ref}}\big)$. We train the reference branch by minimizing $\mathcal{L}_{\mathrm{ref}} = \frac{1}{m_{\mathrm{qry}}} \sum_{j=1}^{m_{\mathrm{qry}}} \big\|\widehat{x}^{\mathrm{HR}}_{\mathrm{ref}}(q_j) - x^{\mathrm{HR}}(q_j)\big\|_2^2 + \lambda_{\mathrm{ref}} \|z_{\mathrm{ref}}\|_2^2$. This stage serves two roles. First, it learns $\mathcal{F}_\theta$ as a latent-conditioned INR that supports arbitrary-coordinate continuous query. Second, it defines a high-quality training-time latent state z_{ref} that acts as the latent target for the later deployable branch.

4.2 No-observation latent prediction

We next instantiate the $\mathcal{O} = \varnothing$ branch in Eq. (4) by learning the deployable predictor $\mathcal{Z}_\beta$. For each sample, the predictor returns the initial no-observation latent state $z_0 = \mathcal{Z}_\beta\big(s, x^{\mathrm{LR}}\big)$. This latent state is decoded by the frozen INR as $\widehat{x}^{\mathrm{HR}}_0(q_j) = \mathcal{F}_\theta\big(s, x^{\mathrm{LR}}, q_j, z_0\big)$. At this stage, $\mathcal{F}_\theta$ and $\mathcal{E}_\alpha$ are fixed, and only $\mathcal{Z}_\beta$ is optimized. The objective combines reconstruction under the frozen decoder with alignment to the reference latent state. We take it to be $\mathcal{L}_\beta = \frac{1}{m_{\mathrm{qry}}} \sum_{j=1}^{m_{\mathrm{qry}}} \big\|\widehat{x}^{\mathrm{HR}}_0(q_j) - x^{\mathrm{HR}}(q_j)\big\|_2^2 + \lambda_{\mathrm{align}} \|z_0 - z_{\mathrm{ref}}\|_2^2$. The first term ensures that the no-observation latent state remains useful for reconstruction. The second term encourages $\mathcal{Z}_\beta$ to approximate the privileged reference latent state from deployable inputs alone. After this stage, WindINR can already answer continuous queries without local observations through $\widehat{x}^{\mathrm{HR}}(p) = \mathcal{F}_\theta(s, x^{\mathrm{LR}}, p, z_0)$.

4.3 Estimating the dataset-adaptive prior

We now instantiate the prior parameter γ introduced in Section 3.2. Over a training set of N samples, we form the latent discrepancies $\Delta z^{(i)} = z^{(i)}_{\mathrm{ref}} - z^{(i)}_0$, $i = 1, \ldots, N$. These samples describe how the deployable no-observation latent state typically differs from the corresponding reference latent state.

We summarize these discrepancies by their empirical mean and covariance: $\mu_{\Delta z} = \frac{1}{N} \sum_{i=1}^{N} \Delta z^{(i)}$, $\Sigma_{\Delta z} = \frac{1}{N-1} \sum_{i=1}^{N} (\Delta z^{(i)} - \mu_{\Delta z})(\Delta z^{(i)} - \mu_{\Delta z})^\top$. We then define $\gamma = (\mu_{\Delta z}, \Sigma_{\Delta z})$. This yields a Gaussian prior on the latent correction increment, $p(\delta z \mid \gamma) = \mathcal{N}(\mu_{\Delta z}, \Sigma_{\Delta z})$, $\delta z = z - z_0$. Compared with a generic isotropic ℓ_2 regularizer with a hand-tuned weight, this prior captures both the typical direction and the covariance structure of latent corrections supported by the training data.

4.4 Observation-guided latent correction

When local observations are available, WindINR instantiates the $\mathcal{O} \neq \varnothing$ branch in Eq. (4) through the correction operator $\mathcal{U}_\gamma$. For implementation, we write $\mathcal{O} = \{(p_n, y_n, M_n, R_n)\}_{n=1}^{N_{\mathrm{obs}}}$, where $p_n \in \Omega$ is the observation coordinate, y_n is the observed value, M_n selects the observed variables, and R_n is the corresponding observation-error covariance or variance.

We implement $\mathcal{U}_\gamma$ as regularized latent correction: $\delta z^\star = \arg\min_{\delta z \in \mathbb{R}^d} \mathcal{J}(\delta z; z_0, \mathcal{O}, \gamma)$, with

$$\begin{aligned} \mathcal{J}(\delta z; z_0, \mathcal{O}, \gamma) = \frac{1}{2} \|\delta z - \mu_{\Delta z}\|^2_{(\Sigma_{\Delta z} + \varepsilon I)^{-1}} \\ + \frac{1}{2} \sum_{n=1}^{N_{\mathrm{obs}}} \big\| M_n \mathcal{F}_\theta\big(s, x^{\mathrm{LR}}, p_n, z_0 + \delta z\big) - y_n \big\|^2_{R_n^{-1}}, \end{aligned} \tag{5}$$

where $\varepsilon > 0$ is a small numerical-stability constant and $\|a\|^2_{A^{-1}} = a^\top A^{-1} a$. The refined latent state is then $z^\star = z_0 + \delta z^\star$. Thus, the first term enforces the dataset-adaptive prior induced by training-set latent discrepancies, while the second term matches the corrected latent state to the available observations through direct decoder evaluations at the observation coordinates. All network parameters inside WindINR remain fixed during this step; only the d-dimensional latent state is updated in this global formulation. If no observations are available, no correction is performed and the latent state remains z_0. A probabilistic MAP derivation of this objective is provided in Appendix A.

Finally, the refined high-resolution field remains continuously queryable as $\widehat{x}^{\mathrm{HR}}_\star(p) = \mathcal{F}_\theta\big(s, x^{\mathrm{LR}}, p, z^\star\big)$. Therefore, WindINR instantiates the formulation in Section 3 by preserving the continuous-query interface of the INR while restricting inference-time correction to a low-dimensional latent correction problem.

5 EXPERIMENTS

We evaluate WindINR's ability to answer local wind queries and correct them using new measurements. Our study combines controlled experiments on simulated wind fields with an evaluation against real UAV observations.

5.1 EXPERIMENTAL SETUP

Training and inference setup. We train WindINR on FuXi-CFD dataset (Lin et al., 2026), a large collection of terrain-resolving CFD simulations. It contains 12,532 wind-field cases spanning varied terrain, providing both the volume and the diversity of terrain–wind configurations needed to learn a reusable local wind representation. Each case covers a $9\,\mathrm{km} \times 9\,\mathrm{km}$ region at 30 m horizontal resolution. We use 8,772/1,879/1,881 cases for training/validation/testing and predict the three wind components u, v, w at 27 height levels between 5 and 214.29 m, conditioned on terrain and a coarse wind background. For WindINR, network weights remain fixed when observations arrive. On FuXi-CFD dataset, we update a 3×3 grid of latent controls with a training-estimated prior and eight L-BFGS outer steps. The same checkpoint, prior, and solver are used across observation counts.

Baselines. We compare with five primary baselines. **FuXi-CFD** (Lin et al., 2026) provides a published background-reconstruction reference without local observations. **Partial FT** performs partial fine-tuning of the latent seed and final output-head layer, while **Full FT** fine-tunes all network weights using the new observations. These two strategies test whether updating network parameters is preferable to correcting WindINR's latent state. **WindSeer** learns an end-to-end mapping from sparse observations to a high-resolution wind field. Since the original architecture does not natively incorporate a low-resolution background field, we augment our local implementation with an additional module to encode and fuse the background information. The implementation is inspired by Achermann et al. (2024), with details in Appendix B.5. **Residual IDW** uses inverse-distance weighting to interpolate observation residuals and adds the resulting correction to the background. **WindINR (Iso.)** uses an isotropic Gaussian prior for ablation.

5.2 SIMULATED OBSERVATIONS

For the simulated-observation experiments, we use high-resolution wind fields from the FuXi-CFD dataset as ground truth. We consider two complementary observation settings: randomly sampled observations for controlled evaluation and a UAV-aided approach scenario that mimics sequential measurements along a flight trajectory.

Table 1: **Random simulated observations.** Case-averaged RMSE in $\mathrm{m\,s^{-1}}$ over 1,881 test cases, with 2,048 held-out query points per case. "Raw" is WindINR before correction; gain and wins compare the corrected result with that raw field (Appendix B.1).

	Raw	Analysis RMSE					
Obs.	mean	u	v	w	Mean	Gain	Wins
256	0.4051	0.4959	0.4905	0.1586	0.3817	5.78%	83.41%
512	0.4051	0.4773	0.4716	0.1514	0.3668	9.45%	97.24%
1,024	0.4051	0.4576	0.4523	0.1442	0.3514	13.26%	99.68%
2,048	0.4052	0.4380	0.4333	0.1379	0.3364	16.98%	100.00%

Random observations. For each test case, we randomly sample 256, 512, 1,024, or 2,048 locations and add independent Gaussian noise with standard deviations $(0.5, 0.5, 0.2)\ \mathrm{m\,s^{-1}}$ to their u, v, w values. Another 2,048 locations are held out for evaluation. Notably, WindSeer has limited flexibility in handling varying numbers of observations. We therefore train separate WindSeer models for each of the four observation densities considered in our experiments. We report case-averaged RMSE for each component and the arithmetic mean of the three component RMSEs. WindINR's background reconstruction has lower mean component RMSE than FuXi-CFD (0.4051 versus 0.4141 $\mathrm{m\,s^{-1}}$; Table 2). Incorporating random observations reduces the error further at every evaluated count (Table 1). WindINR achieves lower mean error than Partial FT, Full FT, WindSeer, and Residual

IDW at all four counts (Table 2). Consistent gains over WindINR (Iso.) demonstrate the effectiveness of the dataset-adaptive prior. Against WindSeer, WindINR improves all three components at every observation count and wins on 91.71% of paired cases with 1,024 observations (Appendix B.6). This result is particularly notable because WindSeer is trained in a fully supervised manner for observation-to-field reconstruction, whereas WindINR achieves superior performance without such supervised training, highlighting the effectiveness of its test-time latent correction mechanism.

Table 2: **Background wind queries and random-observation correction.** Mean component RMSE in $\mathrm{m\,s^{-1}}$. Rows with observations use 1,881 test cases; "0" denotes no local observations. Bold and underlined values mark the lowest and second-lowest errors; dashes indicate inapplicable entries.

Obs.	FuXi-CFD	Partial FT	Full FT	WindSeer	Residual IDW	WindINR (Iso.)	WindINR
0	<u>0.4141</u>	–	–	–	–	–	**0.4051**
256	–	0.3962	0.5047	<u>0.3869</u>	0.3990	0.3886	**0.3817**
512	–	0.3804	0.4837	0.3783	0.3893	<u>0.3696</u>	**0.3668**
1,024	–	0.3696	0.4361	0.3699	0.3814	<u>0.3574</u>	**0.3514**
2,048	–	0.3634	0.3648	0.3600	0.3751	<u>0.3403</u>	**0.3364**

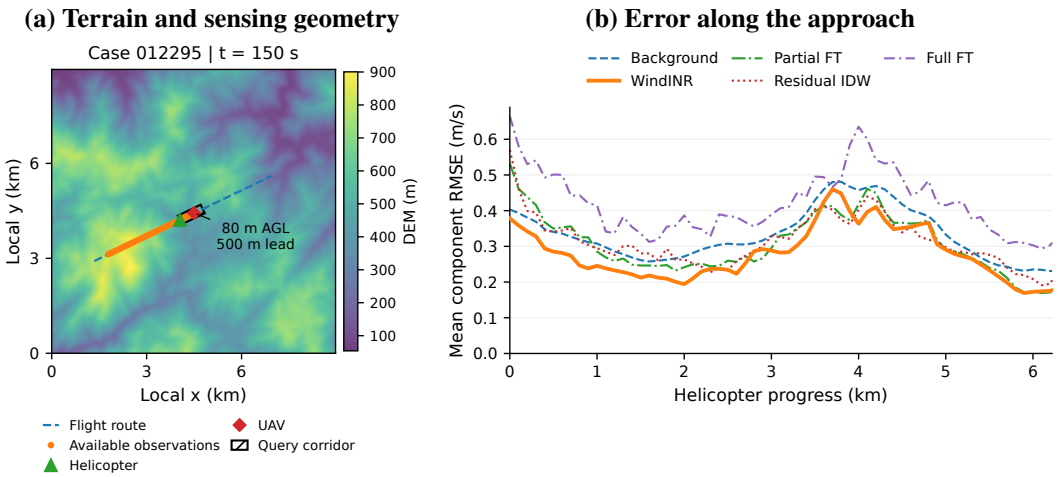


Figure 3: **UAV-aided wind queries on FuXi-CFD dataset.** (a) Terrain, route, observations, and the 800×300 m look-ahead query region at $t = 150$ s. (b) Mean component RMSE across eight cases at each helicopter progress point, excluding queries within 50 m of assimilated observations. Each case queries a steady simulated wind field as the available observation prefix grows.

UAV-aided approach. The trajectory experiment simulates a UAV flying ahead of a helicopter along an approach route using eight held-out CFD cases. As the helicopter advances, only UAV observations already collected along the route are used to correct the estimated wind in its upcoming 800×300 m corridor. We query the corrected field ahead of the helicopter and evaluate it against the CFD truth, excluding locations within 50 m of assimilated observations. Since the CFD fields are steady, the changing observation history and moving query region test whether measurements collected earlier along the route can improve wind estimates farther ahead. Here the number of available observations changes dynamically as the UAV advances. We compare WindINR with interpolation and both fine-tuning strategies; the WindSeer models trained for fixed observation counts are not included in this dynamic-observation setting. WindINR has the lowest average error at most points along the approach (Figure 3). WindINR reduces the background error by 15.17% and improves all eight case averages, as shown in Table 3.

Time efficiency. Under the current paired UAV benchmark, WindINR reduces assimilation time from a median of 8.92 s to 0.63 s per evaluation point, delivering approximately $14\times$ speedup and a 93% reduction in runtime compared with full fine-tuning. This efficiency comes from updating only

Table 3: **Simulated UAV-aided approach.** Query-weighted holdout RMSE in $\mathrm{m\,s^{-1}}$ over eight cases, excluding queries within 50 m of observations. "Improved" counts cases with lower error than the corresponding background.

Method	u	v	w	Mean	Improved
Background	0.4020	0.5051	0.1175	0.3415	–
WindINR	**0.3499**	**0.4117**	**0.1075**	**0.2897**	8/8
Partial FT	0.3843	0.4448	0.1304	0.3198	6/8
Residual IDW	0.3781	0.4467	0.1444	0.3231	5/8
Full FT	0.5198	0.5901	0.1851	0.4317	1/8

the latent state while keeping the network weights frozen, making WindINR particularly suitable for rapid online UAV-field assimilation.

5.3 Real Observations

Tromsø UAV observations and holdouts. We evaluate measured UAV winds collected around Tromsø, Norway, during one episode on 4 September 2026. The coarse physical background uses the Nordic 1 km analysis and AROME-Arctic fields; terrain and roughness are prepared separately. We consider three UAV-derived evaluation protocols: a spatial block, a survey trajectory, and a stopped transect. Each uses 16 aggregated wind measurements for correction, while 12, 19, and 24 separate measurements, respectively, are withheld as verification targets. The spatial block tests queries away from observed locations, whereas the two trajectory protocols use held-out portions of the flights.

Results. We additionally include **Constant** as a trivial baseline. Constant adds a spatially uniform horizontal-wind offset estimated from the observations to the background at every query. WindINR outperforms Constant, Residual IDW, Partial FT, and Full FT on all three protocols (Table 4). Overall, errors in the real-observation experiments are larger than in the simulated settings, reflecting the additional uncertainties and biases present in both the analysis background and UAV measurements. Notably, full fine-tuning becomes substantially less stable under these conditions and can even amplify the error by several times in some cases. In contrast, both Partial FT and WindINR remain considerably more robust, with WindINR consistently achieving better performance than Partial FT. These results suggest that latent-space correction provides a more reliable and effective mechanism for adapting to imperfect real-world observations than direct network fine-tuning. This robustness is particularly encouraging for extending WindINR to more challenging UAV- and helicopter-assisted operations in complex terrain, including potential search-and-rescue applications.

Table 4: **Real-UAV horizontal-wind correction.** Horizontal-vector RMSE in $\mathrm{m\,s^{-1}}$. Bold and underlined values mark the lowest and second-lowest errors among the five displayed estimators. Gain is WindINR's relative improvement over the underlined result (Appendix D.2).

Protocol	N_h	Constant	Residual IDW	Partial FT	Full FT	WindINR	Gain (%)
Tromsø spatial block	12	<u>1.6365</u>	1.7117	1.7389	2.8302	**1.3773**	+15.84
Tromsø survey	19	5.1026	5.1278	<u>5.0533</u>	5.1923	**5.0197**	+0.67
Tromsø stopped transect	24	<u>2.8090</u>	3.8671	2.8396	6.4746	**2.2245**	+20.81

6 Conclusion

WindINR provides continuous local wind estimates by separating reusable terrain-aware representation learning from sample-specific latent-state correction. A shared INR decoder supports arbitrary-coordinate queries, while a deployable predictor initializes compact latent controls from terrain and coarse background winds. Training-time discrepancies between predicted and privileged reference latents define a dataset-adaptive Gaussian prior. At inference, sparse observations update only the latent state, keeping network weights fixed and preserving the continuous query interface. Controlled OSSEs and real UAV observations support this formulation for local wind reconstruction. Future work will assess broader generalization, correlated errors, and end-to-end latency.

AI USE STATEMENT

Generative AI tools assisted with manuscript editing, LaTeX format conversion, preparation of plotting and tabulation scripts, and consistency checks and paired-case audits of the supplied experiment reports. We have reviewed all AI-assisted work. LLM-generated code was verified and tested for correctness by 2 authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

The experiments combine simulated wind fields and synthetic observations with a small measured-UAV evaluation. Wind estimates intended for aviation or other safety-critical applications require independent validation, uncertainty assessment, and appropriate operational safeguards; the present reconstruction scores are not a safety certification.

REPRODUCIBILITY STATEMENT

The formulation and MAP interpretation are given in Section 4 and Appendix A. Appendix B records the case split, selection rule, model configuration, inference settings, and the scope of available evidence. Appendix B.5 adds the density-specific supervised baseline, checkpoint-selection record, and paired evaluation checks. The accompanying source package includes anonymized per-case metric records, a case manifest, and scripts that regenerate the tables and plots. Appendix C adds the mobile protocol, full per-time metric records, terrain arrays, and the distinction between update-time measurements and end-to-end latency. Appendix D records the real-flight protocol roles, the reported evaluation subset, and the limits of the available point-level record. The main real-UAV table is regenerated from the corresponding supplied summaries; no new model inference was performed for this revision.

REFERENCES

Florian Achermann, Thomas Stastny, Bogdan Danciu, Andrey Kolobov, Jen Jen Chung, Roland Siegwart, and Nicholas Lawrance. WindSeer: Real-time volumetric wind prediction over complex terrain aboard a small uncrewed aerial vehicle. *Nature Communications*, 15:3507, 2024. doi: 10.1038/s41467-024-47778-4.

Gérôme Andry, Sacha Lewin, François Rozet, Omer Rochman, Victor Mangeleer, Matthias Pirlet, Elise Faulx, Marilaure Grégoire, and Gilles Louppe. Appa: Bending weather dynamics with latent diffusion models for global data assimilation. In *NeurIPS 2025 Workshop on Machine Learning and the Physical Sciences*, 2025. URL https://ml4physicalsciences.github.io/2025/files/NeurIPS_ML4PS_2025_145.pdf.

Kaifeng Bi, Lingxi Xie, Hengheng Zhang, Xin Chen, Xiaotao Gu, and Qi Tian. Accurate medium-range global weather forecasting with 3D neural networks. *Nature*, 619(7970):533–538, 2023. doi: 10.1038/s41586-023-06185-3.

Kun Chen, Peng Ye, Hao Chen, Kang Chen, Tao Han, Wanli Ouyang, Tao Chen, and Lei Bai. FNP: Fourier neural processes for arbitrary-resolution data assimilation. In *Advances in Neural Information Processing Systems*, volume 37, pp. 137847–137872, 2024.

Rui Chen, Jianfeng Zhang, Yixun Liang, Guan Luo, Weiyu Li, Jiarui Liu, Xiu Li, Xiaoxiao Long, Jiashi Feng, and Ping Tan. Dora: Sampling and benchmarking for 3D shape variational auto-encoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16251–16261, 2025.

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, 2021.

Yago del Valle Inclan Redondo, Enrique Arriaga-Varela, Dmitry Lyamzin, Pablo Cervantes, and Tiago Ramalho. Sparse local implicit image function for sub-km weather downscaling. *arXiv preprint arXiv:2510.20228*, 2025. URL https://arxiv.org/abs/2510.20228.

Hang Fan, Yubao Liu, Yuewei Liu, Zhaoyang Huo, Baojun Chen, and Yu Qin. A novel latent space data assimilation framework with autoencoder-observation to latent space (AE-O2L) network. Part II: Observation and background assimilation with interpretability. *Monthly Weather Review*, 153 (8):1349–1363, 2025. doi: 10.1175/MWR-D-24-0058.1.

Hang Fan, Lei Bai, Ben Fei, Yi Xiao, Kun Chen, Yubao Liu, Yongquan Qu, Fenghua Ling, and Pierre Gentine. Physically consistent global atmospheric data assimilation with machine learning in latent space. *Science Advances*, 12(1):eaea4248, 2026a. doi: 10.1126/sciadv.aea4248.

Hang Fan, Juan Nathaniel, Yi Xiao, Ce Bian, Fenghua Ling, Ben Fei, Lei Bai, and Pierre Gentine. Accurate and efficient hybrid-ensemble atmospheric data assimilation in latent space with uncertainty quantification. *arXiv preprint arXiv:2603.04395*, 2026b. URL https://arxiv.org/abs/2603.04395.

Philipp Hess, Michael Aich, Baoxiang Pan, and Niklas Boers. Fast, scale-adaptive and uncertainty-aware downscaling of Earth system model fields with generative machine learning. *Nature Machine Intelligence*, 7:363–373, 2025. doi: 10.1038/s42256-025-00980-5.

Langwen Huang, Lukas Gianinazzi, Yuejiang Yu, Peter Dominik Dueben, and Torsten Hoefler. DiffDA: a diffusion model for weather-scale data assimilation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 19798–19815. PMLR, 2024.

Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, Alexander Merose, Stephan Hoyer, George Holland, Oriol Vinyals, Jacklynn Stott, Alexander Pritzel, Shakir Mohamed, and Peter Battaglia. Learning skillful medium-range global weather forecasting. *Science*, 382(6677): 1416–1421, 2023. doi: 10.1126/science.adi2336.

Chensen Lin, Ruian Tie, Shihong Yi, Dongqing Liu, Xiaohui Zhong, Zixin Hu, and Hao Li. Reconstructing fine-scale 3D wind fields with terrain-informed machine learning. *Nature Communications*, 17:3713, 2026. doi: 10.1038/s41467-026-70562-5.

Zili Liu, Hao Chen, Lei Bai, Wenyuan Li, Keyan Chen, Zhengyi Wang, Wanli Ouyang, Zhengxia Zou, and Zhenwei Shi. Deriving accurate surface meteorological states at arbitrary locations via observation-guided continuous neural field modeling. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–20, 2024. doi: 10.1109/TGRS.2024.3447073.

Zili Liu, Hao Chen, Lei Bai, Wenyuan Li, Zhengxia Zou, and Zhenwei Shi. Topographic informed Kolmogorov-Arnold neural interpolator for downscaling and correcting meteorological fields from in situ observations. *IEEE Transactions on Geoscience and Remote Sensing*, 63:1–14, 2025. doi: 10.1109/TGRS.2025.3578701.

Xihaier Luo, Wei Xu, Balasubramanya T. Nadiga, Yihui Ren, and Shinjae Yoo. Continuous field reconstruction from sparse observations with implicit neural networks. In *The Twelfth International Conference on Learning Representations*, 2024.

Morteza Mardani, Noah Brenowitz, Yair Cohen, Jaideep Pathak, Chieh-Yu Chen, Cheng-Chin Liu, Arash Vahdat, Mohammad Amin Nabian, Tao Ge, Akshay Subramaniam, Karthik Kashinath, Jan Kautz, and Mike Pritchard. Residual corrective diffusion modeling for km-scale atmospheric downscaling. *Communications Earth & Environment*, 6(1):124, 2025. doi: 10.1038/s43247-025-02042-5.

Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy Networks: Learning 3D reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4460–4470, 2019.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *Computer Vision – ECCV 2020*, volume 12346 of *Lecture Notes in Computer Science*, pp. 405–421. Springer, 2020. doi: 10.1007/978-3-030-58452-8_24.

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 165–174, 2019.

Jaideep Pathak, Shashank Subramanian, Peter Harrington, Sanjeev Raja, Ashesh Chattopadhyay, Morteza Mardani, Thorsten Kurth, David Hall, Zongyi Li, Kamyar Azizzadenesheli, Pedram Hassanzadeh, Karthik Kashinath, and Animashree Anandkumar. FourCastNet: A global data-driven high-resolution weather model using adaptive Fourier neural operators. *arXiv preprint arXiv:2202.11214*, 2022. URL https://arxiv.org/abs/2202.11214.

Ilan Price, Alvaro Sanchez-Gonzalez, Ferran Alet, Tom R. Andersson, Andrew El-Kadi, Dominic Masters, Timo Ewalds, Jacklynn Stott, Shakir Mohamed, Peter Battaglia, Remi Lam, and Matthew Willson. Probabilistic weather forecasting with machine learning. *Nature*, 637:84–90, 2025. doi: 10.1038/s41586-024-08252-9.

Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Advances in Neural Information Processing Systems*, volume 33, pp. 7462–7473, 2020.

Jing-An Sun, Hang Fan, Junchao Gong, Ben Fei, Kun Chen, Fenghua Ling, Wenlong Zhang, Wanghan Xu, Li Yan, Pierre Gentine, and Lei Bai. LO-SDA: Latent optimization for score-based atmospheric data assimilation. *arXiv preprint arXiv:2510.22562*, 2025. URL https://arxiv.org/abs/2510.22562.

Yi Xiao, Qilong Jia, Kun Chen, Lei Bai, and Wei Xue. VAE-Var: Variational autoencoder-enhanced variational methods for data assimilation in meteorology. In *The Thirteenth International Conference on Learning Representations*, 2025.

Xiaoze Xu, Xiuyu Sun, Wei Han, Xiaohui Zhong, Lei Chen, Zhiqiu Gao, and Hao Li. FuXi-DA: A generalized deep learning data assimilation framework for assimilating satellite observations. *npj Climate and Atmospheric Science*, 8(1):156, 2025. doi: 10.1038/s41612-025-01039-3.

Biao Zhang, Jiapeng Tang, Matthias Nießner, and Peter Wonka. 3DShape2VecSet: A 3D shape representation for neural fields and generative diffusion models. *ACM Transactions on Graphics*, 42(4):1–16, 2023. doi: 10.1145/3592442.

A MAP DERIVATION AND THE INFLUENCE OF SPARSE OBSERVATIONS

For the notation below, $b \equiv x^{\mathrm{LR}}$, $x(p) \equiv x^{\mathrm{HR}}(p)$, and $\mu \equiv \mu_{\Delta z}$. The regularized prior covariance is denoted by B_z; the diagonal, scaled covariance used in the FuXi-CFD experiments is specified in Appendix B.

A.1 GAUSSIAN ASSUMPTIONS AND THE CORRECTION OBJECTIVE

Consider one case, with fixed terrain s, background input b, and deployable latent z_0. Assume a latent correction prior $\delta z \sim \mathcal{N}(\mu, B_z)$ with $B_z \succ 0$. Conditional on the latent state, assume observation errors are independent with known covariances $R_n \succ 0$. Then

$$p(\delta z \mid \mathcal{O}, s, b, z_0) \propto p(\delta z) \prod_n p(y_n \mid \delta z, s, b, z_0), \tag{6}$$

$$p(y_n \mid \delta z, s, b, z_0) = \mathcal{N}(M_n \mathcal{F}_\theta(s, b, p_n, z_0 + \delta z), R_n)\,. \tag{7}$$

Taking the negative log and discarding constants independent of δz gives Eq. (5). The quadratic notation is $\|r\|_A^2 = r^\top A r$. If only a subset of wind components is observed, M_n and R_n are restricted to those components. Correlated errors can instead be represented by a joint observation covariance; the independent-noise likelihood is an explicit assumption of the present OSSE.

A correction prior on δz is different from assuming $z \sim \mathcal{N}(0, I)$. The reference-to-background discrepancies characterize errors of the deployable latent predictor in the decoder's learned coordinate system. Their nonzero mean describes a typical correction, and their variances describe relative scales on which latent coordinates may change.

A.2 WHITENING AND SPATIAL CONTROLS

In the global formulation, let $B_z = LL^\top$ and set $\delta z = \mu + La$. Equation (5) becomes

$$\min_a \ \frac{1}{2}\|a\|_2^2 + \frac{1}{2}\sum_n \|M_n \mathcal{F}_\theta(s, b, p_n, z_0 + \mu + La) - y_n\|_{R_n^{-1}}^2\,. \tag{8}$$

Whitening absorbs the prior covariance into the decoder's latent input and avoids explicitly applying an inverse covariance at every objective evaluation. With a diagonal covariance this is a component-wise rescaling, not a rotation learned from cross-coordinate covariance.

The same construction applies to spatial controls. Let η collect coefficients on a control grid, let $\delta z(p; \eta)$ denote their fixed spatial readout, and let their quadratic prior have mean $\bar{\eta}$ and covariance C_η. Define $\eta = \bar{\eta} + L_\eta a$. The observation operator then becomes

$$h(a) = \mathrm{concat}_n \left[M_n \mathcal{F}_\theta\big(s, b, p_n, z_0 + \delta z(p_n; \bar{\eta} + L_\eta a)\big)\right]. \tag{9}$$

A.3 A LOCAL LINEAR VIEW

Stack the observations into y and their covariances into R. Around the whitened prior center, linearize the observation operator as $h(a) \approx h(0) + Ga$, where $G = \partial h / \partial a|_{a=0}$. The local quadratic problem is

$$\begin{aligned} &\min_a \ \frac{1}{2}a^\top a + \frac{1}{2}\|Ga - d\|_{R^{-1}}^2\,, \qquad d = y - h(0), \\ &a^* = (I + G^\top R^{-1} G)^{-1} G^\top R^{-1} d. \end{aligned} \tag{10}$$

For a new query point, let G_q be the Jacobian of the decoded wind with respect to the same controls. Its first-order analysis increment is

$$\widehat{x}_{\mathrm{an}}(q) - \widehat{x}_{\mathrm{prior}}(q) \approx G_q (I + G^\top R^{-1} G)^{-1} G^\top R^{-1} d. \tag{11}$$

Thus observations can change an unobserved query through shared latent sensitivity, not only through geographic proximity. The learned representation determines which field changes are expressible, while the innovation d determines which are supported by the current observations. Directions invisible to the observations remain controlled by the prior.

B FuXi-CFD implementation and evaluation record

B.1 Dataset, split, and metric definitions

Simulation data. FuXi-CFD provides terrain-resolving CFD wind fields over $9\,\mathrm{km} \times 9\,\mathrm{km}$ regions at 30 m horizontal resolution (Lin et al., 2026). Our processed collection contains 12,532 cases, split into 8,772/1,879/1,881 training/validation/test cases. We reconstruct u, v, w, not turbulent kinetic energy. The exported evaluation covers 27 AGL levels from 5 to 214.29 m, not the entire vertical CFD domain. AGL is height above local ground and is distinct from altitude above sea level. The case identifiers are unique and disjoint across the three subsets; the split seed is 42.

The 27 evaluated AGL levels, rounded to two decimals, are

$$5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100,$$
$$106.50, 114.95, 125.94, 140.22, 158.78, 182.91, 214.29 \text{ metres.}$$

Metrics. For the observation-density evaluation, each test case has $N_h = 2,048$ holdout points that are reserved for evaluation rather than used in the observation loss. For component c, we report

$$e_{i,c} = \sqrt{\frac{1}{N_h}\sum_{j=1}^{N_h}(\widehat{x}_{i,c}(q_j) - x_{i,c}(q_j))^2}, \qquad E_c = \frac{1}{N_{\text{test}}}\sum_i e_{i,c}, \quad \overline{E} = \frac{E_u + E_v + E_w}{3}. \tag{12}$$

All errors are in $\mathrm{m\,s^{-1}}$. Thus $\overline{E}$ is a mean of component RMSEs, *not* a vector RMSE or an RMSE pooled over every point. Relative improvement is $100(\overline{E}_{\text{bg}} - \overline{E}_{\text{an}})/\overline{E}_{\text{bg}}$. The case win rate counts strict reductions of the corresponding per-case component mean. Test statistics are recomputed from the per-case records. All primary test means are computed over the 1,881 exported cases, each with 2,048 evaluation points, for a total of 3,852,288 query points per observation density.

B.2 WindINR training and model selection

Model selection and observations. The final WindINR checkpoint is chosen by w RMSE on a fixed 256-case validation subset with 2,048 query points per case. The latest refinement uses four A100 GPUs at most and leaves the 3.55M-parameter architecture unchanged. The test evaluation follows freezing of the selected checkpoint. Synthetic observations sample the CFD field with independent Gaussian noise of standard deviation $(0.5, 0.5, 0.2)\ \mathrm{m\,s^{-1}}$ for (u, v, w). All WindINR observation-density runs use this frozen model, the re-estimated prior, and the same online solver configuration. The supervised comparator is separately trained on the same training split, as detailed in Section 5.

Table 5: WindINR training configuration.

Setting	Value
Terrain / coarse input channels	4 / 3
Terrain / coarse context feature widths	64 / 32
Latent channel dimension	128
Point encoder / decoder / background widths	256 / 256 / 256
Decoder layers / dropout	4 / 0
Reference pooling width	128
Positional encoding frequencies	10
Multi-scale spatial context	Enabled
Pyramid query width	32
Background readout / grid	Spatial / 4×4
Baseline mode	Constant
Reported total / Stage 1 trainable parameters	3,546,308 / 3,546,308
Stage 2 trainable parameters	990,432
Support / query samples per case	8,192 / 8,192
Training batch size	16 cases
Final component loss weights	$1 : 1 : 2$ for $u : v : w$
Stage 2 latent alignment weight	1.0
Maximum concurrent GPUs	4 NVIDIA A100-PCIE-40GB

B.3 Latent-prior estimation and online settings

The prior is re-estimated after the final model is frozen. Its fit split is the 8,772-case training set, using 8,192 high-resolution support points per case; the separate 1,879-case validation set supplies diagnostics only. The export records a shrinkage coefficient of 0.1, numerical regularization 10^{-4}, and 128 latent channels.

Table 6: Observation-density evaluation settings.

Setting	Value
Observed components	u, v, w
Observation-noise standard deviation	$(0.5, 0.5, 0.2)\ \mathrm{m\,s^{-1}}$
Observation counts per case	256, 512, 1,024, 2,048
Independent evaluation points per case	2,048
Prior mode / covariance type	Adaptive / diagonal
Covariance scale / whitening	64 / enabled
Online spatial latent grid	3×3
Optimizer / outer steps	L-BFGS / 8
Learning rate / gradient clipping	1 / 10
Network-weight updates	None

B.4 Background reconstruction

Table 7 gives the component-wise background reconstruction results without local observations. WindINR achieves lower mean and horizontal-component RMSE than FuXi-CFD, while FuXi-CFD has lower vertical-component error.

Table 7: **Background reconstruction without local observations.** Component RMSE and their arithmetic mean in $\mathrm{m\,s^{-1}}$. FuXi-CFD values are from Supplementary Table S1 of Lin et al. (2026), using frequency-domain loss.

Model	u	v	w	Mean
FuXi-CFD	0.5486	0.5473	0.1465	0.4141
WindINR	0.5216	0.5234	0.1704	0.4051

B.5 Density-matched supervised baseline: implementation and training

Offline optimization. All four baseline models use the same 8,772 training cases and 1,879 validation cases as the shared FuXi-CFD split. Each uses seed 0, 16 cases per batch, 8,192 query points per case, a maximum of 20 epochs, learning rate 2×10^{-4}, and weight decay 10^{-4}. Configuration values are hidden width 256, observation width 128, four layers, eight positional-encoding frequencies, zero dropout, IDW power 2, and IDW epsilon 10^{-6}. The raw Stage 2 WindINR background is frozen. The observation count matches the intended test density, and noise standard deviations are $(0.5, 0.5, 0.2)\ \mathrm{m\,s^{-1}}$.

Early stopping uses validation *loss*, patience 5, and minimum improvement 10^{-4}. Table 8 records the last saved checkpoint rather than selecting a later epoch by validation RMSE or test performance. Baseline validation uses the full validation split, whereas WindINR's last refinement is selected using the fixed 256-case subset described above. Both estimators use the same frozen background.

Table 8: Supervised-baseline training record. Loss is the exported validation objective; it is not assigned physical RMSE units.

Observations	Epochs completed	Selected epoch	Val. loss	Val. mean RMSE
256	18	13	0.029364	0.4326
512	20	16	0.028093	0.4224
1,024	19	14	0.026885	0.4124
2,048	17	12	0.025369	0.4012

B.6 Component errors and case-wise comparisons

Table 9: Component-wise comparison on the paired FuXi-CFD test evaluations. RMSE and MAE are in $\mathrm{m\,s^{-1}}$ and averaged over cases. Bold RMSE values are lower within each density pair. "Mean MAE" is the arithmetic mean over u, v, w.

		RMSE				
Obs.	Method	u	v	w	Mean	Mean MAE
256	WindSeer	0.5001	0.4991	0.1615	0.3869	0.2640
256	WindINR	**0.4959**	**0.4905**	**0.1586**	**0.3817**	0.2598
512	WindSeer	0.4882	0.4877	0.1588	0.3783	0.2574
512	WindINR	**0.4773**	**0.4716**	**0.1514**	**0.3668**	0.2502
1,024	WindSeer	0.4799	0.4744	0.1554	0.3699	0.2521
1,024	WindINR	**0.4576**	**0.4523**	**0.1442**	**0.3514**	0.2406
2,048	WindSeer	0.4669	0.4615	0.1515	0.3600	0.2449
2,048	WindINR	**0.4380**	**0.4333**	**0.1379**	**0.3364**	0.2313

The aggregate advantage is present in all three components. At 1,024 observations, the local supervised baseline has $u/v/w$ RMSE $0.4799/0.4744/0.1554$, compared with $0.4576/0.4523/0.1442$ for WindINR. At 2,048 observations the corresponding values are $0.4669/0.4615/0.1515$ and $0.4380/0.4333/0.1379$. These are averages over the same cases, not claims that every query point or case improves.

For case i, define $d_i = \overline{e}_{i,\mathrm{sup}} - \overline{e}_{i,\mathrm{INR}}$, where each $\overline{e}_i$ is the mean of the case's three component RMSEs. Figure 4 plots the empirical distribution of the relative reduction $100 d_i / \overline{e}_{i,\mathrm{sup}}$, and the paired win rate is $N^{-1} \sum_i \mathbf{1}[d_i > 0]$. Rankings in Table 2 use values before rounding. The paired curves and win rates retain WindSeer as their comparator. In contrast, reduction relative to the background compares each method with its own raw score. Table 10 keeps these two notions distinct. At 256 observations, the supervised estimator improves more cases relative to the raw background (89.42% versus 83.41%), even though WindINR has the lower mean error and wins the direct paired comparison on 64.33% of cases. The two robustness summaries need not rank methods identically.

Table 10: Case-wise strict improvement rates. The middle columns compare each estimator with the shared raw background; the last column compares the two estimators directly. No ties count as wins.

Obs.	WindSeer vs. raw	WindINR vs. raw	WindINR vs. supervised
256	89.42%	83.41%	64.33%
512	95.16%	97.24%	78.95%
1,024	94.47%	99.68%	91.71%
2,048	97.98%	100.00%	96.28%

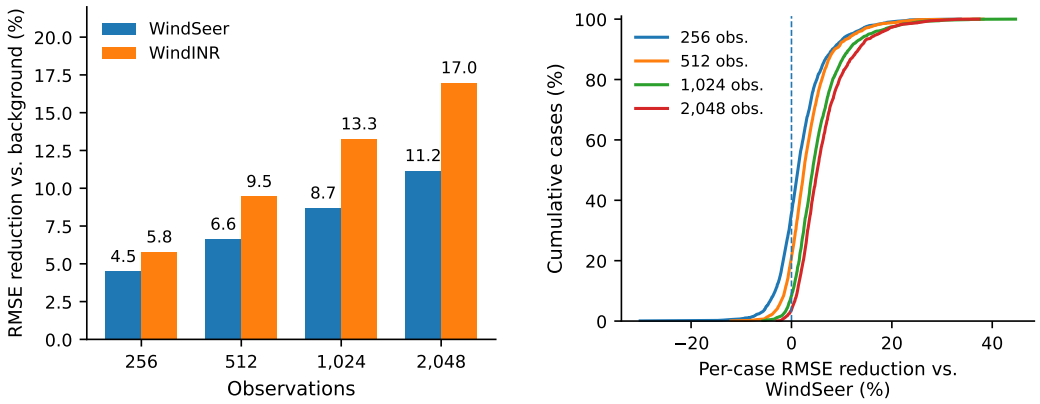


Figure 4: **Same background, two observation-conditioned estimators.** Left: relative reduction of mean component RMSE against the corresponding background. Right: empirical distributions of paired per-case relative RMSE reductions, using each case's WindSeer error as denominator; positive values favor WindINR and negative values favor the supervised baseline. All 1,881 cases are included at each density. These curves describe case variability, not uncertainty over training seeds.

The distributions retain cases where supervised correction is better: WindINR wins on 64.33% of cases at the lowest density and 96.28% at the highest. Mean superiority therefore does not imply a pointwise or per-case guarantee. The comparison concerns the supplied local baseline and shared-background protocol.

C FuXi-CFD mobile-observation experiment: protocols and complete results

This appendix details the simulated UAV-ahead-of-helicopter evaluation in Section 5. It uses the frozen model from the large FuXi-CFD experiment, not a model trained separately for a selected region. Synthetic observations and verification come from the same steady CFD field. The measured-flight experiments remain separate in Appendix D.

C.1 Case selection, checkpoint, and coordinate system

The eight cases are entries 0–7 of the existing 1,881-case test manifest (split seed 42). Their ordered identifiers match the manuscript's test manifest. Selection uses manifest order, not error or improvement. The supplied report states that the route and baseline settings were not tuned on these cases. This is a pilot subset, not a second evaluation of all 1,881 cases or a verified terrain-disjoint split.

The exported DEM and aerodynamic roughness arrays have shape 300×300 with $[y, x]$ indexing. Local coordinates are $x_j = y_j = 30j$ metres for $j = 0, \ldots, 299$, spanning 0–8,970 m. The source-input inventory records `dem`, `roughness`, `u_100m`, and `v_100m`. These field names do not imply 100 m horizontal grid spacing, nor do they specify the model's complete derived-channel preprocessing.

Table 11 summarizes the terrain context for the eight cases; the corresponding maps are shown in Figure 7.

Table 11: Terrain context recomputed from the exported arrays. DEM statistics are input values; they are not inferred wind fields or geographic-position metadata.

Case ID	DEM min (m)	DEM max (m)	Relief (m)	Mean roughness (m)
`001289`	0.0	212.0	212.0	0.685
`006954`	177.0	821.9	644.9	0.837
`000968`	56.2	835.8	779.7	0.905
`000600`	1.5	184.2	182.8	0.526
`012295`	54.0	900.4	846.4	0.945
`002044`	63.0	832.6	769.6	0.789
`011678`	115.0	575.6	460.6	0.972
`005064`	198.0	678.6	480.6	0.621

C.2 Causal sensing and a moving verification region

The route endpoints are $(-0.70, -0.35)$ and $(0.55, 0.25)$ in normalized model coordinates. In local metres they are $(1345.50, 2915.25)$ and $(6951.75, 5606.25)$, giving a 6,218.64 m route. The helicopter and UAV follow the same centerline at 80 m AGL and 20 $\mathrm{m\,s^{-1}}$, with the UAV 500 m ahead until the endpoint clips its progress. Thus the UAV does not remain 500 m ahead after reaching the route end.

The first UAV observation is at 500 m along the route at helicopter time zero. Further observations arrive every 100 m, with an additional endpoint sample, for 59 observations in total. Each includes (u, v, w) with independent Gaussian noise of standard deviation $(0.5, 0.5, 0.2)$ $\mathrm{m\,s^{-1}}$; the base noise seed is 2026. The 64 helicopter evaluations occur at 100 m intervals and the endpoint, covering 0–310.93 s. Only observations with timestamps no later than the current evaluation are available. The observation prefix grows, but every analysis restarts from the background: neither latent controls nor fine-tuned weights are carried forward.

The look-ahead footprint spans 0–800 m along track and ± 150 m across track, with nominal sampling 16×7 at 80 m AGL. Queries beyond the route endpoint are removed; the full query count consequently falls from 112 to 7 near the end. Verification truth is obtained by trilinear interpolation on the native (AGL, y, x) CFD output grid. High-resolution truth is used for synthetic sensing and verification, not as a deployable model input.

The primary score excludes queries within 50 m of any assimilated observation in (x, y, AGL) coordinates. All methods have the same retained count at each case and time. There are 52,640 full-corridor and 47,152 retained query–time evaluations across the eight cases; the retained count ranges from 6 to 110 per record. These are not counts of independent wind realizations or necessarily distinct physical positions. Adjacent corridors overlap, and every time row for a case samples one stationary CFD field.

C.3 Comparator recipes and scope of the comparison

WindINR uses the unchanged 3×3 spatial latent control, adaptive diagonal prior, covariance scale 64, whitening, and L-BFGS with eight outer steps, learning rate 1, and gradient clipping 10. Partial FT uses Adam for 100 steps at learning rate 10^{-4}, updating only `latent_seed` and the final output-head layer with anchor weight 0.01. Full FT uses Adam for 100 steps at 10^{-4} on the full network. Both weight-update recipes specify zero weight decay and no gradient clipping. Residual IDW interpolates observation residuals relative to the same bias-corrected background with inverse-distance exponent 2.

C.4 Aggregation and complete numerical comparison

Let e_{itmc} be the component RMSE for case i, evaluation time t, method m, and component c, and let n_{it} be the query count for the chosen evaluation set. The reported mobile score is

$$E_{mc}^{\mathrm{mobile}} = \frac{\sum_{i,t} n_{it} e_{itmc}}{\sum_{i,t} n_{it}}, \qquad \overline{E}_{m}^{\mathrm{mobile}} = \frac{1}{3} \sum_{c \in \{u,v,w\}} E_{mc}^{\mathrm{mobile}}. \tag{13}$$

This is a query-count-weighted mean of per-record RMSEs, not a pooled pointwise RMSE. Full-corridor and holdout scores use their own counts. Every case has the same geometry and total retained count, so averaging the eight query-weighted case scores yields the same aggregate. A case is improved when its weighted mean component RMSE is below its background score; time-step win counts are descriptive only, since the times are correlated.

Table 12: **Full-corridor sensitivity evaluation.** Component RMSE, mean component RMSE, and mean component MAE in $\mathrm{m\,s^{-1}}$. These scores include nearby queries removed by the primary 50 m exclusion. Primary holdout RMSE is reported only in Table 3.

Evaluation	Method	u	v	w	Mean	MAE
Full corridor	Background	0.4019	0.5058	0.1171	0.3416	0.2930
Full corridor	WindINR	0.3438	0.4024	0.1054	0.2839	0.2355
Full corridor	Partial FT	0.3756	0.4315	0.1272	0.3114	0.2560
Full corridor	Residual IDW	0.3860	0.4471	0.1494	0.3275	0.2668
Full corridor	Full FT	0.5131	0.5760	0.1831	0.4241	0.3410

The holdout reductions relative to the bias-corrected background are 15.17% for WindINR, 6.35% for Partial FT, and 5.41% for Residual IDW; Full FT increases error by 26.40%. WindINR reduces $u/v/w$ RMSE by 12.94/18.49/8.48%, whereas Partial FT and Residual IDW worsen the vertical component. The aggregate ordering is unchanged when nearby queries are restored (Table 12), but this does not remove the need for the exclusion test.

C.5 ALONG-ROUTE ERROR FOR EVERY CASE

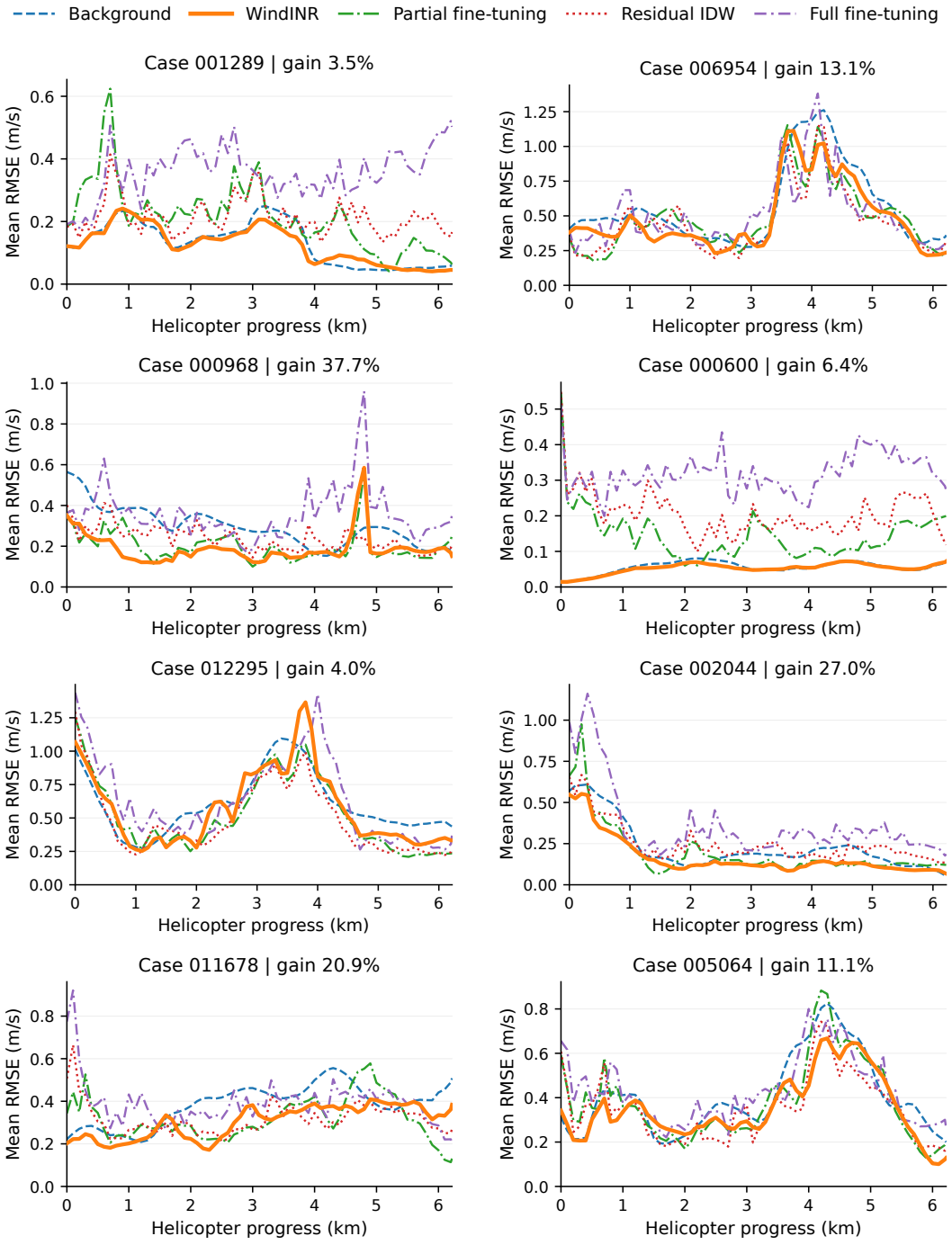


Figure 5: **Complete per-case trajectory diagnostics.** All five methods, all eight cases, and every recorded time step are shown. The primary 50 m holdout metric is used throughout. Vertical scales differ by case to preserve within-case detail.

C.6 Query budget, update cost, and reproducibility limits

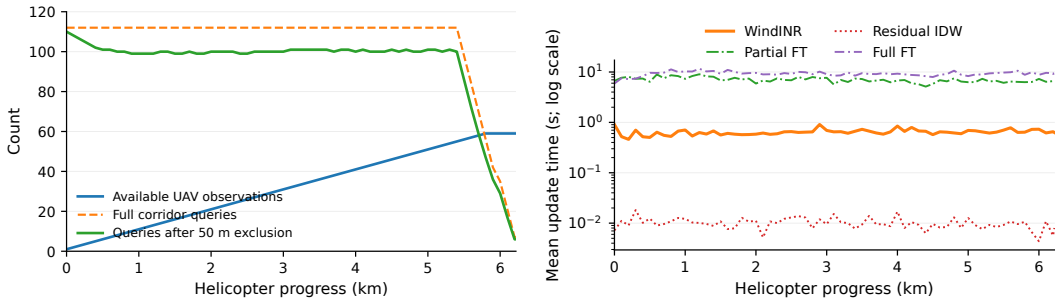


Figure 6: **Available information and measured update cost.** Left: the observation prefix grows to 59 samples while the corridor is clipped near the endpoint; all cases share these counts. Right: mean reported update time across cases at each progress point on a logarithmic axis.

Table 13: Reported update-time distribution across the 512 case–time records (seconds). These values depend on the fixed solver budgets and the code's timing boundaries.

Method	Mean	Median	90th percentile
WindINR	0.643	0.631	0.955
Partial FT	6.995	6.876	10.143
Residual IDW	0.010	0.006	0.024
Full FT	9.260	8.918	13.061

The mean update times are 0.643 s for WindINR, 6.995 s for Partial FT, 9.260 s for Full FT, and 0.010 s for Residual IDW. Optimization-based timers exclude common corridor decoding; Residual IDW uses a separate interpolation-path timer. Four two-case shards ran on separate NVIDIA A100-PCIE-40GB GPUs, with at most four GPUs active. Reported wall time was 38.5 minutes, summed GPU process time 2.42 GPU-hours, and peak allocated memory 968 MiB; independent repeated timing trials are unavailable.

The pilot supports mobile observation-driven correction using an already trained representation. Its remaining scope limits are eight cases, one route geometry, a single altitude and lead distance, one noise realization per case, and stationary simulation truth. Correlated route samples, meteorological evolution, different sensing patterns, and out-of-distribution terrains require separate tests.

C.7 TERRAIN MAPS FOR ALL EIGHT CASES

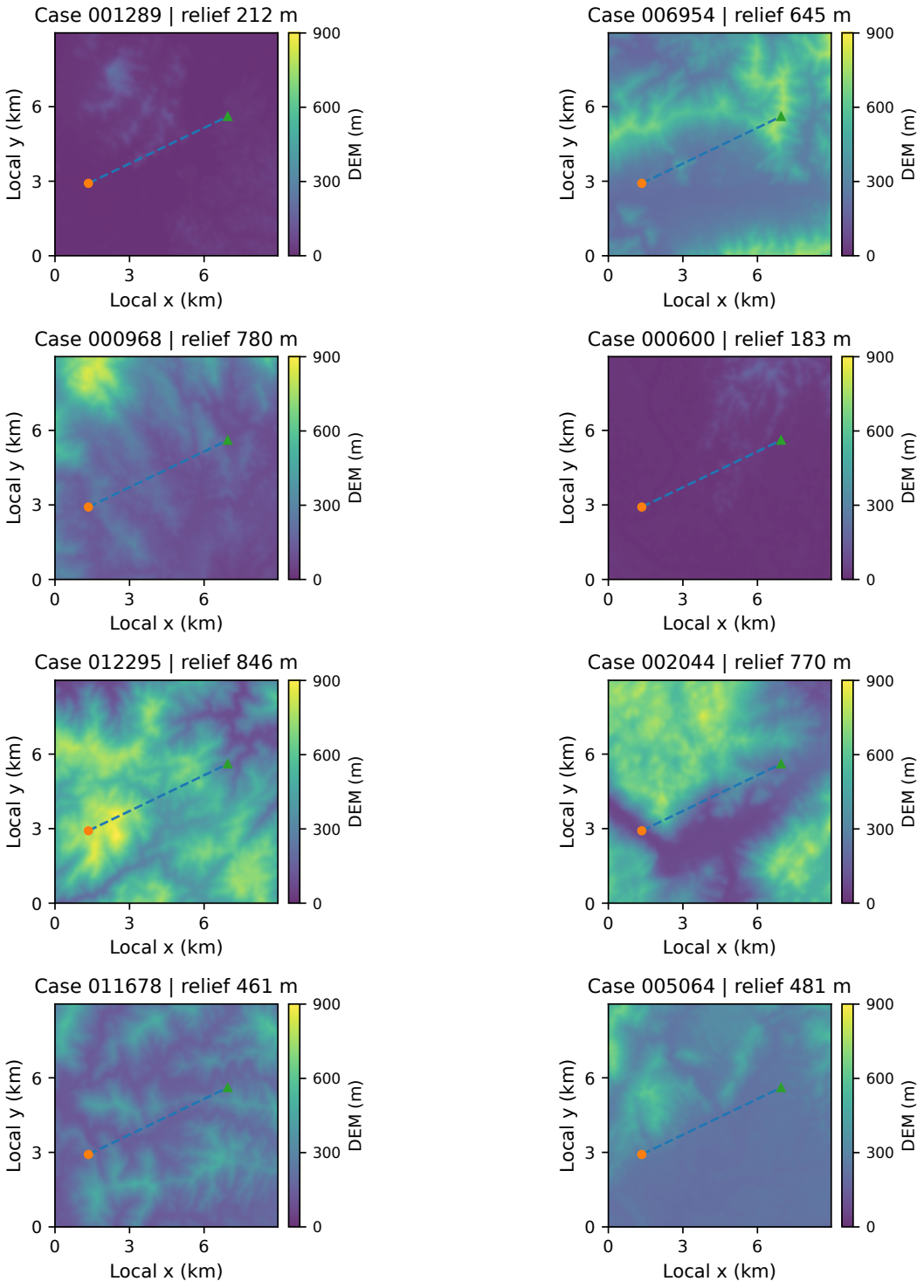


Figure 7: **The eight FuXi-CFD terrain arrays.** DEM values use a shared color scale; axes are local kilometres. Dashed lines show the same normalized flight route in each case; circles and triangles mark its start and end. Terrain relief is the array maximum minus minimum.

D REAL-UAV OBSERVATIONS: PROTOCOLS AND SUPPLEMENTARY DIAGNOSTICS

This appendix supports the three-protocol Tromsø evaluation in Section 5.3: one Tromsø spatial-block test, and two retrospective Tromsø prefix/suffix tests.

D.1 DATA SOURCES AND PREPARATION

Tromsø episode. The evaluated measurements were collected on 4 September 2026 and comprise a horizontal survey and a transect with 15 s stops. We report one spatial-block test and two prefix/suffix protocols (survey and stopped transect). Timestamps are treated as UTC; acquisition and preparation details are retained in the accompanying manifest.

Available background products are the Nordic 1 km postprocessed 10 m analysis and AROME-Arctic fields at 10, 80, and 120 m. The terrain preparation uses Copernicus GLO-30 resampled to a 10 m output grid, in EPSG:25833, plus WorldCover-derived roughness. The resampled grid spacing is not the native DEM resolution.

D.2 HOLDOUTS, METRICS, AND CORRECTION SETTINGS

Each unit uses 16 observation super-observations. The Tromsø spatial block has 12 holdout super-observations, and the survey and stopped-transect protocols have 19 and 24, respectively.

The Tromsø spatial geometry was recorded as locked before its measurements were first scored. Its reported nearest-observation distances have minimum 107.4 m and median 151.3 m. The other Tromsø tests are retrospective, and the stopped-transect suffix consists entirely of revisits within 50 m of observations.

The primary score is the horizontal-vector RMSE on the held-out measurements,

$$R_{uv} = \sqrt{\frac{1}{N_h}\sum_{j=1}^{N_h}[(\widehat{u}_j - u_j)^2 + (\widehat{v}_j - v_j)^2]} = \sqrt{R_u^2 + R_v^2}. \tag{14}$$

This is not the arithmetic mean of component RMSEs in Eq. (12). For the four baselines $\mathcal{B} = \{\text{Constant}, \text{Residual IDW}, \text{Partial FT}, \text{Full FT}\}$ displayed in Table 4, define

$$R_{\text{best}} = \min_{m\in\mathcal{B}} R_m, \qquad g = 100\,\frac{R_{\text{best}} - R_{\text{WindINR}}}{R_{\text{best}}}. \tag{15}$$

Positive g means lower error than all four baselines. Constant is the strongest comparator on the spatial block and stopped transect, while Partial FT is strongest on the survey; the corresponding gains are 15.84%, 20.81%, and 0.67%, respectively.

Correction and fine-tuning settings. All methods use the same 16 aggregated u, v observations and the same held-out queries. Constant adds one observation-derived horizontal-wind offset to the frozen background at every query. Residual IDW interpolates the observation residuals with inverse-distance exponent 2, horizontal scale 100 m, and vertical scale 50 m. Partial FT updates `decoder.latent_seed` and the final output-head layer, with parameter-anchor weight 0.01. Full FT updates all network parameters. Both use Adam for 100 steps at learning rate 10^{-4}, with zero weight decay and no gradient clipping. Only the observation points are used for fitting; held-out measurements are not used for fitting or model selection.