# Meta-INR: Efficient Encoding of Volumetric Data via Meta-Learning Implicit Neural Representation

Maizhe Yang* Kaiyuan Tang† Chaoli Wang‡

University of Notre Dame

**ABSTRACT**

Implicit neural representation (INR) has emerged as a promising solution for encoding volumetric data, offering continuous representations and seamless compatibility with the volume rendering pipeline. However, optimizing an INR network from randomly initialized parameters for each new volume is computationally inefficient, especially for large-scale time-varying or ensemble volumetric datasets where volumes share similar structural patterns but require independent training. To close this gap, we propose Meta-INR, a pretraining strategy adapted from meta-learning algorithms to learn initial INR parameters from partial observation of a volumetric dataset. Compared to training an INR from scratch, the learned initial parameters provide a strong prior that enhances INR generalizability, allowing significantly faster convergence with just a few gradient updates when adapting to a new volume and better interpretability when analyzing the parameters of the adapted INRs. We demonstrate that Meta-INR can effectively extract high-quality generalizable features that help encode unseen similar volume data across diverse datasets. Furthermore, we highlight its utility in tasks such as simulation parameter analysis and representative timestep selection. The code is available at https://github.com/spacefarers/MetaINR.

## 1 INTRODUCTION

Volume data representation is a long-standing theme for scientific computing and visualization. Recently, researchers have explored using *implicit neural representation* (INR) [17] for volume data representation [10, 5, 21, 20]. Using an architecture of *multilayer perception* (MLP), INR maps spatial coordinates to corresponding voxel values for volume fitting and stores the parameters of the MLP to represent the underlying volume. In this context, INR offers three advantages. First, the model size of a fully connected network is usually much smaller than the original volume, implying that INR can achieve highly compressive results. Second, INR can be naturally embedded into the volume rendering pipeline. A renderer can directly access the voxel values along a ray by inferring the trained network, eliminating the need to decode the entire volume beforehand. Third, once trained, an INR can achieve arbitrary-scale interpolation without accessing the original data, significantly enhancing the convenience of post-hoc analysis.

Despite its effectiveness, the current INR network parameters optimized on one volume are not generalizable enough to be adapted to other unseen volumes. When users represent new simulation volume data using INR, the parameters of the trained INR cannot be directly reused to accelerate training on other volumes, which leads to inefficient encoding for time-varying or ensemble datasets. Inspired by *meta-learning* in neural networks [7], we propose Meta-INR, a pretraining strategy for INR that only uses partial observation of a volumetric dataset but enables rapid adaptation to unseen volumes with similar structures in a few gradient steps.

The contributions of our work are as follows. First, we are the first to apply meta-learning techniques to volume data representation. We demonstrate that Meta-INR, pretrained using no more than 1% of data samples, can adapt to unseen similar volumes efficiently. Our method reduces the computational overhead of encoding new volumes by leveraging meta-learned priors, leading to significantly faster encoding than training from scratch. Second, we show that the parameters of adapted INRs exhibit enhanced interpretability, making them useful for simulation parameter analysis and representative timestep selection tasks. Finally, we evaluate Meta-INR on various datasets by comparing its performance with other training strategies.

## 2 RELATED WORK

**INR for scientific visualization.** Using deep learning techniques for data representation [24] has been extensively studied recently. One solution uses INR, which inputs spatial coordinates and outputs corresponding voxel values to achieve the generation and reduction of scientific data. Typically, INR leverages the MLP architecture to represent volumetric data. For example, Lu et al. [10] compressed a single scalar field by optimizing an INR with weight quantization. Han and Wang [5] proposed CoordNet, a coordinate-based network to tackle diverse data and visualization generation tasks. Tang and Wang [21] presented STSR-INR, leveraging an INR to generate simultaneous spatiotemporal super-resolution for time-varying multivariate volumetric data. Han et al. [6] proposed KD-INR to handle large-scale time-varying volumetric data compression when volumes are only sequentially accessible during training. Tang and Wang [20] designed ECNR to achieve efficient time-varying data compression by combining INR with the Laplacian pyramid for multiscale fitting. Li and Shen [8] leveraged Gaussian distribution to model the probability distribution of an INR network to achieve efficient isosurface extraction.

Besides the vanilla MLP architecture, multiple works integrate grid parameters into INR to achieve efficient encoding and rendering. Weiss et al. [25] implemented fV-SRN, achieving significant rendering speed gain over [10] using a volumetric latent grid. Wurster et al. [27] proposed APMGSRN, which uses multiple spatially adaptive feature grids to represent a large volume. Xiong et al. [28] designed MDSRN to simultaneously reconstruct the data and assess the reconstruction quality in one INR network. Tang and Wang [22] developed StyleRF-VolVis, leveraging a grid-based encoding INR to represent a volume rendering scene that supports various editings. Yao et al. [29] proposed ViSNeRF, utilizing a multidimensional INR representation for visualization synthesis of dynamic scenes, including changes of transfer functions, isovalues, timesteps, or simulation parameters. Gu et al. [4] and Lu et al. [9] presented NeRVI and FCNR, respectively, utilizing INRs for the effective compression of a large collection of visualization images. Unlike existing works that mainly focus on network architecture design, this paper aims to develop a pretraining strategy for optimizing the initial parameters of an INR network to enhance the representation generalizability.

*e-mail: myang9@nd.edu

†e-mail: ktang2@nd.edu

‡e-mail: chaoli.wang@nd.edu

**Meta-learning.** Meta-learning is a deep learning technique primarily aiming for few-shot learning [18]. Numerous recent studies have explored using it to optimize the initialization of neural networks, enabling them to adapt to new tasks with just a few steps of gradient descent. For instance, Sitzmann et al. [16] leveraged meta-learners to generalize INR across shapes. Tancik et al. [19] employed meta-learning to initialize INR network parameters according to the underlying class of represented signals. Emilien et al. [2] proposed COIN++, a neural compression framework that supports encoding various data modalities with a meta-learned base network. Similar to these works, we develop Meta-INR based on existing meta-learning algorithms [3, 11]. However, we focus on applying meta-learning in INR for volume data representation, setting it apart from existing studies.

## 3 META-INR

Meta-INR is a pretraining approach designed to optimize the initial parameters of an INR network for efficient finetuning on unseen volumes. The training pipeline of Meta-INR consists of two sequential stages: *meta-pretraining* and *volume-specific finetuning*.

- The *meta-pretraining* stage optimizes a meta-model on a sparse subsampled volumetric dataset, utilizing less than 1% of the original data, to learn the initial parameters of an INR network that supports rapid adaptation to other unseen volumes within the dataset.
- The *volume-specific finetuning* stage finetunes the initial parameters of the meta-model on a specific volume, resulting in a volume-specific adapted INR for high-fidelity volume reconstruction.

The adapted INR shares the same network architecture, $\Phi : \mathbb{R}^3 \rightarrow \mathbb{R}$ as the meta-model, which maps a 3D coordinate to the corresponding voxel value. Let $\theta_m$ denote the parameters of the meta-model. For a volumetric dataset $\mathbf{D} = \{d_1, d_2, \ldots, d_i, \ldots, d_T\}$ that contains $T$ timesteps or ensembles, one volume $d_i$ consists of a set of coordinate-value pairs $(\mathbf{C}_i, \mathbf{V}_i)$, where $\mathbf{C} = \{(x_1, y_1, z_1), (x_2, y_1, z_1), \ldots\}$ is a set of spatial coordinates and $\mathbf{V} = \{v_1, v_2, \ldots\}$ is the corresponding voxel values at these positions. After meta-pretraining, we finetune $\theta_m$ on each set of coordinate-value pairs in $\mathbf{D}$ independently, resulting in a series of volume-specific adapted INRs with parameters $\{\theta_1, \theta_2, \ldots, \theta_T\}$ that can represent each volume within the temporal or ensemble sequence.

### 3.1 Meta-Pretraining

The meta-pretraining stage optimizes the parameters of $\theta_m$, serving as the initial parameters in the volume-specific finetuning stage. Meta-pretraining is data efficient, requiring only a small portion of the data to derive sufficiently generalizable $\theta_m$. In this paper, we achieve this by spatiotemporally subsampling the original dataset $\mathbf{D}$. Specifically, we subsample $\mathbf{D}$ using an interval of $\lambda_s$ on each spatial dimension and $\lambda_t$ on the temporal dimension, resulting in a downsampled dataset $\hat{\mathbf{D}}$.

The spatial subsampling interval $\lambda_s = 4$ and temporal subsampling interval $\lambda_t = 2$ are chosen empirically, balancing pretraining efficiency and quality of the prior. This configuration retains structural patterns for most datasets without significant accuracy loss. It further implies that only $\frac{1}{(4^3 \times 2)} \times 100\% \approx 0.78\%$ original data samples are used for pretraining.

Then, we consider optimizing $\theta_m$ as a meta-learning problem and propose to leverage a MAML-like algorithm [3]. In particular, we randomly initialize the meta-model parameters and iteratively update $\theta_m$ through a nested inner and outer loop. In the inner loop, for each subsampled volumetric dataset $\hat{d}_i$ in $\hat{\mathbf{D}}$, we finetune a cloned set of parameters $\theta'$ using $K$ gradient steps on randomly sampled batches of coordinate-value pairs $(\mathbf{C}_i, \mathbf{V}_i)$. The inner-loop loss is computed as the mean squared error (MSE) between the predicted and ground-truth (GT) voxel values. After processing all volumes in $\hat{\mathbf{D}}$, the outer loop accumulates the gradients from the inner loop and updates $\theta_m$ using the average gradient across the volumes. The optimization continues until $\theta_m$ converges.

**Algorithm 1:** Meta-Pretraining

**Input:** Volumetric dataset $\mathbf{D}$ with $T$ timesteps or ensembles, spatial and temporal downsampling intervals $\lambda_s$ and $\lambda_t$, inner and outer loop learning rates $\alpha$ and $\beta$, number of inner-loop steps $K$.
**Output:** Optimized meta-model parameters $\theta_m$.

```
Subsample D under λ_s and λ_t to obtain downsampled dataset D̂ = {d̂_1, ..., d̂_T'} with T' timesteps or ensembles
Randomly initialize meta-model parameters θ_m
while not done do
    Initialize gradients: ∇θ_m = 0;
    for all d̂_i do
        Clone parameters: θ' ← θ_m;
        Randomly sample batches of coordinate-value pairs (C_i, V_i) from d̂_i;
        for k = 1 to K do
            Compute loss: L ← MSE(Φ(C_i; θ'), V_i);
            θ' ← θ' − α∇_θ' L;
        end
        ∇θ_m ← ∇θ_m + (θ − θ');
    end
    Update meta-model parameters: θ_m ← θ_m − β ∇θ_m / T';
end
return θ_m
```

### 3.2 Volume-Specific Finetuning

Once the meta-pretraining stage finishes, we utilize $\theta_m$ as initial parameters to finetune each adapted INR on a specific volume in $\mathbf{D}$. The finetuning process mirrors the inner-loop adaptation during meta-pretraining, where the pre-trained initial parameters $\theta_m$ are taken and updated in $K$ gradient steps. The only difference is that we utilize all available data points in this stage instead of a subsampled version of the volumetric dataset. This ensures that the finetuned parameters follow a more accurate gradient descent direction from $\theta_m$, leading to improved reconstruction accuracy. After the volume-specific finetuning stage, we can obtain a series of volume-wise adapted INRs, each representing a specific volume within the target time-varying or ensemble sequence.

Table 1: Experimented datasets and their respective settings.

| dataset | dimension (x×y×z) | # of timesteps or ensembles |
|---|---|---|
| earthquake | 256×256×96 | 598 |
| half-cylinder [14] | 640×240×80 | 20 |
| ionization [26] | 600×248×248 | 30 |
| Tangaroa [12] | 300×180×120 | 30 |
| vortex [15] | 128×128×128 | 15 |
| Nyx [1] | 256×256×256 | 209 |

## 4 RESULTS AND DISCUSSION

### 4.1 Datasets, Training, Baselines, and Metrics

**Datasets and network training.** Table 1 lists the datasets used to evaluate Meta-INR. In particular, we utilize the time-varying datasets, including the half-cylinder, ionization, Tangaroa, and vortex datasets, to evaluate the reconstruction accuracy of the Meta-INR compared with other baseline methods. The time-varying earthquake dataset is leveraged to showcase the interpretability of the parameters of adapted INRs under t-SNE projection. We

Table 2: Average PSNR (dB), LPIPS, CD values across all timesteps, and overall encoding time (ET), pretraining time (PT), and total time (TT) for encoding different time-varying datasets. "p.t." stands for pretrained. The chosen isovalues for computing CD are $0.0$, $-0.6$, $-0.8$, and $-0.2$, respectively. 's', 'm', and 'h' denote seconds, minutes, and hours. The best metric values are shown in bold.

| dataset | method | PSNR↑ | LPIPS↓ | CD↓ | ET | PT | TT |
|---|---|---|---|---|---|---|---|
| half-cylinder | SIREN | 48.89 | 0.0729 | 0.5081 | 4h 28m | — | 4h 28m |
| | p.t. SIREN | 35.87 | 0.1113 | 2.6655 | 42m 19s | 2h 39m | 3h 21m |
| | Meta-INR | **55.92** | **0.0705** | **0.2989** | 43m 45s | 2h 14m | 2h 58m |
| ionitzation | SIREN | 48.43 | 0.0591 | 0.5534 | 21h 7m | — | 21h 7m |
| | p.t. SIREN | 40.67 | 0.1189 | 1.6666 | 3h 21m | 11h 51m | 15h 12m |
| | Meta-INR | **51.30** | **0.0576** | **0.4622** | 3h 15m | 11h 28m | 14h 43m |
| Tangaroa | SIREN | 43.22 | 0.1989 | 1.3144 | 3h 29m | — | 3h 29m |
| | p.t. SIREN | 40.46 | 0.3387 | 16.075 | 33m 23s | 2h 10m | 2h 43m |
| | Meta-INR | **48.68** | **0.1969** | **0.5499** | 33m 59s | 2h 7m | 2h 41m |
| vortex | SIREN | 46.06 | 0.0358 | 0.2484 | 24m 22s | — | 24m 22s |
| | p.t. SIREN | 26.22 | 0.2541 | 2.0146 | 5m 17s | 22m 49s | 28m 06s |
| | Meta-INR | **48.19** | **0.0294** | **0.2029** | 5m 10s | 18m 44s | 23m 54s |

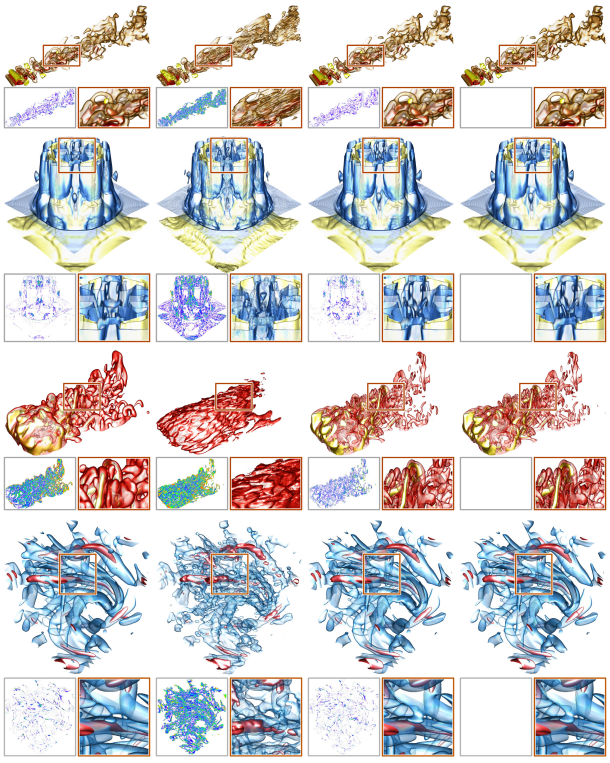



Figure 1: Comparing different methods on volume rendering results. Top to bottom: half-cylinder, ionization, Tangaroa, and vortex.

demonstrate the ability of Meta-INR on simulation parameter analysis using the ensemble Nyx dataset with three simulation parameters: $h_0$, $OmM_0$, and $OmB_0$. Meta-INR uses a seven-layer SIREN model [17] as its network backbone, with a hidden layer dimension of 256. It is trained across all experiments with 500 outer steps and sets the number of inner-loop steps $K = 16$. We set both inner and outer loop learning rates $\alpha$ and $\beta$ to 0.0001 during meta-pretraining and 0.00001 during volume-specific finetuning. Both stages optimize the parameters with a batch size of 50,000 coordinate-value pairs for each iteration.

**Baselines.** We compare Meta-INR with two baseline strategies:

- SIREN [17] is the backbone network architecture of Meta-INR. Here, SIREN as a baseline method means training each INR network independently for each volume from scratch.
- Pretrained SIREN is a vanilla pretraining baseline method that optimizes the initial parameters on the same subsampled dataset without leveraging meta-learning techniques (i.e., no inner-loop updating). After pretraining, we finetune the learned initial parameters using the same number of adaptation steps as the Meta-INR for fair comparisons.

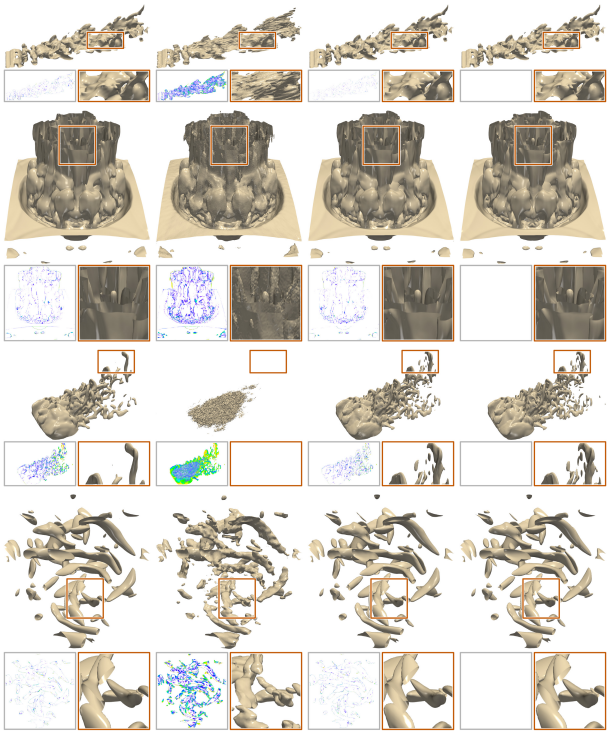


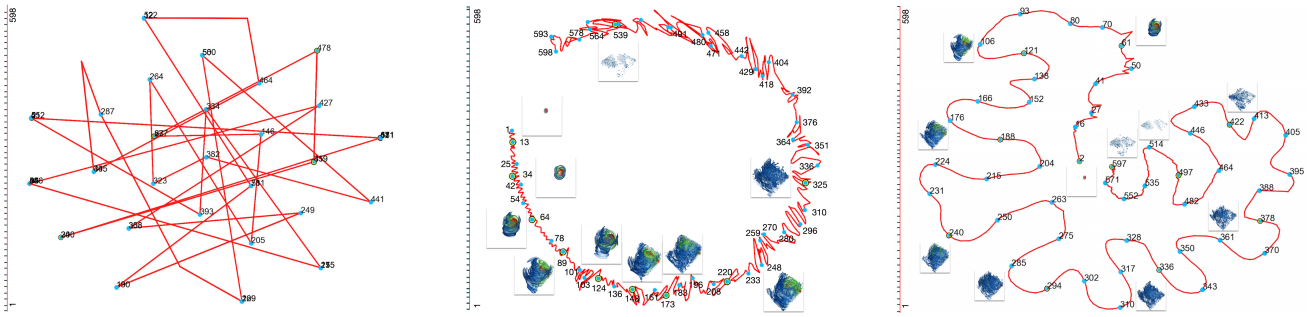

Figure 2: Comparing different methods on isosurface rendering results. Top to bottom: half-cylinder, ionization, Tangaroa, and vortex. The chosen isovalues are reported in Table 2.

**Evaluation metrics.** We use three metrics to evaluate the reconstruction accuracy of Meta-INR and baseline methods. We utilize *peak signal-to-noise ratio* (PSNR) to measure the volume reconstruction accuracy, and *learned perceptual image patch similarity* (LPIPS) to evaluate rendered image qualities for volumes reconstructed by different methods. *Chamfer distance* (CD) is used to assess similarity at the surface level by calculating the average distance of isosurfaces.

### 4.2 Time-Varying Data Representation

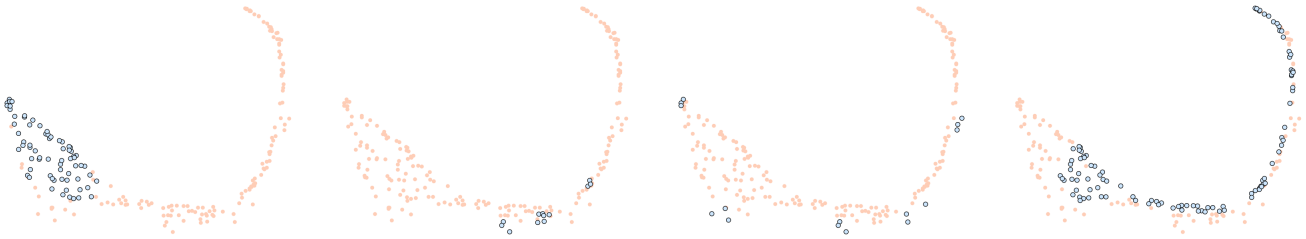
Time-varying data representation directly evaluates the effectiveness of Meta-INR in speeding up model training and improving reconstruction fidelity. We consider all methods on four datasets with various dimensions, as seen in Table 1.

**Quantitative comparison.** In Table 2, we report quantitative results of all different methods. Meta-INR performs the best across the three quality metrics for all datasets. Although SIREN achieves decent quality, sometimes comparable to Meta-INR, the need to retrain a model from scratch for each volume implies that a significant encoding time is required for the model to converge. For Meta-INR, despite the model taking the majority of total time for meta-pretraining, only a few adaptation steps are needed for encoding, saving considerable time compared to SIREN. In particular, the encoding time of adapted INRs is 5.87× faster on average across all datasets than SIREN, which trains each INR from scratch. Pretrained SIREN performs the worst in terms of the three quality metrics as it fails to capture the intrinsic patterns from partial observation of volumetric datasets and struggles to converge efficiently during encoding. Through diverse evaluation across



(a) SIREN [17] (b) autoencoder [13] (c) Meta-INR

Figure 3: t-SNE projections using parameters from different methods trained on the time-varying earthquake dataset, where selected timesteps are marked on the timeline on the left. SIREN's model parameters are not interpretable and fail to establish a correlation between timesteps. Volume rendering images are shown adjacent to selected timesteps for autoencoder and Meta-INR.



(a) $h_0 = 550,000$ (b) $h_0 = 700,000$ (c) $OmM_0 = 120,000$ (d) $OmM_0 = 155,000$

Figure 4: t-SNE projections of Meta-INR models trained on the ensemble Nyx dataset where each point represents a volume. Volumes with $h_0$ or $OmM_0$ matching the specified values are highlighted in light blue.

multiple datasets varying in size and complexity, we found Meta-INR to be highly adaptable. Meta-INR's generalizability stems from its meta-learning framework, which uses inner loop adaptation during meta-pretraining that mimics the finetuning process, forcing parameters into a region where a small number of gradient steps can minimize task-specific loss. This design ensures the prior captures shared structural patterns while remaining sensitive to volume-specific variations.

**Qualitative comparison.** In Figures 1 and 2, we compare the volume and isosurface rendering results for different methods. A pixel-wise difference image is also provided on the bottom left to show the differences between each method and the GT. Pretrained SIREN performs the worst among all three methods, showing significant deviations from the GT. In many cases, it demonstrates block-like artifacts. Both SIREN and Meta-INR demonstrate excellent visual fidelity for the reconstructed volumes. Although they perform similar results in simple datasets such as vortex, Meta-INR can achieve significantly higher accuracy for complex ones with many details, such as Tangaroa. This is because such details are already captured in the meta-model, which serves as the prior for quick adaptation. Then, during volume-specific finetuning, only slight parameter adaptation is needed.

### 4.3 Representative Timestep Selection

We showcase the interpretability of Meta-INR by analyzing its ability on the representative timestep selection task of the earthquake dataset. Porter et al. [13] demonstrated the effectiveness of deep learning techniques for selecting representative timesteps. In our scenario, an INR network trained on one specific volume can also be considered an alternative representation of that volume. However, as shown in Figure 3, when we use t-SNE [23] to project the learned parameters of each SIREN at each timestep to a 2D space and connect the points in the order of timesteps, the resulting projection view is meaningless and offers no interpretability. This is because of significantly increased noise and randomness in training, which leads to models finding drastically different local minima. In contrast, Meta-INR's volume-specific finetuning starts with a common prior, eliminating most noise and focusing on each volume's differences. In particular, the connected points of SIREN across different timesteps lack continuity, resulting in numerous sharp turnings. As SIREN encodes volumes at each timestep independently from scratch, their parameter representations fail to capture a smooth transition of the dataset along the time dimension. Unlike SIREN, the connected points are meaningful and smooth when leveraging t-SNE to project the parameters of adapted INRs finetuned on each timestep. We select representative timesteps following [13] and observe reasonable results. Moreover, we can see that the connected points of Meta-INR are smoother than the results obtained by [13] using an autoencoder. This difference arises from the need to downsample the data before training the autoencoder. Unlike Meta-INR, the network architecture of the autoencoder is constrained by volume dimensions and cannot directly process high-resolution volumetric data. Therefore, the connected points for the projections of the autoencoder are less smooth due to the noise introduced from downsampling.

### 4.4 Simulation Parameter Analysis

When analyzing ensemble datasets, it is often insightful to see how changes in each parameter affect the simulation. Meta-INR can aid in this process by effectively visualizing the relative differences between each volume via t-SNE projection. We apply Meta-INR to the Nyx dataset with three simulation parameters, $h_0$, $OmM_0$, and $OmB_0$. Similar to the time-varying datasets, we perform meta-pretraining on the subsampled Nyx dataset, which is downsampled along the spatial and ensemble dimensions, and then conduct volume-specific finetuning to fit all volumes corresponding to different simulation parameters. In Figure 4, we visualize the pattern in model parameters using t-SNE and mark t-SNE projections sharing the same parameter values. In (a) and (b), we highlight all projections with $h_0$ equal to 550,000 and 700,000, respectively. We can see that $h_0 = 550,000$ corresponds to projections centralized on the left side, and the projections of $h_0 = 700,000$ are centralized

at the bottom of the plot. Similarly, in (c) and (d), which mark projections with $OmM_0$ equal 120,000 and 155,000, respectively, we observe that $OmM_0 = 120,000$ corresponds to projections located at the outer region of the plot. On the other hand, $OmM_0 = 155,000$ corresponds to projections close to the inner region. These results show that the parameters of adapted INRs assimilate information about the simulation parameters during the encoding process.

## 5 Conclusions and Future Work

We have presented Meta-INR, a pretraining method designed to optimize a meta-model that can adapt to unseen volume data efficiently. The generalizability of the meta-model allows for fast convergence during volume-specific finetuning while retaining the model interpretability within its parameters. The evaluation of various volumetric data representation tasks demonstrates better quantitative and qualitative performance of the meta-pretraining than other training strategies.

For future work, we would like to explore continual learning to improve the capabilities of the meta-model for handling more complex time-varying volume data with significantly larger timesteps and variations. Moreover, we want to investigate meta-pretraining on grid-based INRs such as fV-SRN [25] or APMGSRN [27]. Finally, we plan to incorporate transfer learning techniques to learn potentially more challenging variations across variables for multivariate datasets, allowing a meta-model trained on one variable sequence to be effectively used in finetuning another.

## Acknowledgments

This research was supported in part by the U.S. National Science Foundation through grants IIS-1955395, IIS-2101696, OAC-2104158, and IIS-2401144, and the U.S. Department of Energy through grant DE-SC0023145. The authors would like to thank the anonymous reviewers for their insightful comments.

## References

[1] A. S. Almgren, J. B. Bell, M. J. Lijewski, Z. Lukić, and E. Van Andel. Nyx: A massively parallel AMR code for computational cosmology. *The Astrophysical Journal*, 765(1):39, 2013. doi: 10.1088/0004-637X/765/1/39 2

[2] E. Dupont, H. Loya, M. Alizadeh, A. Golinski, Y. W. Teh, and A. Doucet. COIN++: Neural compression across modalities. *Transactions on Machine Learning Research*, 2022, 2022. 2

[3] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of International Conference on Machine Learning*, pp. 1126–1135, 2017. doi: 10.5555/3305381.3305498 2

[4] P. Gu, D. Z. Chen, and C. Wang. NeRVI: Compressive neural representation of visualization images for communicating volume visualization results. *Computers & Graphics*, 116:216–227, 2023. doi: 10.1016/J.CAG.2023.08.024 1

[5] J. Han and C. Wang. CoordNet: Data generation and visualization generation for time-varying volumes via a coordinate-based neural network. *IEEE Transactions on Visualization and Computer Graphics*, 29(12):4951–4963, 2023. doi: 10.1109/TVCG.2022.3197203 1

[6] J. Han, H. Zheng, and C. Bi. KD-INR: Time-varying volumetric data compression via knowledge distillation-based implicit neural representation. *IEEE Transactions on Visualization and Computer Graphics*, 30(10):6826–6838, 2024. doi: 10.1109/TVCG.2023.3345373 1

[7] T. M. Hospedales, A. Antoniou, P. Micaelli, and A. J. Storkey. Meta-learning in neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):5149–5169, 2022. doi: 10.1109/TPAMI.2021.3079209 1

[8] H. Li and H.-W. Shen. Improving efficiency of iso-surface extraction on implicit neural representations using uncertainty propagation. *IEEE Transactions on Visualization and Computer Graphics*, 31(1):1–13, 2024. doi: 10.1109/TVCG.2024.3365089 1

[9] Y. Lu, P. Gu, and C. Wang. FCNR: Fast compressive neural representation of visualization images. In *Proceedings of IEEE VIS Conference (Short Papers)*, pp. 31–35, 2024. doi: 10.1109/VIS55277.2024.00014 1

[10] Y. Lu, K. Jiang, J. A. Levine, and M. Berger. Compressive neural representations of volumetric scalar fields. *Computer Graphics Forum*, 40(3):135–146, 2021. doi: 10.1111/cgf.14295 1

[11] A. Nichol, A. Joshua, and J. Schulman. On first-order meta-learning algorithms. *arXiv:1803.02999*, 2018. doi: 10.48550/arXiv.1803.02999 2

[12] S. Popinet, M. Smith, and C. Stevens. Experimental and numerical study of the turbulence characteristics of airflow around a research vessel. *Journal of Atmospheric and Oceanic Technology*, 21(10):1575–1589, 2004. doi: 10.1175/1520-0426(2004)021<1575:EANSOT>2.0.CO;2 2

[13] W. P. Porter, Y. Xing, B. R. von Ohlen, J. Han, and C. Wang. A deep learning approach to selecting representative time steps for time-varying multivariate data. In *Proceedings of IEEE VIS Conference (Short Papers)*, pp. 131–135, 2019. doi: 10.1109/VISUAL.2019.8933759 4

[14] I. B. Rojo and T. Günther. Vector field topology of time-dependent flows in a steady reference frame. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):280–290, 2019. doi: 10.1109/TVCG.2019.2934375 2

[15] D. Silver and X. Wang. Tracking and visualizing turbulent 3D features. *IEEE Transactions on Visualization and Computer Graphics*, 3(2):129–141, 1997. doi: 10.1109/2945.597796 2

[16] V. Sitzmann, E. Chan, R. Tucker, N. Snavely, and G. Wetzstein. MetaSDF: Meta-learning signed distance functions. In *Proceedings of Advances in Neural Information Processing Systems*, pp. 10136–10147, 2020. doi: 10.48550/arXiv.2006.09662 2

[17] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein. Implicit neural representations with periodic activation functions. In *Proceedings of Advances in Neural Information Processing Systems*, pp. 7462–7473, 2020. doi: 10.48550/arXiv.2006.09661 1, 3, 4

[18] Y. Song, T. Wang, P. Cai, S. K. Mondal, and J. P. Sahoo. A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities. *ACM Computing Surveys*, 55(13s):1–40, 2023. doi: 10.1145/3582688 2

[19] M. Tancik, B. Mildenhall, T. Wang, D. Schmidt, P. P. Srinivasan, J. T. Barron, and R. Ng. Learned initializations for optimizing coordinate-based neural representations. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2846–2855, 2021. doi: 10.1109/CVPR46437.2021.00287 2

[20] K. Tang and C. Wang. ECNR: Efficient compressive neural representation of time-varying volumetric datasets. In *Proceedings of IEEE Pacific Visualization Conference*, pp. 72–81, 2024. doi: 10.1109/PacificVis60374.2024.00017 1

[21] K. Tang and C. Wang. STSR-INR: Spatiotemporal super-resolution for time-varying multivariate volumetric data via implicit neural representation. *Computers & Graphics*, 119:103874, 2024. doi: 10.1016/j.cag.2024.01.001 1

[22] K. Tang and C. Wang. StyleRF-VolVis: Style transfer of neural radiance fields for expressive volume visualization. *IEEE Transactions on Visualization and Computer Graphics*, 31(1):613–623, 2025. doi: 10.1109/TVCG.2024.3456342 1

[23] L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. 4

[24] C. Wang and J. Han. DL4SciVis: A state-of-the-art survey on deep learning for scientific visualization. *IEEE Transactions on Visualization and Computer Graphics*, 29(8):3714–3733, 2023. doi: 10.1109/TVCG.2022.3167896 1

[25] S. Weiss, P. Hermüller, and R. Westermann. Fast neural representations for direct volume rendering. *Computer Graphics Forum*, 41(6):196–211, 2022. doi: 10.1111/cgf.14578 1, 5

[26] D. Whalen and M. L. Norman. Ionization front instabilities in primordial H II regions. *The Astrophysical Journal*, 673:664–675, 2008. doi: 10.1086/524400 2

[27] S. W. Wurster, T. Xiong, H.-W. Shen, H. Guo, and T. Peterka. Adaptively placed multi-grid scene representation networks for large-scale data visualization. *IEEE Transactions on Visualization and Computer Graphics*, 30(1):965–974, 2024. doi: 10.1109/TVCG.2023.3327194

1, 5

[28] T. Xiong, S. W. Wurster, H. Guo, T. Peterka, and H.-W. Shen. Regularized multi-decoder ensemble for an error-aware scene representation network. *IEEE Transactions on Visualization and Computer Graphics*, 31(1):645–655, 2025. doi: 10.1109/TVCG.2024.3456357 1

[29] S. Yao, Y. Lu, and C. Wang. ViSNeRF: Efficient multidimensional neural radiance field representation for visualization synthesis of dynamic volumetric scenes. In *Proceedings of IEEE Pacific Visualization Conference*, 2025. Accepted. 1