FlowGINO: Continuous Reconstruction from Sparse Observations along with Aleatoric and Epistemic Uncertainty Estimation

Alif Bin Abdul Qayyum[1] **Byung-Jun Yoon**[1,2]
[1]Department of Electrical and Computer Engineering, Texas A&M University
[2]Computing and Data Sciences, Brookhaven National Laboratory

Abstract

Mapping out physical fields in continuous domain from sparse sensor data observations is a difficult challenge and an active research endeavor in many scientific fields. Since the processes that create this data are often not fully understood, there is increasing interest in leveraging deep neural networks to address this problem. Despite significant progress in using deep learning methods like Implicit Neural Representations (INRs) and Fourier Neural Operators (FNOs) for reconstructing physical fields, there remains a notable gap in research on quantifying the uncertainty of these reconstructions. For high-stakes applications, such as climate modeling, it is critically important to estimate and disentangle the two types of uncertainty: reducible *Epistemic Uncertainty* and irreducible *Aleatoric Uncertainty*. Effectively quantifying and separating these uncertainties is essential for the reliable application of deep learning models across scientific domains. We introduce **Flow**-matching **G**eometry **I**nformed **N**eural **O**perator (**FlowGINO**) which is not only capable of reconstructing continuous physical fields but also capable of estimating both epistemic and aleatoric uncertainty in a disentangled manner.

1 Introduction

Reconstructing complex physical fields from sparse sensor networks presents a fundamental challenge across various scientific and engineering disciplines. Despite the widespread necessity of, either knowledge on continuous physical fields or high resolution discrete observations, in areas like geophysics [52], astronomy [18], biochemistry [76], and fluid mechanics [11], sparse sampling still often remains the only practical solution for data acquisition. Accurate representation of climate variables, e.g. temperature, salinity, humidity, and wind/current velocity relies heavily on successful reconstruction of data from a limited number of observations.

Scientific data representing complex physical systems [41, 10] presents distinct challenges that differentiate it from conventional image and video data, due to sensor mobility and other physical constraints for high resolution data acquisition. Traditional physics-based methods rely on partial differential equations (PDEs) derived from fundamental physical principles [26]. However, these models often lack the accuracy and computational efficiency needed for complex systems like weather [4] and epidemiology [42], and integrating real-world data for calibration is challenging [49]. Machine learning-based approaches, specifically Implicit Neural Representations [8, 47, 44, 45, 59, 39] and Fourier Neural Operators [35, 31, 48, 72, 37, 77, 36], have concurrently emerged as an alternative for nonlinear field reconstruction. Multiplicative and Modulated Gabor Network (MMGN) [39] uses INR (Implicit Neural Representation) to create a continuous representation. Geometry-Informed Neural Operator (GINO) [37] integrates both Graph Neural Operator [34] and

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: MLxOR: Mathematical Foundations and Operational Integration of Machine Learning for Uncertainty-Aware Decision-Making.

Fourier Neural Operator [35] to reconstruct the field in a continuous space. Several variants of Fourier Neural Operators have also been explored for similar research questions [77, 72].

Although these approaches excel at continuous reconstruction, they lack the inherent ability to quantify uncertainty. For mission-critical applications like weather forecasting and climate modeling, reliable predictions are paramount. Therefore, quantifying uncertainty is essential to improve model robustness and avoid catastrophic consequences.

The introduction of Diffusion [61, 62, 22, 63] and Flow Matching models [38, 1] has spurred a paradigm shift in generative modeling, driving remarkable progress in a wide range of research areas, most notably in computer vision [13, 53, 50, 54, 38, 73, 56]. There has also been a surge of interest in applying diffusion and flow matching models for tasks that require robust uncertainty quantification [9, 17, 3, 28, 46].

Distinguishing between aleatoric and epistemic uncertainty is crucial for a useful uncertainty estimate [29, 19]. While aleatoric uncertainty reflects the inherent noise and ambiguity of a problem, epistemic uncertainty highlights what the model doesn't know, which can be reduced with more data [12, 27]. Separating these two types offers valuable insights for improving a model and can guide practical decisions in climate modeling, such as where to place new sensors for better data collection [6].

We introduce **Flow**-matching **G**eometry **I**nformed **N**eural **O**perator (**FlowGINO**) for continuous reconstruction of physical fields from sparse observations. Our specific contributions are summarized below:

- We propose a novel architecture that integrates both *FNO* [35, 31] and *Flow-matching* [38] to achieve two different objectives simultaneously:
    1. Continuous reconstruction of physical field from arbitrary and sparse observations.
    2. Estimating epistemic and aleatoric uncertainty separately with one single model.

2 Problem Statement

The objective is to precisely recreate a physical field, $\mathbf{u}(x,t)$. This field changes over both space (x) and time (t), and it can represent climate variables like temperature, velocity, or displacement. We only have a few observations of $\mathbf{u}$ at specific locations $(x)_i$ and at specific moments in time $(t)_i$. We define these observations as the training data $\mathcal{D}_{train} = \{\mathbf{u}((x)_i, (t)_i)\}$ where $\{((x)_i, (t)_i)\} \in T_{train}$ consists the set of locations and timepoints for which we have the observations in the training data $\mathcal{D}_{train}$. For, any time point $(t)_{test}$ and with a few sparse observations $\mathbf{u}_{so} = \{\mathbf{u}((x)_{test}, (t)_{test})\}$ where $(x)_{so} = \{(x)_{test}\}$ consists some sparse locations, the goal is to obtain a predictive distribution:

$$p\left(\mathbf{u}((x)_c)|\mathbf{u}_{so}, (t)_{test}, \mathcal{D}_{train}\right) = \int p\left(\mathbf{u}((x)_c)|\mathbf{u}_{so}, (t)_{test}, \phi\right) p(\phi|\mathcal{D}_{train}) d\phi \tag{1}$$

where $\mathbf{u}((x)_c)$ indicates a simplified representation of $\mathbf{u}((x)_c, (t)_{test})$, $(x)_c$ represents any location in the coninuous Eucledian space, $p\left(\mathbf{u}((x)_c)|\mathbf{u}_{so}, (t)_{test}, \phi\right)$ represents the likelihood function, and $p(\phi|\mathcal{D}_{train})$ represents the posterior over model parameters ϕ for $\mathcal{D}_{train}$. Uncertainty in $p\left(\mathbf{u}((x)_c)|\mathbf{u}_{so}, (t)_{test}, \mathcal{D}_{train}\right)$ stems from two distinct sources: aleatoric uncertainty and epistemic uncertainty [15].

3 Methodology

Our proposed **FlowGINO** mainly consists of two components: (i) *GINO(Geometry Informed Neural Operator)* [37], g_θ and (ii) *Flow-matching UNET* [38], f_θ.

Given the set of sparse observations $\mathbf{u}_{so}$ at timepoint $(t)_{test}$, the *GINO* component, g_θ generates a context $\hat{\mathbf{u}}_{lr}$ for the locations $\{(x)_c\}$ in the high resolution frame.

$$\hat{\mathbf{u}}_{lr} = g_\theta(\mathbf{u}_{so}, (t)_{test}, \{(x)_c\}) \tag{2}$$

With the context $\hat{\mathbf{u}}_{lr}$ and a random noise n, the *Flow-matching UNET*, f_θ obtains the high resolution output $\hat{\mathbf{u}}_{hr}$ for the locations $\{(x)_c\}$ in the high resolution frame.

$$\hat{\mathbf{u}}_{hr} = f_\theta(\{\hat{\mathbf{u}}_{lr}, \mathbf{n}\}) \tag{3}$$

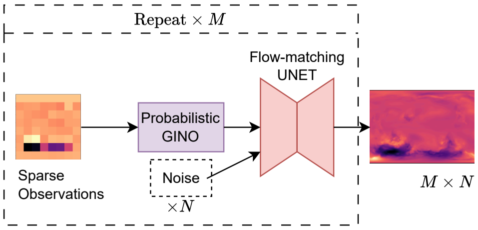

Figure 1: FlowGINO.

Geometry Informed Neural Operator: The *GINO* component, g_θ, primarily consists of three main subcomponents: (i) *Geometry Encoder*: Multiple local kernel integration layers through GNO (Graph Neural Operator) [34], (ii) *Global Model*: Sequence of FNO (Fourier Neural Operator) [35] layers for global kernel integration, and (iii) *Geometry Decoder*: Final kernel integral layers through GNO (Graph Neural Operator) [34]. The sparse observations $\mathbf{u}_{so}$ and their corresponding geometries, $\{(x)_{so}, t\}$ are passed on to the GNO encoder layers, which maps them to a latent regular grid. This is then passed into an Global FNO model. The output from the FNO model is projected back onto the domain of the input geometry for each query point, $\{(x)_c, t\}$ in the high resolution frame using the GNO decoder layer.

Flow-matching UNET: The Flow-matching UNET model is a standard U-shaped Neural Network for 2D grid-like data. The concatenated context $\hat{\mathbf{u}}_{lr}$ and random noise $\mathbf{n}$, along with the noise scale n_t is passed onto this UNET and the high resolution output $\hat{\mathbf{u}}_{hr}$ is extracted from this model.

3.1 Optimization of Model Weights

For probabilistic adaptation of the model, we deploy two approaches: (i) MCD (Monte-Carlo Dropout) [19], and (ii) VBLL (Variational Bayesian Last Layer) [21] implementation onto the GINO component, g_θ. Let, $\mathbf{u}_{so}(t)$ be the sparse observations for any timepoint, t and $\mathbf{u}(t)$ be the high resolution ground truth. The optimization objective function is to minimize:

$$\mathcal{L}_{fm}(\theta) = \mathbb{E}_{\mathbf{u}\sim p(\mathbf{u})}\mathbb{E}_{\mathbf{n}\sim\mathcal{N}(0,1)}\mathbb{E}_{n_t\sim U(0,1)}\left[\|f_\theta\left(\{g_\theta(\mathbf{u}_{so}), (1-n_t)\mathbf{n} + n_t\mathbf{u}\}, n_t\right) - (\mathbf{u}-\mathbf{n})\|^2\right] \tag{4}$$

where $g_\theta(\mathbf{u}_{so})$ is a simplified representation of the low-resolution context output of g_θ from sparse observations, $g_\theta(\mathbf{u}_{so}, (x)_{so}, t)$ with $(x)_{so}$ be the set of locations of those sparse observations.

Optimization of MCD Adaptation: We apply dropout layer in the *FNO* layers of the GINO component g_θ of our model. The optimization objective is $\mathcal{L}_{fm}$ mentioned in Equation 4.

Optimization of VBLL Adaptation: We append an additional Variational Bayesian linear layer [21] at the end of GINO component g_θ. Considering b_θ the Bayesian Layer, the output of the GINO component is:

$$\hat{\mathbf{u}}_{\mathbf{lr}} = b_\theta^T g_\theta(\mathbf{u}_{so}, (x)_{so}, t) + \epsilon, \epsilon \sim \mathcal{N}(0, \Sigma), b_\theta \sim \mathcal{N}(\bar{b}, S) \tag{5}$$

And the predictive distribution of the GINO component becomes:

$$p(\hat{\mathbf{u}}_{\mathbf{lr}}|\mathbf{u}_{so}, \bar{b}, S, g_\theta) = \mathcal{N}(\bar{b}^T g_\theta(\mathbf{u}_{so}, \cdot), g_\theta(\mathbf{u}_{so}, \cdot)^T S g_\theta(\mathbf{u}_{so}, \cdot) + \Sigma) \tag{6}$$

Following the optimization objective to minimize $\mathcal{L}_{vbll}(g_\theta, \bar{b}, S, \Sigma)$ for regression problems mentioned in Harrison et al. [21], the objective function to optimize the overall model is to minimize:

$$\mathcal{L}_{fmvbll} = \alpha\mathcal{L}_{fm} + \beta(\mathcal{L}_{vbll}) \tag{7}$$

where α and β are hyperparameters to balance the two losses.

3.2 Estimation of Epistemic & Aleatoric Uncertainty

Uncertainty is measured using variance and the law of total variance [12, 64, 55] is applied to decompose total uncertainty (TU) across model predictions into its AU and EU components.

$$\text{Var}(\hat{\mathbf{u}}) = \underbrace{\text{Var}_{\theta\sim p(\theta|\mathcal{D}_{train})}\left[\mathbb{E}_{\hat{\mathbf{u}}\sim p(\mathbf{u}|\mathbf{u}_{so},\theta)}\left[\hat{\mathbf{u}}\right]\right]}_{\text{EU}} + \underbrace{\mathbb{E}_{\theta\sim p(\theta|\mathcal{D}_{train})}\left[\text{Var}_{\hat{\mathbf{u}}\sim p(\mathbf{u}|\mathbf{u}_{so},\theta)}\left[\hat{\mathbf{u}}\right]\right]}_{\text{AU}} \tag{8}$$

Inspired by the HyperDiffusion method [6], we first generate one sample of the context $\hat{\mathbf{u}}_{lr}$ from the predictive distribution and then add N different noise samples to it to get N different predictions by passing the noise concatenated samples to f_θ. We repeat this for M number of times by generating M different samples of $\hat{\mathbf{u}}_{lr}$ from the predictive distribution obtained from g_θ. Applying 8 to the generated samples $[\hat{\mathbf{u}}] = \{\cdots, \hat{\mathbf{u}}_{i,j}, \cdots, \hat{\mathbf{u}}_{M,N}\}$, the decomposed aleatoric and epistemic uncertainty can be measured.

$$\widehat{AU} = \mathbb{E}_{i \in M} \left[\mathrm{Var}_{j \in N} \left[\hat{\mathbf{u}} \right] \right] \tag{9}$$

$$\widehat{EU} = \mathrm{Var}_{i \in M} \left[\mathbb{E}_{j \in N} \left[\hat{\mathbf{u}} \right] \right] \tag{10}$$

The total uncertainty can be measured by $\widehat{TU} = \widehat{AU} + \widehat{EU}$. Also the ensemble prediction can be computed by taking the expectation over the predicted samples as $\mathbb{E}_{i \in M} \left[\mathbb{E}_{j \in N} \left[\hat{\mathbf{u}} \right] \right]$. Figure 1 describes the overall methodology of FlowGINO model.

4 Experiments

4.1 Experimental Setup

Dataset: We have utilized The Community Earth System Model version 2 (CESM2) dataset [10], a global climate model to simulate Earth's climate. We followed the similar processing employed in Luo et al. [39]. The dataset dimensions are 1212 (time), 192 (lat), and 288 (lon).

Training & Evaluation: The training dataset consists of data from 102 timepoints, evenly spaced in the total 1212 timepoints. At test phase, we downsampled the data into a dimension of 25 (lat), 37 (lon) for input as the sparse observation, and a dimension of 97 (lat), 145 (lon) for the high resolution frame. We set $M = 32, N = 8$ for both adaptations of our proposed FlowGINO model. We set droptout rate to 0.25 for the MCD adaptation. For evaluation of the constructed high resolution field, we employed four metrics: PSNR [23], SSIM [67], LPIPS [74] and CRPS [43].

Baselines: For comparative evaluation, we utilized MMGN [39] and GINO [37] as baselines. We also employed a VBLL-GINO method for ablation study, where we appended VBLL [21] to GINO [37] model. For deployment of the MMGN model in the test dataset, we applied linear interpolation of the learned latents, following the suggestions mentioned in Luo et al. [39] for forecasting.

4.2 Results

Table 1 demonstrates the results on both the train and test dataset. Except for the LPIPS metric in test dataset, our proposed FlowGINO with MCD adaptation performed the best among all the models. The VBLL adaptation performed better than baselines for both PSNR and SSIM metrics in the test dataset. All models performed much worse on the test set compared to train set, specifically the MMGN model. In terms of the CRPS metric, the MCD adaptation of FlowGINO obtained better results in train set and the VBLL adaptation obtained better result for test set.

Table 1: Evaluation Results

Model	Train Set				Test Set			
	PSNR (↑)	SSIM (↑)	LPIPS (↓)	CRPS (↓)	PSNR (↑)	SSIM (↑)	LPIPS (↓)	CRPS (↓)
MMGN	46.7425	0.9738	0.0259	-	28.1412	0.4236	**0.1849**	-
GINO	41.1518	0.9419	0.1887	-	34.2361	0.6888	0.3928	-
GINO-VBLL	40.3315	0.9269	0.2222	-	33.5479	0.629	0.3968	-
FlowGINO* (MCD)	**53.3361**	**0.9953**	**0.0055**	**0.0024**	**35.5939**	**0.7763**	0.1856	0.0206
FlowGINO* (VBLL)	47.8042	0.9824	0.0416	0.0043	35.3303	0.7543	0.2741	**0.0202**

5 Conclusion

In this work, we propose FlowGINO, a novel Neural Operator model integrated with Flow-matching for continuous reconstruction of physical fields from sparse and arbitrary observations. We compared our model with two of the most suitable existing baselines, one based on Implicit Neural Representations and the other based on Neural Operator. Empirical evaluations demonstrate the superiority of our model over the baselines. Our model is also capable of estimating both epistemic and aleatoric uncertainty with only one single model. We employ two Bayesian Neural Network approaches for probabilistic adaptation of the model. We leave other sophisticated adaptation to Bayesian Neural Networks or other probabilistic models as future work.

Funding Statement

This work was supported in part by the Department of Energy (DOE) award DE-SC0012704.

References

[1] M. S. Albergo and E. Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://arxiv.org/abs/2209.15571`.

[2] R. Angell and D. R. Sheldon. Inferring latent velocities from weather radar data using gaussian processes. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL `https://proceedings.neurips.cc/paper_files/paper/2018/file/23451391cd1399019fa0421129066bc6-Paper.pdf`.

[3] L. Berry, A. Brando, and D. Meger. Shedding light on large generative networks: Estimating epistemic uncertainty in diffusion models. In *The 40th Conference on Uncertainty in Artificial Intelligence*, 2024. URL `https://openreview.net/forum?id=512IkGDqA8`.

[4] S. L. Brunton, J. L. Proctor, and J. N. Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016. doi: 10.1073/pnas.1517384113. URL `https://www.pnas.org/doi/abs/10.1073/pnas.1517384113`.

[5] C. Bülte, P. Scholl, and G. Kutyniok. Probabilistic neural operators for functional uncertainty quantification. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL `https://openreview.net/forum?id=gangoPXSRw`.

[6] M. Chan, M. Molina, and C. Metzler. Estimating epistemic and aleatoric uncertainty with a single model. *Advances in Neural Information Processing Systems*, 37:109845–109870, 2024.

[7] Y. Chen, S. Liu, and X. Wang. Learning continuous image representation with local implicit image function. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8628–8638, 2021.

[8] Z. Chen and H. Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5939–5948, 2019.

[9] E. S. Crafts and U. Villa. Can diffusion models provide rigorous uncertainty quantification for bayesian inverse problems? *arXiv preprint arXiv:2503.03007*, 2025.

[10] G. Danabasoglu, J. Lamarque, J. Bacmeister, D. A. Bailey, A. K. DuVivier, J. Edwards, L. K. Emmons, J. Fasullo, R. Garcia, A. Gettelman, C. Hannay, M. M. Holland, W. G. Large, P. H. Lauritzen, D. M. Lawrence, J. T. M. Lenaerts, K. Lindsay, W. H. Lipscomb, M. J. Mills, R. Neale, K. W. Oleson, B. Otto-Bliesner, A. S. Phillips, W. Sacks, S. Tilmes, L. van Kampenhout, M. Vertenstein, A. Bertini, J. Dennis, C. Deser, C. Fischer, B. Fox-Kemper, J. E. Kay, D. Kinnison, P. J. Kushner, V. E. Larson, M. C. Long, S. Mickelson, J. K. Moore, E. Nienhouse, L. Polvani, P. J. Rasch, and W. G. Strand. The community earth system model version 2 (cesm2). *Journal of Advances in Modeling Earth Systems*, 12(2), Feb. 2020. ISSN 1942-2466. doi: 10.1029/2019ms001916. URL `http://dx.doi.org/10.1029/2019MS001916`.

[11] Y. Deng, H.-X. Yu, J. Wu, and B. Zhu. Learning vortex dynamics for fluid inference and prediction. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=nYWqxUwFc3x`.

[12] S. Depeweg, J.-M. Hernandez-Lobato, F. Doshi-Velez, and S. Udluft. Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning. In *International conference on machine learning*, pages 1184–1193. PMLR, 2018.

[13] P. Dhariwal and A. Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[14] P. Dimitrakopoulos, G. Sfikas, and C. Nikou. Implicit neural representation inference for low-dimensional bayesian deep learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=5KUiMKRebi`.

[15] C. Ekmekci and M. Cetin. Uncertainty quantification for deep unrolling-based computational imaging. *IEEE Transactions on Computational Imaging*, 8:1195–1209, 2022. doi: 10.1109/TCI.2022.3233185.

[16] R. Everson and L. Sirovich. Karhunen–loève procedure for gappy data. *J. Opt. Soc. Am. A*, 12(8):1657–1664, Aug 1995. doi: 10.1364/JOSAA.12.001657. URL `https://opg.optica.org/josaa/abstract.cfm?URI=josaa-12-8-1657`.

[17] M. A. Finzi, A. Boral, A. G. Wilson, F. Sha, and L. Zepeda-Núñez. User-defined event sampling and uncertainty quantification in diffusion models for physical dynamical systems. In *International Conference on Machine Learning*, pages 10136–10152. PMLR, 2023.

[18] H. Gabbard, C. Messenger, I. S. Heng, F. Tonolini, and R. Murray-Smith. Bayesian parameter estimation using conditional variational autoencoders for gravitational-wave astronomy. *Nature Physics*, 18(1):112–117, 2022.

[19] Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In M. F. Balcan and K. Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA, 20–22 Jun 2016. PMLR. URL `https://proceedings.mlr.press/v48/gal16.html`.

[20] S. Gao, X. Liu, B. Zeng, S. Xu, Y. Li, X. Luo, J. Liu, X. Zhen, and B. Zhang. Implicit diffusion models for continuous super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10021–10030, 2023.

[21] J. Harrison, J. Willes, and J. Snoek. Variational bayesian last layers. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=Sx7BIiPzys`.

[22] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[23] A. Horé and D. Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th International Conference on Pattern Recognition*, pages 2366–2369, 2010. doi: 10.1109/ICPR.2010.579.

[24] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[25] X. Hu, H. Mu, X. Zhang, Z. Wang, T. Tan, and J. Sun. Meta-sr: A magnification-arbitrary network for super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1575–1584, 2019.

[26] T. J. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Courier Corporation, 2012.

[27] E. Hüllermeier and W. Waegeman. Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. *Machine Learning*, 110(3):457–506, Mar. 2021. ISSN 1573-0565. doi: 10.1007/s10994-021-05946-3. URL `http://dx.doi.org/10.1007/s10994-021-05946-3`.

[28] M. Jazbec, E. Wong-Toi, G. Xia, D. Zhang, E. Nalisnick, and S. Mandt. Generative uncertainty in diffusion models. In *The 41st Conference on Uncertainty in Artificial Intelligence*, 2025. URL `https://openreview.net/forum?id=UcfIMqIHNt`.

[29] A. Kendall and Y. Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017.

[30] B. Kim, V. C. Azevedo, N. Thuerey, T. Kim, M. Gross, and B. Solenthaler. Deep fluids: A generative network for parameterized fluid simulations. *Computer Graphics Forum*, 38(2): 59–70, May 2019. ISSN 1467-8659. doi: 10.1111/cgf.13619. URL `http://dx.doi.org/10.1111/cgf.13619`.

[31] N. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, and A. Anandkumar. Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research*, 24(89):1–97, 2023.

[32] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL `https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf`.

[33] K. Lee and K. T. Carlberg. Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. *Journal of Computational Physics*, 404:108973, Mar. 2020. ISSN 0021-9991. doi: 10.1016/j.jcp.2019.108973. URL `http://dx.doi.org/10.1016/j.jcp.2019.108973`.

[34] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar. Neural operator: Graph kernel network for partial differential equations. *arXiv preprint arXiv:2003.03485*, 2020.

[35] Z. Li, N. B. Kovachki, K. Azizzadenesheli, B. liu, K. Bhattacharya, A. Stuart, and A. Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=c8P9NQVtmnO`.

[36] Z. Li, D. Z. Huang, B. Liu, and A. Anandkumar. Fourier neural operator with learned deformations for pdes on general geometries. *Journal of Machine Learning Research*, 24(388):1–26, 2023.

[37] Z. Li, N. B. Kovachki, C. Choy, B. Li, J. Kossaifi, S. P. Otta, M. A. Nabian, M. Stadler, C. Hundt, K. Azizzadenesheli, and A. Anandkumar. Geometry-informed neural operator for large-scale 3d PDEs. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=86dXbqT5Ua`.

[38] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=PqvMRDCJT9t`.

[39] X. Luo, W. Xu, B. Nadiga, Y. Ren, and S. Yoo. Continuous field reconstruction from sparse observations with implicit neural networks. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=kuTZMZdCPZ`.

[40] B. Lusch, J. N. Kutz, and S. L. Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature communications*, 9(1):4950, 2018.

[41] M. Martin, P. Dash, A. Ignatov, V. Banzon, H. Beggs, B. Brasnett, J.-F. Cayula, J. Cummings, C. Donlon, C. Gentemann, R. Grumbine, S. Ishizaki, E. Maturi, R. W. Reynolds, and J. Roberts-Jones. Group for high resolution sea surface temperature (ghrsst) analysis fields inter-comparisons. part 1: A ghrsst multi-product ensemble (gmpe). *Deep Sea Research Part II: Topical Studies in Oceanography*, 77–80:21–30, Nov. 2012. ISSN 0967-0645. doi: 10.1016/j.dsr2.2012.04.013. URL `http://dx.doi.org/10.1016/j.dsr2.2012.04.013`.

[42] F. A. Massucci, J. Wheeler, R. Beltrán-Debón, J. Joven, M. Sales-Pardo, and R. Guimerà. Inferring propagation paths for sparsely observed perturbations on complex networks. *Science Advances*, 2(10), Oct. 2016. ISSN 2375-2548. doi: 10.1126/sciadv.1501638. URL `http://dx.doi.org/10.1126/sciadv.1501638`.

[43] J. E. Matheson and R. L. Winkler. Scoring rules for continuous probability distributions. *Management Science*, 22(10):1087–1096, June 1976. ISSN 1526-5501. doi: 10.1287/mnsc.22.10.1087. URL `http://dx.doi.org/10.1287/mnsc.22.10.1087`.

[44] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4460–4470, 2019.

[45] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[46] M. H. Parikh, X. Fan, and J.-X. Wang. Conditional flow matching for generative modeling of near-wall turbulence with quantified uncertainty. *arXiv preprint arXiv:2504.14485*, 2025.

[47] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019.

[48] J. Pathak, S. Subramanian, P. Harrington, S. Raja, A. Chattopadhyay, M. Mardani, T. Kurth, D. Hall, Z. Li, K. Azizzadenesheli, et al. Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators. *arXiv preprint arXiv:2202.11214*, 2022.

[49] M. Raissi, P. Perdikaris, and G. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, Feb. 2019. ISSN 0021-9991. doi: 10.1016/j.jcp.2018.10.045. URL `http://dx.doi.org/10.1016/j.jcp.2018.10.045`.

[50] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

[51] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, Nov. 2005. ISBN 9780262256834. doi: 10.7551/mitpress/3206.001.0001. URL `http://dx.doi.org/10.7551/mitpress/3206.001.0001`.

[52] M. Reichstein, G. Camps-Valls, B. Stevens, M. Jung, J. Denzler, N. Carvalhais, and Prabhat. Deep learning and process understanding for data-driven earth system science. *Nature*, 566 (7743):195–204, Feb. 2019. ISSN 1476-4687. doi: 10.1038/s41586-019-0912-1. URL `http://dx.doi.org/10.1038/s41586-019-0912-1`.

[53] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[54] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35: 36479–36494, 2022.

[55] J. S. Schreck, D. J. Gagne, C. Becker, W. E. Chapman, K. Elmore, D. Fan, G. Gantos, E. Kim, D. Kimpara, T. Martin, et al. Evidential deep learning: Enhancing predictive uncertainty estimation for earth system science applications. *Artificial Intelligence for the Earth Systems*, 3 (4):230093, 2024.

[56] J. Schusterbauer, M. Gui, P. Ma, N. Stracke, S. A. Baumann, V. T. Hu, and B. Ommer. Boosting latent diffusion with flow matching. In *ECCV*, 2024.

[57] D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM National Conference*, ACM '68, page 517–524, New York, NY, USA, 1968. Association for Computing Machinery. ISBN 9781450374866. doi: 10.1145/800186.810616. URL `https://doi.org/10.1145/800186.810616`.

[58] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[59] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33: 7462–7473, 2020.

[60] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33: 7462–7473, 2020.

[61] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. pmlr, 2015.

[62] Y. Song and S. Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

[63] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=PxTIG12RRHS`.

[64] M. Valdenegro-Toro and D. S. Mori. A deeper look into aleatoric and epistemic uncertainty disentanglement. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1508–1516. IEEE, 2022.

[65] F. Vasconcelos, B. He, N. M. Singh, and Y. W. Teh. UncertaINR: Uncertainty quantification of end-to-end implicit neural representations for computed tomography. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL `https://openreview.net/forum?id=jdGMBgYvfX`.

[66] X. Wang, X. Chen, B. Ni, H. Wang, Z. Tong, and Y. Liu. Deep arbitrary-scale image super-resolution via scale-equivariance pursuit. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1786–1795, 2023. doi: 10.1109/CVPR52729.2023.00178.

[67] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861.

[68] T. Weber, E. Magnani, M. Pförtner, and P. Hennig. Uncertainty quantification for fourier neural operators. In *ICLR 2024 Workshop on AI4DifferentialEquations In Science*, 2024. URL `https://openreview.net/forum?id=knSgoNJcnV`.

[69] Y. Xie, T. Takikawa, S. Saito, O. Litany, S. Yan, N. Khan, F. Tombari, J. Tompkin, V. sitzmann, and S. Sridhar. Neural fields in visual computing and beyond. *Computer Graphics Forum*, 41 (2):641–676, May 2022. ISSN 1467-8659. doi: 10.1111/cgf.14505. URL `http://dx.doi.org/10.1111/cgf.14505`.

[70] S. Xu, Y. Wang, M. Fan, B.-J. Yoon, and X. Qian. Uncertainty-aware continuous implicit neural representations for remote sensing object counting. In S. Dasgupta, S. Mandt, and Y. Li, editors, *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 4105–4113. PMLR, 02–04 May 2024. URL `https://proceedings.mlr.press/v238/xu24b.html`.

[71] M. Yadav, D. Sheldon, and C. Musco. Faster kernel interpolation for gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pages 2971–2979. PMLR, 2021.

[72] Q. Yang, A. Hernandez-Garcia, P. Harder, V. Ramesh, P. Sattigeri, D. Szwarcman, C. D. Watson, and D. Rolnick. Fourier neural operators for arbitrary resolution climate data downscaling. *Journal of Machine Learning Research*, 25(420):1–30, 2024.

[73] M. Yazdani, Y. Medghalchi, P. Ashrafian, I. Hacihaliloglu, and D. Shahriari. Flow matching for medical image synthesis: Bridging the gap between speed and quality. *arXiv preprint arXiv:2503.00266*, 2025.

[74] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[75] Z. Zhang, C. Moya, W. T. Leung, G. Lin, and H. Schaeffer. Bayesian deep operator learning for homogenized to fine-scale maps for multiscale pde. *Multiscale Modeling & Simulation*, 22(3):956–972, 2024. doi: 10.1137/23M160342X. URL `https://doi.org/10.1137/23M160342X`.

[76] E. D. Zhong, T. Bepler, B. Berger, and J. H. Davis. Cryodrgn: reconstruction of heterogeneous cryo-em structures using neural networks. *Nature Methods*, 18(2):176–185, Feb. 2021. ISSN 1548-7105. doi: 10.1038/s41592-020-01049-4. URL `http://dx.doi.org/10.1038/s41592-020-01049-4`.

[77] W. Zhong and H. Meidani. Physics-informed geometry-aware neural operator. *Computer Methods in Applied Mechanics and Engineering*, 434:117540, 2025.

A Related Works

A.1 Classical Reconstruction Methods

Physical field reconstruction resembles regression task with the objective is to predict outcomes at unobserved locations. The simplest method involves construction of a linear model from existing data. A common technique for incorporating non-linearity is inverse distance weighting (IDW) [57], which performs a weighted average using a power parameter to modulate the influence of neighboring observations. Recently, *Gaussian Process* has also become a potent alternative to classical methods [51]. Practical application of *Gaussian Processes* is often limited by the computational cost of inverting the covariance matrix, which scales with a time complexity of $\mathcal{O}(n^3)$, making it infeasible for large datasets [2, 71]. Physical field reconstruction can also be considered as a model reduction problem by approximating a continuous spatial field, $\mathbf{u}(x,t)$, as a superposition of basis functions, or modes, $\phi_i(x,t)$: $\mathbf{u}(x,t) \approx \hat{\mathbf{u}}(x,t) = \sum_{i=1}^{M} a_i \phi_i(x,t)$. This is commonly extended to reconstruct a full field from partial measurements by incorporating a mask function, defined as 1 where data is present and 0 where it is missing [16]. To address the limitations of linear methods, recent research works have employed deep learning to construct a nonlinear manifold, significantly improving performance on slowly decaying Kolmogorov $n-$width problems [33, 40, 30].

A.2 Deep Learning Reconstruction Methods

Super Resolution refers to a problem statement where the goal is to typically upscale a low-resolution image $\mathbf{u}_{lr}$ into a high-resolution image $\mathbf{u}_{hr}$. In the context of continuous field reconstruction where the objective is to reconstruct the low-resolution $\mathbf{u}_{lr}$ into a continuous domain, such paired data is scarce. Various deep learning methods have been implemented for continuous super-resolution [25, 60, 7, 20, 66], specifically based on implicit neural network. **Implicit Neural Representation**s (INRs) are coordinate-based neural networks used primarily for either parameterizing the sensor domain or the density domain [69]. Multiplicative Modulated Gabor Network (MMGN) [39] proposes a decomposition of the physical field into time(t) and space(x) dependent functions, and utilizes INR to reconstruct the space(x)-dependent function from a time(t)-dependent learned context, $z(t)$. For the out-of-distribution data reconstruction scenario where the trained model does not have a learned context, it proposes linear interpolation between time(t)-dependent learned contexts for continuous reconstruction. **Fourier Neural Operator**s (FNO) have emerged as an alternative for continuous reconstruction from sparse observations by learning the underlying partial differential equation that governs the system [35, 31]. Geometry Informed Neural Operator (GINO) [37] integrates both Graph Neural Operator (GNO) [34] and FNO for learning the continuous physical field where both the input and output data have arbitrary geometries. Several different variants of Fourier Neural Operators have been explored for reconstruction and arbitrary downscaling of climate-related physical fields [72, 77]. While *INR-based* and *FNO-based* models have shown significant promise for continuous reconstruction from sparse observations, the crucial task of quantifying the confidence or uncertainty in their predictions has largely been underexplored.

A.3 Uncertainty Quantification of INR & FNO

A few studies have focused on quantifying uncertainty in INR [65] and FNO [5, 68] models, also on uncertainty aware INR [70, 14] and FNO [75] models. *UncertaINR* [65] explores and compares various Bayesian deep learning techniques in INRs for medical image reconstruction. Bayesian Operator Learning [75] introduces a robust, mesh-free framework that provides uncertainty estimates for high-stakes problems like multiscale PDEs with noisy data. However, UQ for continuous reconstruction models from sparse observations have largely been underexplored, specially models capable of estimating both aleatoric and epistemic uncertainty with a single model. HyperDM [6] model proposes a hyper diffusion model that introduces a novel approach for estimating both uncertainty estimates from a single model for physical fields, however this approach is not inherently capable of reconstructing from sparse observations.

B Preliminaries

B.1 Uncertainty in Predictive Distribution

Uncertainty in any model's prediction primarily originates from two distinct sources of uncertainties: *Aleatoric* and *Epistemic* uncertainties.

Aleatoric Uncertainty: Aleatoric uncertainty (AU) stems from the intrinsic randomness of the measurement process itself and is formally represented by the likelihood function in Equation 1. This form of uncertainty is irreducible, meaning it cannot be decreased by acquiring more data or improving the model [29]. AU quantifies the inherent ill-posedness of a task in the context of predictive modeling. It is frequently linked to noise, measurement errors, and the fundamental unpredictability present in the observed phenomena.

Consider an inverse problem of reconstructing a physical field $\mathbf{u}$ from sparse observation $\mathbf{u}_{so}$:

$$\mathbf{u}_{so} = \mathcal{F}(\mathbf{u}) + \eta, \eta \sim \mathcal{N}(0, \sigma^2) \tag{11}$$

Even with a perfect inverse mapping $\mathcal{F}^{-1}$, there will be uncertainties in reconstructed $\mathbf{u}$ due to randomness in η. This randomness is measured by the variance σ^2 of η [27].

Epistemic Uncertainty: Epistemic uncertainty (EU) stems from a model's lack of knowledge or an incomplete understanding of a problem. Unlike aleatoric uncertainty, this type of uncertainty is reducible by training the model on more data [27]. EU essentially reflects the limitations in a model's ability to accurately capture the underlying patterns in the data. In practice, if we randomly initialize M different model weights $\{\theta_i\}_{i=1}^{M}$, and train them with training data in hand, discrepancies will inevitably arise in the model weights (*Epistemic Uncertainty*) due to random initialization of weights. Further addition of training data will reduce this discrepancies in model weights, effectively will reduce EU [27, 12].

B.2 Flow Matching Model

Flow Matching offers an efficient and robust alternative to traditional diffusion models for learning continuous transformations between probability distributions [38]. Flow Matching aims to directly learn a velocity field that transports samples from a simple base distribution, e.g. Gaussian noise, $x_0 \sim p(x_0) = \mathcal{N}(0, 1)$ to a complex target data distribution, $x_1 \sim p(x_1)$.

Flow Matching defines a family of probability paths $p_t(x)$ that smoothly interpolate between a source distribution $p_0(x)$ at time $t = 0$ and the target data distribution $p_1(x)$ at time $t = 1$. These paths are deterministic and governed by an ordinary differential equation: $\frac{d}{dt}(x_t) = v_t(x_t)$, where $v_t(x_t)$ is the velocity field at time t and position x_t. In this work, we design this velocity field in a context-based manner where the velocity field is conditioned on a context c, and the differential equation becomes $\frac{d}{dt}(x_t) = v_t(x_t, c)$. The objective in Flow Matching model is to learn this velocity field by approximating a neural network, v_θ. The training objective for the neural network v_θ is to minimize the loss:

$$\mathcal{L}(\theta) = \mathbb{E}_{x_0 \sim \mathcal{N}(0,1), x_1 \sim p(x_1), t \sim \mathcal{U}(0,1)} \left[\| v_\theta((1-t)x_0 + t x_1, c, t) - (x_1 - x_0) \|_2^2 \right] \tag{12}$$

C Experimental Setup

C.1 Dataset: Simulation-based Global Surface Temperature

The dataset is derived from the pre-industrial control run of the Community Earth System Model version 2 (CESM2), a cutting-edge climate model from the National Center for Atmospheric Research (NCAR). This dataset, consisting of monthly averaged global surface temperature data, represents a complex atmospheric field characterized by a wide range of spatial and temporal variability, from turbulence to large-scale climatic patterns. Its intricate nature provides a rigorous testbed for our methodology.

The CESM2 model, developed by NCAR, simulates Earth system interactions. Its pre-industrial control run provides monthly averaged global surface temperature data, serving as a baseline of stable climate absent anthropogenic influences. This dataset, encompassing complex atmospheric variability, offers a robust platform for testing our proposed FlowGINO model. These data are accessible via the WCRP's CMIP initiative, with coordinating support from PCMDI at pcmdi.llnl.gov.

C.2 Data Processing

The total number of samples in the dataset is 1212. We sampled 102 datapoints from the overall dataset. The datapoints sampled for the training set are evenly spaced in time. For example the time indices for datapoints in the training set are $\{0, 11, 22, \cdots, 1212\}$. For the primary experiment, we downsampled each original datapoint from dimension (192×288) to (97×145) and (25×37) to create a pair of high-resolution and low-resolution sample to represent the continuous physical field and the sparse observations accrodingly.

C.3 Evaluation Metrics

Peak Signal to Noise Ratio(PSNR): PSNR evaluates the quality of the reconstructed signal by measuring the ratio between the peak signal power and noise power present in the signal. For a reconstructed output $\hat{u}$, and its corresponding gournd-truth u, the peak signal to noise ratio is calculated using the following formula:

$$\textbf{PSNR}(u, \hat{u}) = 10 \log_{10} \left(\frac{(\text{MAX}(u) - \text{MIN}(u))^2}{\frac{1}{hw} \sum_{x^c} (u(x^{(c)}) - \hat{u}(x^{(c)}))^2} \right) \tag{13}$$

Here, $h \times w$ is the dimension of the reconstructed signal.

Structural Similarity Index Measure(SSIM): SSIM assesses the perceived quality degradation caused by processing, such as data compression or loss in digital transmission. It is a well-established metric for evaluating the quality of images or uniform 2D grid-like data, particularly in tasks like super-resolution, by comparing their structural resemblance. For a super resolved predicted output $\hat{u}$, and its corresponding gournd-truth u, the structural similarity index measure between the window w_x of u and window $w_{\hat{x}}$ is calculated using the following formula:

$$\textbf{SSIM}(u_{w_x}, \hat{u}_{w_{\hat{x}}}) = \frac{(2\mu_x \mu_{\hat{x}} + c_1)(2\sigma_{x\hat{x}} + c_2)}{(\mu_x^2 + \mu_{\hat{x}}^2 + c_1)(\sigma_x^2 + \sigma_{\hat{x}}^2 + c_2)} \tag{14}$$

Here,

- $\mu_x =$Sample mean of window w_x
- $\mu_{\hat{x}} =$Sample mean of window $w_{\hat{x}}$
- $\sigma_x^2 =$Variance of window w_x
- $\sigma_{\hat{x}^2} =$Variance of window $w_{\hat{x}}$
- $\sigma_{x\hat{x}} =$Covariance of windows w_x and $w_{\hat{x}}$
- $c_1 = (k_1 L)^2, c_2 = (k_2 L)^2$
- $k_1 = 0.01, k_2 = 0.03$
- L is the range of values for u and $\hat{x}$, $L = 2$ is chosen as the range is $[-1, 1]$

For evaluating the metric for the overall frame of the reconstructed field and ground truth, **SSIM** over all windows are averaged.

$$\textbf{SSIM}(u, \hat{u}) = \frac{1}{|\{w_x\}||\{w_{\hat{x}}\}|} \sum_{w_x, w_{\hat{x}}} \textbf{SSIM}(u_{w_x}, \hat{u}_{w_{\hat{x}}}) \tag{15}$$

Learned Perceptual Image Patch Similarity(LPIPS): LPIPS is a perceptual similarity metric that utilizes feature representations extracted from a pre-trained deep convolutional neural network (CNN), such as AlexNet [32], VGG [58], SqueezeNet [24] or a fine-tuned version of them. In this work, we utilize the VGG [58] model for evaluating this metric.

LPIPS compares the feature activations of two images: a reference image, $\mathbf{u}$, and a test image, $\hat{\mathbf{u}}$, rather than their raw pixel values. The mathematical formulation for the LPIPS metric between two images $\mathbf{u}$ and $\hat{\mathbf{u}}$ is given by:

$$\textbf{LPIPS}(\mathbf{u}, \hat{\mathbf{u}}) = \sum_{i=1}^{L} w_i \|\phi_i(\mathbf{u}) - \phi_i(\hat{\mathbf{u}})\|_2^2 \tag{16}$$

Here, ϕ_i's are the pretrained model weights and w_i's are layer specific weights.

Continuous Ranked Probability Score(CRPS): This metric computes a holistic indicator of the quality of reconstructed field $\hat{\mathbf{u}}$. For the location x in space, and the reconstructed field and ground truth at the location, $\mathbf{u}_x$ and $\hat{\mathbf{u}}_x$ accordingly, the metric is evaluated by following location:

$$\textbf{CRPS}(\hat{\mathbf{u}}_x) = \int_{-\infty}^{\infty} [F(\hat{\mathbf{u}}_x) - 1_{\hat{\mathbf{u}}_x \geq \mathbf{u}_x}]^2 d\mathbf{u}_x, \tag{17}$$

where F is the cumulative distribution function and 1 stands for the Heaviside step function. To evaluate this metric for the overall reconstructed field, it is averaged over all the locations, x.

$$\textbf{CRPS}(\hat{\mathbf{u}}) = \frac{1}{|\{x\}|} \sum_{x} \textbf{CRPS}(\hat{\mathbf{u}}_x)) \tag{18}$$

D Additional Results

We first present some qualitative results and then present some additional experimental results conducted in different experimental setups.

D.1 Qualitative Results

Figure 2 presents some visual results of predictions from different models. Each rows show predictions for data at timepoints $\{0, 1, 3, 5, 6, 7, 9, 11\}$. Among these timepoints, only data from timepoints $\{0, 11\}$ were present in the training dataset. We observe that baseline MMGN is capable of reconstructing high frequency components on data points that existed in training set. However, its prediction on test dataset is visually very different from the ground truth. This illustrates that even though MMGN [39] is capable of capturing high frequency components, its reconstruction capacity largely depends on the quality of learned latents. The MMGN [39] model learns time-specific latents $z(t)$ for those timepoints t that are present in training dataset, and struggle on test timepoints. FlowGINO model is capable of both accurate reconstruction and high frequency component capturing.

D.2 Uncertainy Estimation with FlowGINO

Figure 3 and 4 shows the prediction and corresponding epistemic and aleatoric uncertainty estimates from the MCD and VBLL adaptation of FlowGINO accordingly. Each rows show predictions for data at timepoints $\{0, 1, 3, 5, 6, 7, 9, 11\}$. Among these timepoints, only data from timepoints $\{0, 11\}$ were present in the training dataset.

For the MCD adaptation of the FlowGINO model, both aleatoric and epistemic uncertainty look visually quite similar for all timepoints. Also there is not that much visually differentiable differences between uncertainty estimates between training and test timepoints (first and last rows correspond to data at timepoints present in training set).

For the VBLL adaptation of the FlowGINO model, there is stark visual differences between aleatoric and epistemic uncertainties across all the timepoints. Also the uncertainty estimates are much lower for the train timepoints compared to test timepoints (first and last rows correspond to data at timepoints present in training set).

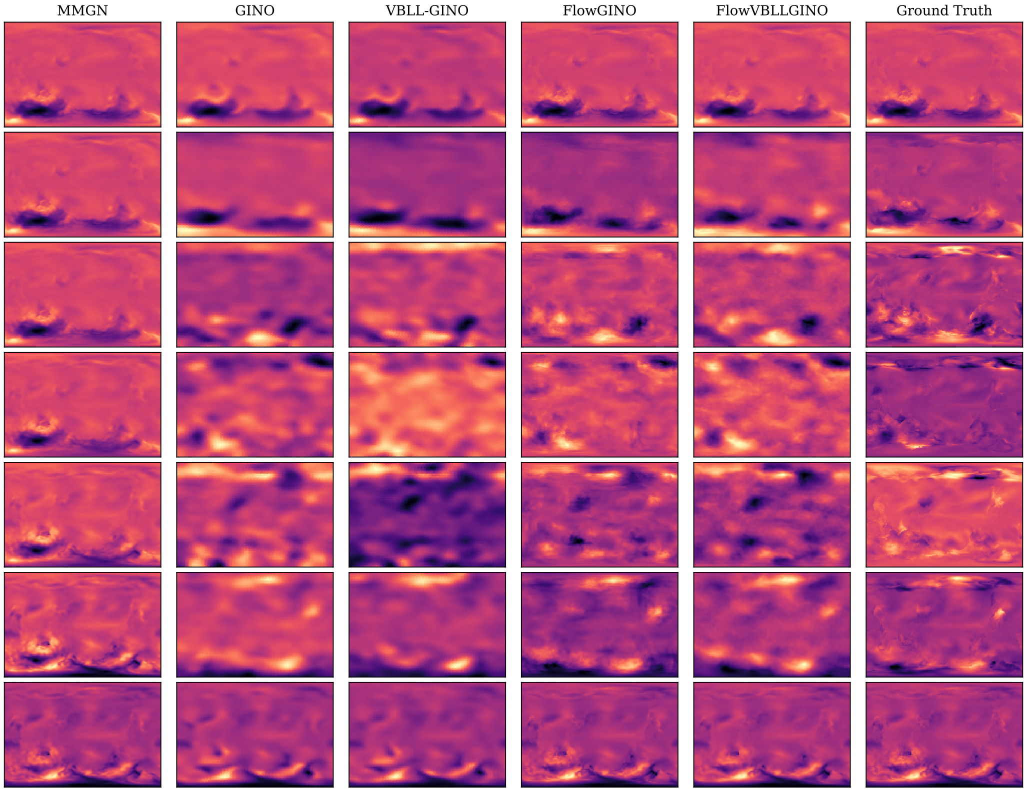

Figure 2: Predictions from different results.

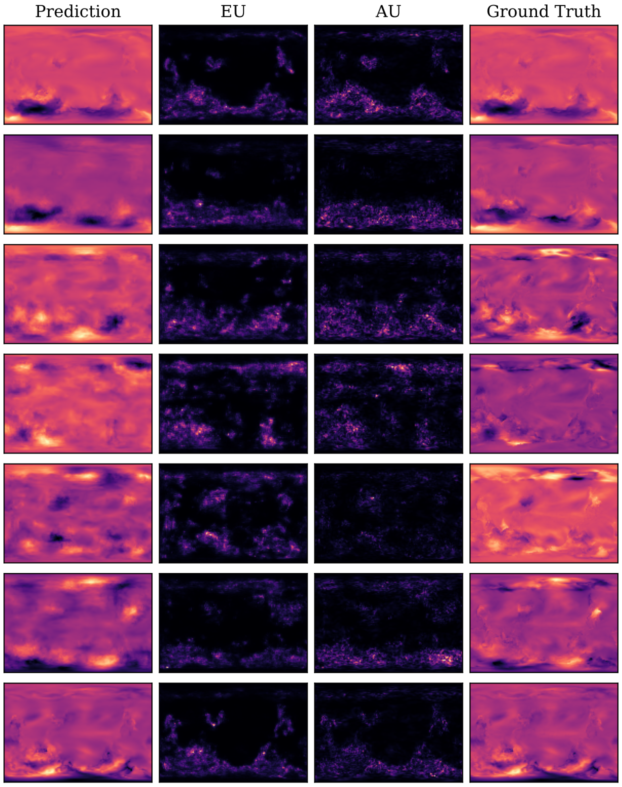

Figure 3: Epistemic & Aleatoric uncertainty estimation with MCD adaptation of FlowGINO.

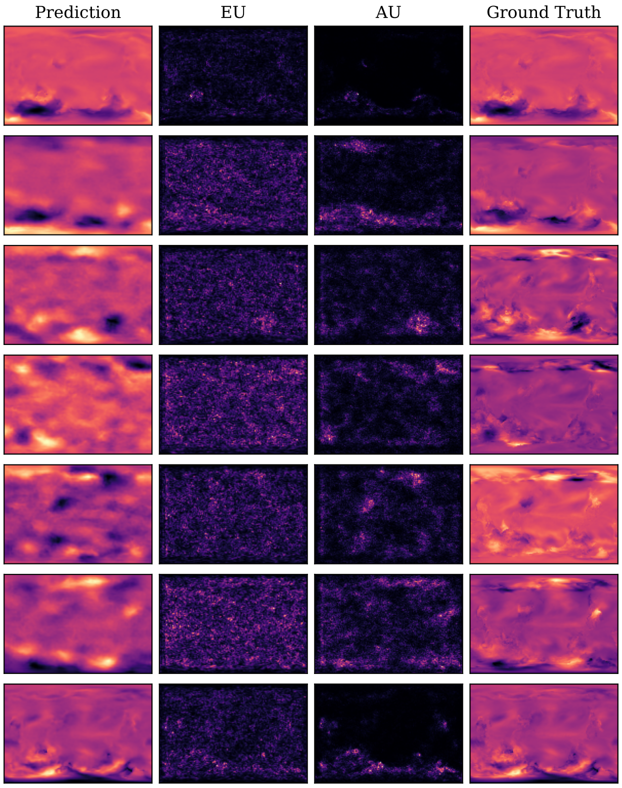

Figure 4: Epistemic & Aleatoric uncertainty estimation with VBLL adaptation of FlowGINO.

E Additional Tasks

We further conduct experiments in three several evaluation scenarios to evaluate the reconstruction capacity of our model: (i) *Reconstruction from Extremely Sparse Observations*, (ii) *Reconstruction from Moderately Sparse Observations*, and (iii) *Reconstruction from Mildly Sparse Observations*.

Reconstruction from Extremely Sparse Observations: We downsample the high resolution data with dimension (192×288) to (8×8) data and then evaluate the reconstruction capabilities of our model. We trained the models with four different scenarios with different number of data samples at different timepoints: $16, 32, 64, 128$. Table 2 shows the evaluation results for this reconstruction scenario. MMGN model performs best in the training set, but GINO-VBLL and FlowGINO perform better overall for the test set. In terms of the LPIPS metric, MMGN model obtains the lowest value for all cases. The MCD adaptation of the FlowGINO model obtains lower CRPS value across all scenarios.

Reconstruction from Moderately Sparse Observations: We downsample the high resolution data with dimension (192×288) to (16×16) data and then evaluate the reconstruction capabilities of our model. We trained the models with four different scenarios with different number of data samples at different timepoints: $16, 32, 64, 128$. Table 3 shows the evaluation results for this reconstruction scenario. MMGN model performs best in the training set, but FlowGINO performs better overall for the test set. In terms of the LPIPS metric, MMGN model obtains the lowest value for all cases. The MCD adaptation of the FlowGINO model obtains lower CRPS value across all scenarios.

Reconstruction from Mildly Sparse Observations: We downsample the high resolution data with dimension (192×288) to (32×32) data and then evaluate the reconstruction capabilities of our model. We trained the models with four different scenarios with different number of data samples at different timepoints: $16, 32, 64, 128$. Table 4 shows the evaluation results for this reconstruction scenario. MMGN model performs best in the training set, except the case for $T_{train} = 128$. But

Table 2: Evaluation results for extreme sparse reconstruction task.

$\mathbf{u}_{(8,8)} \rightarrow \mathbf{u}_{(192,288)}, T_{train} = 16$								
Model	Train Set				Test Set			
	PSNR (↑)	SSIM (↑)	LPIPS (↓)	CRPS (↓)	PSNR (↑)	SSIM (↑)	LPIPS (↓)	CRPS (↓)
MMGN	**46.6948**	**0.9546**	**0.0690**	-	28.0145	0.3311	**0.2288**	-
GINO	42.1094	0.9239	0.1796	-	30.0127	0.3814	0.3291	-
GINO-VBLL	43.1372	0.9436	0.1475	-	**30.0630**	**0.3857**	0.3415	-
FlowGINO(MCD)	37.7819	0.8467	0.2854	**0.0157**	29.1762	0.3507	0.3799	**0.0455**
FlowGINO(VBLL)	29.6849	0.2527	0.4718	0.0448	29.2836	0.2202	0.4706	0.0461
$\mathbf{u}_{(8,8)} \rightarrow \mathbf{u}_{(192,288)}, T_{train} = 32$								
Model	Train Set				Test Set			
	PSNR (↑)	SSIM (↑)	LPIPS (↓)	CRPS (↓)	PSNR (↑)	SSIM (↑)	LPIPS (↓)	CRPS (↓)
MMGN	**44.9331**	**0.9624**	**0.0717**	-	27.5597	0.3344	**0.2246**	-
GINO	43.3198	0.9531	0.1540	-	30.6466	0.4339	0.3348	-
GINO-VBLL	43.8460	0.9590	0.1297	-	30.6335	0.4262	0.3314	-
FlowGINO(MCD)	43.3923	0.9474	0.2463	**0.0074**	**30.7992**	**0.4594**	0.3777	**0.0367**
FlowGINO(VBLL)	40.6105	0.9059	0.3091	0.0106	30.3391	0.3987	0.3899	0.0400
$\mathbf{u}_{(8,8)} \rightarrow \mathbf{u}_{(192,288)}, T_{train} = 64$								
Model	Train Set				Test Set			
	PSNR (↑)	SSIM (↑)	LPIPS (↓)	CRPS (↓)	PSNR (↑)	SSIM (↑)	LPIPS (↓)	CRPS (↓)
MMGN	**47.4978**	**0.9728**	**0.0943**	-	28.2149	0.3846	**0.2388**	-
GINO	43.2889	0.9533	0.1380	-	30.9415	0.4629	0.3255	-
GINO-VBLL	44.1765	0.9615	0.1115	-	**31.0180**	0.4662	0.3249	-
FlowGINO(MCD)	45.8433	0.9717	0.1853	**0.0052**	30.9085	**0.4673**	0.3325	**0.0353**
FlowGINO(VBLL)	44.8832	0.9479	0.1724	0.0066	30.9385	0.4525	0.3536	0.0363
$\mathbf{u}_{(8,8)} \rightarrow \mathbf{u}_{(192,288)}, T_{train} = 128$								
Model	Train Set				Test Set			
	PSNR (↑)	SSIM (↑)	LPIPS (↓)	CRPS (↓)	PSNR (↑)	SSIM (↑)	LPIPS (↓)	CRPS (↓)
MMGN	**46.9616**	**0.9710**	0.1144	-	28.4653	0.4310	**0.2481**	-
GINO	43.8930	0.9573	0.1194	-	**31.3444**	**0.4977**	0.3244	-
GINO-VBLL	44.1813	0.9600	**0.1140**	-	31.2490	0.4858	0.3174	-
FlowGINO(MCD)	45.8077	0.9660	0.1594	**0.0056**	31.1081	0.4730	0.3285	**0.0347**
FlowGINO(VBLL)	44.3541	0.9568	0.1325	0.0068	30.9303	0.4556	0.3174	0.0356

Table 3: Evaluation results for moderately sparse reconstruction task.

$\mathbf{u}_{(16,16)} \rightarrow \mathbf{u}_{(192,288)}, T_{train} = 16$								
Model	Train Set				Test Set			
	PSNR (↑)	SSIM (↑)	LPIPS (↓)	CRPS (↓)	PSNR (↑)	SSIM (↑)	LPIPS (↓)	CRPS (↓)
MMGN	**46.6972**	**0.9546**	**0.0690**	-	28.0143	0.3311	**0.2288**	-
GINO	42.9691	0.9468	0.1710	-	33.2471	0.5975	0.3166	-
GINO-VBLL	43.3425	0.9469	0.1291	-	33.6210	0.6110	0.3166	-
FlowGINO(MCD)	42.7737	0.9442	0.2423	**0.0081**	**33.8617**	**0.6338**	0.3456	**0.0262**
FlowGINO(VBLL)	23.7195	-0.0224	0.3547	0.0824	23.6741	0.0165	0.3782	0.0821
$\mathbf{u}_{(16,16)} \rightarrow \mathbf{u}_{(192,288)}, T_{train} = 32$								
Model	Train Set				Test Set			
	PSNR (↑)	SSIM (↑)	LPIPS (↓)	CRPS (↓)	PSNR (↑)	SSIM (↑)	LPIPS (↓)	CRPS (↓)
MMGN	**49.3047**	**0.9825**	**0.0696**	-	27.9254	0.3459	**0.2264**	-
GINO	43.3134	0.9545	0.1575	-	33.6848	0.6222	0.3194	-
GINO-VBLL	43.8417	0.9575	0.1362	-	33.8904	0.6333	0.3172	-
FlowGINO(MCD)	42.5389	0.8956	0.2311	**0.0097**	**34.0405**	**0.6374**	0.3515	**0.0259**
FlowGINO(VBLL)	41.9163	0.8559	0.2024	0.0113	34.0218	0.6306	0.3521	0.0260
$\mathbf{u}_{(16,16)} \rightarrow \mathbf{u}_{(192,288)}, T_{train} = 64$								
Model	Train Set				Test Set			
	PSNR (↑)	SSIM (↑)	LPIPS (↓)	CRPS (↓)	PSNR (↑)	SSIM (↑)	LPIPS (↓)	CRPS (↓)
MMGN	**46.4090**	**0.9663**	**0.0965**	-	28.1193	0.3839	**0.2397**	-
GINO	44.2540	0.9625	0.1164	-	34.0957	0.6566	0.3018	-
GINO-VBLL	44.4059	0.9642	0.1104	-	**34.1569**	0.6559	0.3103	-
FlowGINO(MCD)	42.9999	0.9518	0.1917	**0.0075**	33.9583	**0.6626**	0.3159	**0.0250**
FlowGINO(VBLL)	30.3100	0.3822	0.4381	0.0409	29.9385	0.3511	0.4412	0.0417
$\mathbf{u}_{(16,16)} \rightarrow \mathbf{u}_{(192,288)}, T_{train} = 128$								
Model	Train Set				Test Set			
	PSNR (↑)	SSIM (↑)	LPIPS (↓)	CRPS (↓)	PSNR (↑)	SSIM (↑)	LPIPS (↓)	CRPS (↓)
MMGN	**47.1473**	0.9718	0.1131	-	28.4214	0.4296	**0.2468**	-
GINO	44.7533	0.9636	**0.1036**	-	34.6674	0.6922	0.2944	-
GINO-VBLL	44.6601	0.9644	**0.1036**	-	34.5325	0.6740	0.3036	-
FlowGINO(MCD)	46.5103	**0.9774**	0.1497	**0.0049**	**34.7715**	**0.7003**	0.3075	**0.0224**
FlowGINO(VBLL)	46.4927	0.9587	0.1175	0.0055	34.6321	0.6708	0.3179	0.0231

FlowGINO performs better overall for the test set. In terms of the LPIPS metric, MMGN model obtains the lowest value for all cases. The MCD adaptation of the FlowGINO model obtains lower CRPS value across all scenarios.

Table 4: Evaluation results for mildly sparse reconstruction task.

$$\mathbf{u}_{(32,32)} \rightarrow \mathbf{u}_{(192,288)}, T_{train} = 16$$

Model	Train Set				Test Set			
	PSNR (↑)	SSIM (↑)	LPIPS (↓)	CRPS (↓)	PSNR (↑)	SSIM (↑)	LPIPS (↓)	CRPS (↓)
MMGN	**46.6935**	**0.9545**	**0.0690**	-	28.0147	0.3311	**0.2289**	-
GINO	41.4539	0.9242	0.2242	-	34.5181	0.7042	0.3255	-
GINO-VBLL	43.9765	0.9585	0.1378	-	35.9267	0.7494	0.3187	-
FlowGINO(MCD)	42.6504	0.9440	0.2399	**0.0081**	**36.1245**	**0.7741**	0.3468	**0.0199**
FlowGINO(VBLL)	22.4499	-0.0306	0.3846	0.0937	22.4847	0.0006	0.4050	0.0930

$$\mathbf{u}_{(32,32)} \rightarrow \mathbf{u}_{(192,288)}, T_{train} = 32$$

Model	Train Set				Test Set			
	PSNR (↑)	SSIM (↑)	LPIPS (↓)	CRPS (↓)	PSNR (↑)	SSIM (↑)	LPIPS (↓)	CRPS (↓)
MMGN	**48.2583**	**0.9739**	**0.0742**	-	28.0351	0.3477	**0.2303**	-
GINO	43.7617	0.9583	0.1414	-	35.8233	0.7506	0.3109	-
GINO-VBLL	43.9584	0.9583	0.1299	-	36.2527	0.7587	0.3156	-
FlowGINO(MCD)	44.7964	0.9621	0.1470	**0.0065**	36.5041	**0.7829**	0.2741	**0.0181**
FlowGINO(VBLL)	41.6988	0.9178	0.2922	0.0094	**36.5584**	0.7698	0.3609	0.0185

$$\mathbf{u}_{(32,32)} \rightarrow \mathbf{u}_{(192,288)}, T_{train} = 64$$

Model	Train Set				Test Set			
	PSNR (↑)	SSIM (↑)	LPIPS (↓)	CRPS (↓)	PSNR (↑)	SSIM (↑)	LPIPS (↓)	CRPS (↓)
MMGN	**47.6655**	**0.9745**	**0.0954**	-	28.2297	0.3858	**0.2400**	-
GINO	43.6803	0.9530	0.1315	-	36.0042	0.7593	0.2998	-
GINO-VBLL	44.6384	0.9652	0.1086	-	36.5408	0.7749	0.3045	-
FlowGINO(MCD)	44.1955	0.9572	0.2295	**0.0066**	**36.9981**	**0.8040**	0.3332	**0.0176**
FlowGINO(VBLL)	44.2772	0.9586	0.1449	0.0069	36.1194	0.7546	0.3149	0.0197

$$\mathbf{u}_{(32,32)} \rightarrow \mathbf{u}_{(192,288)}, T_{train} = 128$$

Model	Train Set				Test Set			
	PSNR (↑)	SSIM (↑)	LPIPS (↓)	CRPS (↓)	PSNR (↑)	SSIM (↑)	LPIPS (↓)	CRPS (↓)
MMGN	46.6480	0.9676	0.1126	-	28.4337	0.4294	**0.2459**	-
GINO	44.7390	0.9658	0.1060	-	36.5785	0.7883	0.2931	-
GINO-VBLL	44.7899	0.9652	**0.1026**	-	36.8930	0.7923	0.2976	-
FlowGINO(MCD)	**46.9878**	**0.9797**	0.1362	**0.0046**	**37.5084**	**0.8244**	0.2745	**0.0160**
FlowGINO(VBLL)	45.8479	0.9502	0.1199	0.0061	36.8102	0.7712	0.3144	0.0180

Result Summary: MMGN model performs best on the training dataset across all evaluation scenarios. However, when data samples in training size is considerably higher and the input observations are not extremely sparse. Our proposed FlowGINO model performs best in almost all test set scenarios. However, GINO and GINO-VBLL showed better performances in some of the extreme sparse observation scenarios where the spatial observations were extremely sparse and the number of data samples in the training dataset was also very low. However, MMGN model obtained low LPIPS value in almost all evaluation scenarios, even when performances over the other two metrics were very poor. This raises the concern over applicability of LPIPS metric in *NOT Real-Life Image* like data reconstruction tasks.