




MoE-INR: Implicit Neural Representation with Mixture-of-Experts for Time-Varying Volumetric Data Compression

Jun Han , Kaiyuan Tang , and Chaoli Wang 

Abstract—Implicit neural representations (INRs) have emerged as a transformative paradigm for time-varying volumetric data compression and representation, owing to their ability to model high-dimensional signals effectively. INRs represent scalar fields based on sampled coordinates, typically using either a single network for the entire field or multiple networks across different spatial domains. However, these approaches often face challenges in modeling complex patterns and introducing boundary artifacts. To address these limitations, we propose MoE-INR, an INR architecture based on a mixture-of-experts (MoE) framework. MoE-INR automates irregular subdivisions of spatiotemporal fields and dynamically assigns them to different expert networks. The architecture comprises three key components: a policy network, a shared encoder, and multiple expert decoders. The policy network subdivides the field and determines which expert decoder is responsible for a given input coordinate. The shared encoder extracts hidden representations from the input coordinates, and the expert decoders transform these high-dimensional features into scalar values. This design results in a unified framework accommodating diverse INR types, including conventional, grid-based, and ensemble. We evaluate the effectiveness of MoE-INR on multiple time-varying datasets with varying characteristics. Experimental results demonstrate that MoE-INR significantly outperforms existing non-MoE and MoE-based INRs and traditional lossy compression methods across quantitative and qualitative metrics under various compression ratios.

Index Terms—Time-varying data compression, implicit neural representation, volume visualization, mixture-of-experts

1 INTRODUCTION

Implicit neural representations (INRs) are a class of neural networks that encode a set of coordinates into high-dimensional signals. INRs have gained popularity and attention in scientific visualization because they can effectively fit volumetric fields for data representation and compression [22, 35, 38]. However, existing INRs for scientific data compression, while powerful, are often limited by their architectures. Conventional INRs [10, 22] utilize a single network to model the entire field, making it difficult to preserve complex features. Grid-based INRs [12, 32] manually decompose the field into multiple spatial blocks and apply multiple neural networks to represent each block, causing boundary discontinuity and lacking the capability to compress with few parameters. In this paper, we aim to develop a unified INR architecture that can automatically subdivide the field and represent complex patterns in each subdivision.

Several challenges remain in developing a unified INR. First, it is difficult to unify both regular and irregular partitions without incurring additional storage costs. While existing methods focus on regular subdivisions—such as in space [11] or time [12]—irregular decompositions typically require extensive storage to record grouping results, making them impractical for compression. Second, avoiding boundary discontinuity across the partition remains a challenge. Partitioning disrupts the assumption of cross-boundary continuity, often resulting in visible artifacts along subdivision edges. Third, an effective INR must independently capture diverse and complex features without compromising its ability to model other regions. Achieving this is essential to maintain consistent, high-quality results across all subdivisions, especially in domains with heterogeneous characteristics.

Recent advances in *large language models* (LLMs) [16, 33] have

leveraged the *mixture-of-experts* (MoE) paradigm to partition inputs and allow the network to focus on distinct regions or features, enabling more effective and specialized learning. In MoE, a classifier groups signals and routes those within the same class to a dedicated decoder for prediction. Inspired by this, we propose MoE-INR, an INR enhanced with a novel MoE framework for compressing time-varying volumetric data. By incorporating MoE into INR, the field is decomposed into multiple regular or irregular subdivisions, enabling each expert to specialize in a particular feature or group of similar features. This specialization reduces the learning burden on individual models and improves their capacity to capture fine-grained details. The modular structure of MoE-INR promotes both efficiency and adaptability in representing complex, time-varying scientific data. Moreover, its flexible design accommodates various existing INR structures, including: single encoder-decoder models (e.g., CoordNet [10], NeurComp [22]); grid-based models that partition the field into multiple subregions (e.g., DCINR [11], ECNR [32]); and ensemble-based models using multiple decoders for uncertainty quantification (e.g., RMDSRN [39]). A summary comparing MoE-INR with these representative INR architectures is provided in Table 1.

Table 1: Properties of mainstream INR techniques for time-varying scientific data compression and representation.

INR architecture	data partition	boundary discontinuity	compression quality	compression speed	uncertainty estimation	temporal coherence	inference time
MoE-INR (ours)	arbitrary	no	high	middle	yes	high	slow
CoordNet [10]	no	no	middle	slow	no	high	slow
NeurComp [22]	no	no	middle	slow	no	high	slow
Devkota et al. [7]	no	no	middle	fast	no	low	fast
RMDSRN [39]	no	no	low	fast	yes	low	fast
IV-SRN [35]	no	no	low	fast	no	low	fast
APMGSRN [38]	no	no	low	fast	no	low	fast
KD-INR [12]	time	no	middle	slow	no	high	slow
DCINR [11]	space	yes	high	fast	no	high	fast
ECNR [32]	space	yes	low	middle	no	middle	fast
Wu et al. [37]	space	yes	low	fast	no	low	fast

The effectiveness of MoE-INR is demonstrated on a variety of time-varying datasets. Data-, image-, and isosurface-level metrics are applied to evaluate MoE-INR. Both visual quality and quantitative values show that our approach significantly outperforms different types of state-of-the-art INRs and traditional lossy compression methods for time-varying data compression under various compression ratios, ranging from thousands to tens of thousands. Our work takes an important step towards making neural representations more flexible and effective for scientific data compression.

The key contributions of this work are summarized below:

- We propose a unified INR that accommodates various types of

- J. Han is with the Division of Emerging Interdisciplinary Areas and the Center for Ocean Research in Hong Kong and Macau (CORE), The Hong Kong University of Science and Technology, Hong Kong, China. E-mail: hanjun@ust.hk.
- K. Tang and C. Wang are with Department of Computer Science and Engineering, the University of Notre Dame, Notre Dame, IN, USA. Email: {ktang2, chaoli.wang}@nd.edu.

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxx/TVCG.201x.xxxxxx

INRs, including conventional, grid-based, and ensemble ones.

- We develop a novel MoE with a pre-training scheme that improves learning capability and compression effectiveness.
- We thoroughly evaluate MoE-INR under different ablations and show its superior performance under various compression ratios.

2 RELATED WORK

This section overviews related works on lossy compression, deep learning for scalar field reduction, and MoE.

Lossy compression. Lossy compression is a critical solution for scientific data management and compression due to the increasing capability of running large-scale simulations on high-performance computing platforms. Lindstrom developed ZFP [21], a data reduction algorithm designed to enhance input/output performance through block segmentation. Soler et al. [30] crafted a compression technique with topological oversight, imposing constraints on the discrepancy between the persistence diagrams of the original and compressed data. Ballester et al. developed TTHRESH [3], a tensor decomposition-based compressor that achieves state-of-the-art performance among existing lossy compressors. Ainsworth et al. [1] proposed a compression framework for adaptive data representation that ensures user-specified tolerance. Xin et al. proposed SZ3 [20], a modular compression framework that automatically selects the best-fit predictor for each spatial block given a user-specified error. Yan et al. introduced TopoSZ [40], which preserves topological features for scalar field compression.

Deep learning for scalar field reduction. Due to the remarkable achievements in deep learning-based data representation, the scientific visualization community has explored such techniques for data reduction in recent years [34]. Lu et al. developed NeurComp [22], an INR-based compressor with weight quantization, to represent a single scalar field. Han and Wang introduced CoordNet [10], a coordinate-based neural network for handling scientific data generation and visualization generation tasks. Weiss et al. presented fV-SRN [35], using GPU tensor cores to integrate the INR reconstruction into raytracing kernels for fast volume visualization. Wurster et al. developed APMGSRN [38], leveraging adaptive feature grids in INR to fit regions of high complexity, which helps INR perform better at complex regions. Han et al. established KD-INR [12], a two-stage compression pipeline, enabling sequential compression through knowledge distillation. Tang and Wang presented ECNR [32], incorporating the Laplacian pyramid into INRs for compressing time-varying data through hierarchical representation. Devkota et al. [7] introduced an INR with hash encoding for compressing a single scalar field. Wu et al. [37] proposed an INR framework that decomposed a field into multiple spatial domains, applied INRs to model each domain, and utilized lossy compression to reduce the size of INRs. Xiong et al. designed RMDSRN [39], an ensemble INR for scalar field representation with uncertainty estimation. In the above INRs, the data is either represented by a single INR or modeled by multiple INRs across spatial regions. However, our work provides a unified framework that supports automatic data partition and better pattern preservation learned by MoE for time-varying volumetric data compression.

Mixture-of-experts. MoE was proposed by Nowlan and Hinton [26], which applied supervision for a system composed of separate networks. In MoE, each network handles a subset of the training samples and specializes in a different part of the data space. Due to its effectiveness, MoE has been widely used in different tasks. For example, Fedus et al. [8] simplified the MoE routing algorithm and incorporated it into an LLM for natural language processing tasks. Mustafa et al. presented LIMoE [25], a language-image MoE, enabled to understand multimodal information for better data representation. Cao et al. [6] designed a mixture of local experts and a mixture of global experts to learn local and global information from images for image fusion tasks. Zhao et al. [43] proposed a MoE INR to compress a single medical volume. Mi and Xu proposed Switch-NeRF [23], a *neural radiance field* (NeRF) with MoE, to automatically decompose a scene into multiple parts for scene rendering. Ben et al. introduced Neural Experts [5], an INR equipped with MoE, to model image, audio, and mesh by offering supervision for expert assignment. Yu et al. [41] leveraged MoE to

boost the performance of continual learning in vision-language models. Unlike the aforementioned works, which apply MoE to images, medical data, or language, our work introduces a novel MoE framework within an INR architecture specifically designed to represent and compress time-varying volumetric data.

3 MoE-INR

In this section, we provide an overview of MoE-INR. Then, detailed model design rationale and architecture are offered. Finally, the optimization procedure is described.

3.1 Overview

Figure 1 illustrates the MoE-INR framework, which consists of three key modules: a policy network, a shared encoder, and multiple expert decoders. The policy network automatically subdivides the field and stores the grouping results, reducing the need for explicitly recording the mapping between coordinates and their clusters, particularly beneficial for irregular decompositions. This subdivision can be optimized in either a supervised or unsupervised manner. A shared encoder processes all coordinates (x, y, z, t) across subfields to learn high-dimensional representations, preserving boundary continuity and eliminating artifacts during decompression. Each expert decoder is responsible for a distinct, non-overlapping subdomain of the voxel space, as determined by the classification output of the policy network. By assigning each decoder to independently learn a specific region, the learning burden is distributed, allowing for more effective modeling of complex features within each subdomain.

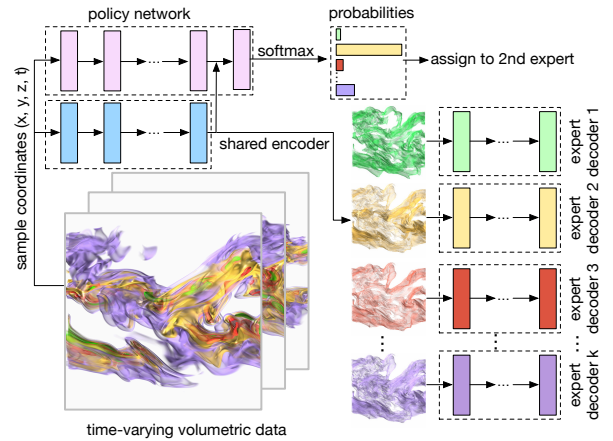


Fig. 1: The overview of our MoE-INR framework. Given a coordinate, the policy network decides which expert decoder will process according to the predicted probabilities. A shared encoder will learn a high-dimensional representation of the coordinate and feed it to the corresponding expert decoder for decompression.

3.2 Network Design Exploration

We provide the detailed design and function of each module in MoE-INR and offer the design and hyperparameter choice through ablation and hyperparameter studies for MoE-INR.

Policy network. The policy network is designed to accurately and automatically classify coordinates into different expert decoders, with each expert decoder specializing in handling one or more specific patterns. Drawing inspiration from the design of MoE in scene rendering [23] and image representation [5], our policy network consists of several fully connected layers with a $\sin(\cdot)$ activation function [29] and one fully connected layer with a softmax activation function. The softmax function can convert the feature representation into a set of probabilities, which can be used for determining the decoder assignment. The concatenated representation, derived from the outputs of the policy network and the shared encoder, will be fed into the last fully-connected layer. The decoder corresponding to the highest probability is then selected and activated to decompress the input coordinate.

Shared encoder. The shared encoder has two primary purposes. First, the representation learned by the encoder is injected into the

Table 2: Average PSNR, average LPIPS, compression time (CT), and memory usage (MU) under different parameter allocations using the vortex dataset. The compression ratio (CR) is 5,065. The best one is highlighted in bold.

allocation type	shared encoder	expert decoders	policy network	PSNR (dB) ↑	LPIPS ↓	CT (hours) ↓	MU (MB) ↓
larger encoder	96.90%	0.82%	2.28%	47.13	0.023	8.72	17,845
larger decoder	7.71%	90.22%	2.07%	31.71	0.188	8.79	17,921
balanced allocation	49.34%	48.43%	2.23%	41.44	0.054	8.74	17,663

Table 3: Average PSNR, average LPIPS, CT, and MU under different numbers of experts using the ionization (PD) dataset. The CR is 10,078. The best one is highlighted in bold.

# experts	PSNR (dB) ↑	LPIPS ↓	CT (hours) ↓	MU (MB) ↓
2	54.05	0.046	11.83	8,177
3	54.34	0.045	12.86	10,067
4	55.01	0.041	17.25	12,085
5	55.52	0.039	18.09	13,945
6	56.31	0.038	20.78	15,929
7	58.76	0.024	21.59	17,913
8	58.75	0.024	24.23	19,901
9	58.42	0.025	25.36	21,745

policy network to assist the decoder assignment. Second, this representation serves as the input to the experts, enabling accurate decoding of voxel values. Inspired by recent advances in NeRFs [24, 31, 44], we employ a *positional encoding* (PE) function, i.e., a learnable Fourier transformation, to map low-dimensional coordinates into high-dimensional representations. These enriched embeddings are then passed through a series of fully connected layers with $\sin(\cdot)$ activation functions, progressively increasing the feature dimensionality within the encoder. Finally, a residual block with bottleneck [15] is followed to refine the learned representation into a compact one.

Expert decoders. Expert decoders take the high-dimensional representation produced by the shared encoder as input and predict the voxel values at the corresponding coordinates. Based on the probabilities predicted by the policy network, only a single expert decoder is activated to decode the representation into the original data space. While all expert decoders share the same architecture, they utilize different parameter sets. Each decoder consists of several fully connected layers with a $\sin(\cdot)$ activation function, followed by a final fully connected layer without an activation function.

Parameter allocation. Given that MoE-INR consists of three different modules, a key question arises regarding how to allocate the parameters to each module. To identify the optimal configuration, we explore three different allocation strategies:

- **A larger shared encoder:** most parameters are allocated to the shared encoder.
- **Larger expert decoders:** most parameters are assigned to the expert decoders.
- **Balanced distribution:** parameters are evenly distributed between the shared encoder and the expert decoders.

Since the reconstruction process is carried out by the encoder and expert decoders rather than the policy network, we do not consider cases where the number of parameters in the policy network exceeds that of the other two modules. Table 2 reports the average PSNR, average LPIPS, total compression time (CT) in hours, and memory usage (MU) in MB for each allocation strategy. The compression ratio (CR) equals the data size divided by the model size. Although these three parameter allocations share a similar CT and MU, the allocation with a large encoder achieves the best performance regarding PSNR and LPIPS values. This performance advantage can be attributed to the encoder’s dual roles: (1) assisting the policy network in accurately assigning experts and (2) providing a compact and precise representation to the decoder for decompression. In contrast, the other two modules are responsible for only one task. Allocating more parameters to the encoder ensures it has sufficient learning capacity to handle both roles effectively. Thus, we chose the architecture with a larger encoder for MoE-INR. The detailed configuration of MoE-INR is shown in Figure 2, where M is the number of initial neurons specified by users. Additionally, the number of parameters, including weights and biases, in MoE-INR is calculated as

$$\begin{aligned} \# \text{ params} = & \underbrace{4M + M + M^2 + M + 9M \times k + k}_{\# \text{ params in the policy network}} \\ & + \underbrace{4M + M + 8M^2 + 4M + 32M^2 + 8M + 36M^2 + 12M}_{\# \text{ params in the shared encoder}} \\ & + \underbrace{k \times (8M + 1)}_{\# \text{ params in } k \text{ expert decoders}} \end{aligned} \quad (1)$$

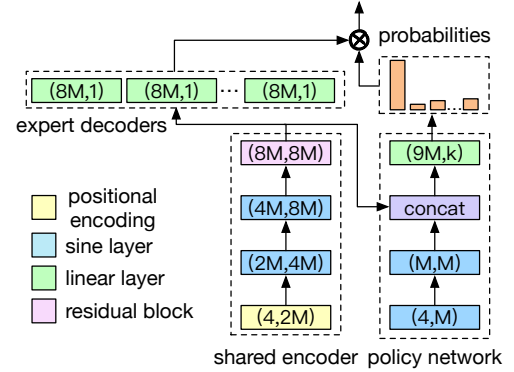


Fig. 2: Architecture details of MoE-INR. (\cdot, \cdot) denotes the input dimension and output dimension in each layer.

Number of experts. An important hyperparameter in MoE-INR is the number of experts. Increasing the number of experts leads to finer subdivisions of the data, which can potentially improve compression quality. However, it also introduces additional complexity in expert assignment, which may degrade overall performance. To identify the optimal trade-off between the number of experts and compression quality, we evaluate MoE-INR with varying numbers of experts, as reported in Table 3. The study shows that with 7 experts, MoE-INR achieves the best PSNR and LPIPS values. When the number of experts exceeds 7, the PSNR value slightly decreases, likely due to the increased difficulty for the policy network to correctly classify and assign coordinates, resulting in more misclassifications. Although adding one extra expert increases CT and MU, the performance improvement is notable, particularly when increasing the number of experts from 6 to 7. Based on these observations, we set 7 experts in MoE-INR for all experiments presented in this paper.

Table 4: Average PSNR and LPIPS values under different pre-training schemes with 7 experts using the ionization (T) dataset. The CR is 5,033. The best one is highlighted in bold.

pre-training scheme	PSNR (dB) ↑	LPIPS ↓
no	51.35	0.090
random partition	37.92	0.230
spatial partition	53.90	0.071
temporal partition	53.93	0.064
voxel clustering	56.71	0.047
load balancing	39.27	0.213

3.3 Optimization

The optimization process includes two stages: *pre-training* and *training*. At the pre-training stage, we optimize the policy network and shared encoder, aiming to offer a good initialization for expert assignment and feature representation. At the training stage, all three modules are jointly optimized for predicting voxel and refining expert assignments given a set of coordinates.

Pre-training. It is crucial for initializing the parameters in INRs for guaranteeing performance [2, 29], and the policy network plays an

Table 5: Relationship between MoE-INR and existing INRs for scientific data compression and representation.

	pre-training scheme	# experts	aggregation	INR type	exemplar INR
MoE-INR	none	1	none	conventional	CoordNet [10], NeurComp [22]
	spatial partition	k	none	grid-based	ECNR [32], DCINR [11]
	none	k	average	ensemble	RMDSRN [39]

essential role in subdividing the field and identifying which expert will handle the input coordinate. For that, we investigate different MoE-INR pre-training schemes. Generally, we consider two types of pre-training: *supervised* and *unsupervised*. For supervised pre-training, we generate an expert assignment y_c for each coordinate c , as ground truth (GT) and train the policy network through a cross-entropy loss, defined as

$$-\log \frac{e^{\mathbf{p}_c}}{\sum_{i=1}^k e^{\mathbf{p}_i}}, \quad (2)$$

where \mathbf{p} is a 1D vector with k elements, outputted from the policy network. \mathbf{p}_j indicates the probability of assigning the given coordinate to the j -th expert. We study four different methods to obtain the GT assignments:

- **Random partition:** we randomly assign one expert for each coordinate.
- **Spatial partition:** we partition the spatial space into k non-overlapping blocks, and each expert handles one spatial block for all time steps.
- **Temporal partition:** we divide the temporal sequence into k time intervals without overlap, and each expert processes one of these intervals for all spatial coordinates.
- **Voxel clustering:** we use the k-means algorithm to group the voxels into k clusters, and each expert is responsible for one cluster.

For unsupervised pre-training, we utilize load balancing [19, 23] as the criterion for grouping coordinates. The rationale behind load balancing is that it encourages evenly distributed assignments to each expert. This ensures that each expert processes a similar number of coordinates, rather than allowing a few experts to handle most of the coordinates. The load balancing loss is formulated as

$$\frac{k}{N^2} \sum_{j=1}^k c_j s_j, \quad (3)$$

where c_j is the number of coordinates assigned to the j -th expert decoder, s_j is the sum of assignment probabilities to the j -th expert decoder for all coordinates, and N is the total number of coordinates.

The pre-training process offers two benefits. First, it guides the policy network on how to group coordinates, rather than guessing from scratch. Second, it balances the assignments among the experts, as it helps prevent favoring some experts while neglecting others.

Table 4 reports the average PSNR and LPIPS values under different pre-training schemes. Clearly, using voxel clustering-based pre-training achieves the best performance. However, random partition and load balancing fail to produce a meaningful assignment to different decoders, leading to inferior performance compared to no pre-training. Thus, we chose voxel clustering as our pre-training scheme. Refer to the Appendix for the statistics and visualization regarding expert assignments.

Training. After pre-training, we jointly optimize the policy network, shared encoder, and expert decoders. This training process allows the policy network to refine expert assignments and updates the shared encoder and expert decoders for reconstruction by measuring the difference between the predicted and GT voxel values. The loss for each voxel value is calculated as

$$\sum_{j=1}^k \mathbf{p}_j \|v_j - v\|_2, \quad (4)$$

where v_j is the predicted value from the j -th expert decoder and v is the GT voxel value.

4 RELATIONSHIP WITH EXISTING INRS

Existing INRs for scientific data representation and compression can be broadly classified into three categories: conventional [10, 12, 22],

grid-based [32, 35, 38], and ensemble [39]. MoE-INR unifies these INRs by offering various data partition schemes and controlling the number of experts, as summarized in Table 5. When a single expert is set in MoE-INR, it reduces to a conventional INR, such as CoordNet and NeurComp, which utilizes a single encoder and decoder to learn the entire field. By leveraging more than two experts in MoE-INR and a spatial-based pre-training scheme, it functions as a grid-based INR, similar to fV-SRN and APMGSRN, which models the field across different spatial domains. However, unlike existing grid-based INRs, which optimize each spatial or spatiotemporal block independently (i.e., each block has one encoder and one decoder) and introduce block artifacts, MoE-INR avoids these artifacts by employing a shared encoder for all blocks. Furthermore, by applying multiple experts to decode a single coordinate and aggregating the results through an average operation, MoE-INR becomes an ensemble INR, such as RMDSRN. In terms of computational cost for visualization, grid-based INRs, such as fV-SRN and APMGSRN, achieve the fastest volume decompression, as their primary operations involve efficient feature interpolation. In contrast, conventional INRs incur the highest computational cost, since volume generation relies entirely on matrix multiplications throughout the network. The design of MoE-INR offers both generalization and flexibility, allowing it to accommodate various INR architecture choices. Refer to Section 5 for detailed quantitative and qualitative comparisons.

Table 6: Summary of each dataset.

dataset	variable	dimension ($x \times y \times z \times t$)	data size (GB)
argon bubble [9]	intensity	$640 \times 256 \times 256 \times 150$	23.44
combustion [14]	CHI, HR, MF, VORT	$480 \times 720 \times 120 \times 100$	15.45
ionization [36]	H+, H2, PD, T	$600 \times 248 \times 248 \times 100$	13.74
Tangaroa [28]	vorticity magnitude	$600 \times 360 \times 240 \times 150$	28.97
vortex [13]	vorticity	$256 \times 256 \times 256 \times 90$	5.63

5 RESULTS

In this section, we provide the configurations at both training and inference stages as well as the quantitative metrics for assessing the quality of decompressed data. Then we compare MoE-INR against various state-of-the-art compressors, including learning-based and lossy compression, illustrate the capability of uncertainty estimation, and evaluate the performance under various CRs.

5.1 Configurations, Metrics, and Baselines

Training and inference configurations. Time-varying volumetric datasets from various domains (e.g., chemistry and physics) and with different characteristics (e.g., turbulence and feature tracking) are considered for evaluation. This broad selection demonstrates the versatility and general applicability of our approach across a wide range of scenarios. Table 6 lists the datasets used for evaluation. For multivariate datasets, each variable is individually compressed. MoE-INR was implemented by PyTorch [27]. The compression and decompression were conducted on an NVIDIA GeForce RTX 4090 GPU. We normalized the coordinates and values into $[-1, 1]$ to fit the output range of the $\sin(\cdot)$ activation function. We initialized all parameters in MoE-INR following Sitzmann et al. [29]. For optimization, the Adam optimizer [17] was applied. We set the initial learning rate to 2×10^{-5} with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The learning rate decayed by half with a multistep scheduler. We pre-trained MoE-INR with 30 epochs and trained with 270 epochs. The 32-bit floating point was used for training and inference, while the 16-bit floating point was used for model storage. The batch size is set to 16,000. Refer to the Appendix for model configurations and the total CT of each dataset under different CRs.

Evaluation metrics. We calculate the error between the decompressed and GT volumes using data-level *peak signal-to-noise* (PSNR), image-level *learned perceptual image patch similarity* (LPIPS) [42], and isosurface-level *chamfer distance* (CD) [4]. LPIPS measures the

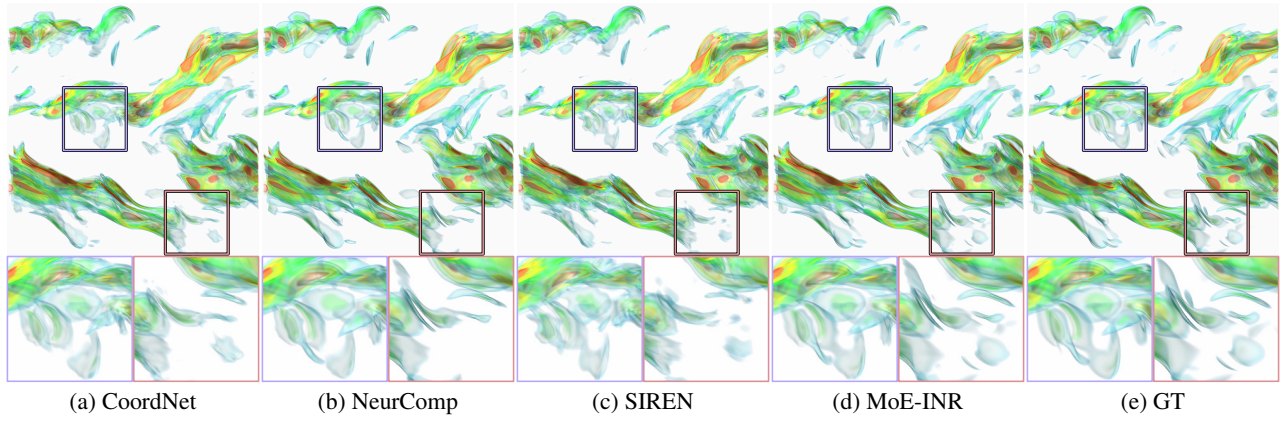


Fig. 3: Comparison of volume rendering results between MoE-INR and conventional INRs using the combustion (CHI) dataset. The CR is 5,053.

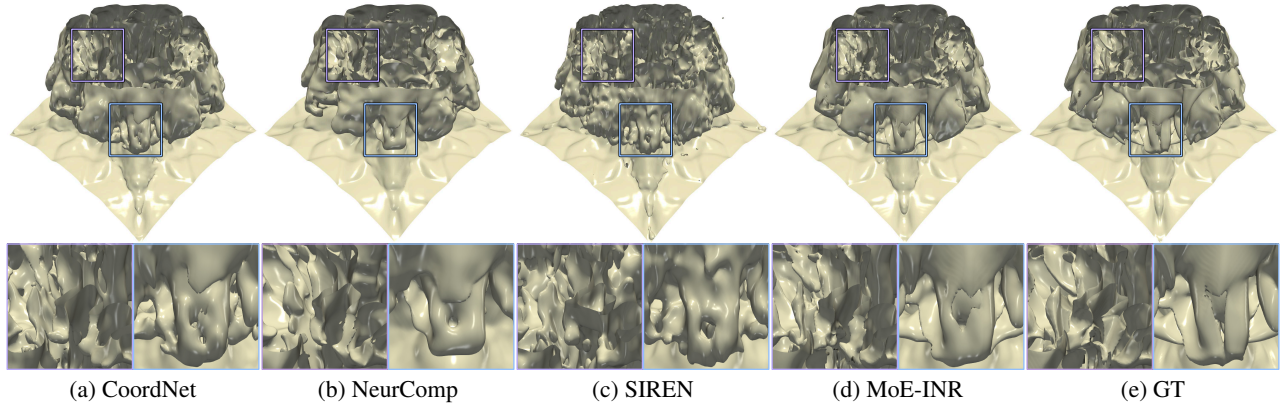


Fig. 4: Comparison of isosurface rendering results between MoE-INR and conventional INRs using the ionization (H+) dataset. The CR is 2,862. The chosen isovalue is -0.98 .

distance between deep features in a deep convolutional neural network extracted from two input images. AlexNet [18] is leveraged for extracting image representation. For CD, it measures the minimum distance of the isosurfaces extracted from the decompressed and GT volumes. The distance is computed by L_2 norm. When considering the PSNR metric, a higher value denotes a superior quality. In contrast, for both LPIPS and CD metrics, superior quality is indicated by a lower score.

Baselines. Our comparison consists of two categories: learning-based compressors and traditional lossy compressors. We compare three conventional INR architectures (i.e., CoordNet [10], NeurComp [22], and SIREN [29]), three grid-based INRs (i.e., ECNR [32], fV-SRN [35], and APMGSRN [38]), and other INRs, including one ensemble INR (i.e., RMDSRN [39]) and two INRs with MoE (i.e., Switch-NeRF [23] and Neural Experts [5]). The grid size for feature interpolation of fV-SRN and APMGSRN is set to $4 \times 4 \times 4$ and $2 \times 2 \times 2$, respectively. The number of neurons in fV-SRN and APMGSRN is identified by a user-specific CR. Switch-NeRF was originally designed as an MoE-based INR for scene rendering, while Neural Experts is for image and mesh representation. We adopt both solutions for time-varying volumetric data compression as MoE-based INR baselines. For traditional lossy compressors, we choose three state-of-the-art solutions, i.e., ZFP [21], SZ3 [20], and TTHRESH [3].

In terms of visualization, we decompress the data from the compressor for each solution and render it via the traditional rendering pipeline. We apply the same rendering setting, i.e., transfer function, lighting condition, and viewpoint, to each dataset for a fair comparison. Refer to the accompanying video for frame-to-frame comparison.

5.2 Comparison against Learning-Based Compressors

Visual comparison against conventional INRs. We compare MoE-INR with three conventional INR-based compressors, i.e., CoordNet, SIREN, and NeurComp, under the same CR. Figure 3 presents the

volume rendering images using the combustion (CHI) dataset. As highlighted in the purple box, CoordNet and SIREN fail to recover the tiny cyan structure with a tail shape, while NeurComp incorrectly splits a single feature into two separate ones. Additionally, Figure 4 illustrates the isosurface rendering images using the ionization (H+) dataset. Similarly, conventional INRs struggle to preserve high-quality geometric patterns in the extracted isosurfaces, highlighted by the blue box. This limitation stems from the reliance of conventional INRs on a single network to learn complex and simple patterns from a field. Consequently, the network prioritizes memorizing simple structures over complex ones, as simple patterns contribute more significantly to reducing the overall loss.

Visual comparison against grid-based INRs. We compare MoE-INR with three grid-based INRs, i.e., APMGSRN, ECNR, and fV-SRN, under the same CR. The volume rendering and isosurface rendering images are displayed in Figures 5 and 6, respectively. It is evident that ECNR exhibits boundary artifacts (i.e., the discontinuity across the boundary edge between two partitioned blocks) and fails to maintain high-quality structures in the rendered images, even under a small CR. The boundary discontinuity is caused by the training strategy (i.e., each partitioned block is independently modeled by an INR encoder) in ECNR, which disrupts the boundary continuity. In contrast, the results from MoE-INR are free from boundary discontinuity. This is attributed to the design of a single encoder in MoE-INR to learn all input coordinates, ensuring continuity across partitioned regions. Although fV-SRN and APMGSRN are free of boundary discontinuity, they require a large feature map to construct a hybrid representation from coordinates and lack the capability to recover volume under few parameters.

Visual comparison against other INRs. We evaluate the performance of MoE-INR by comparing it against one ensemble INR (i.e., RMDSRN) and two INRs with MoE (i.e., Switch-NeRF and Neural

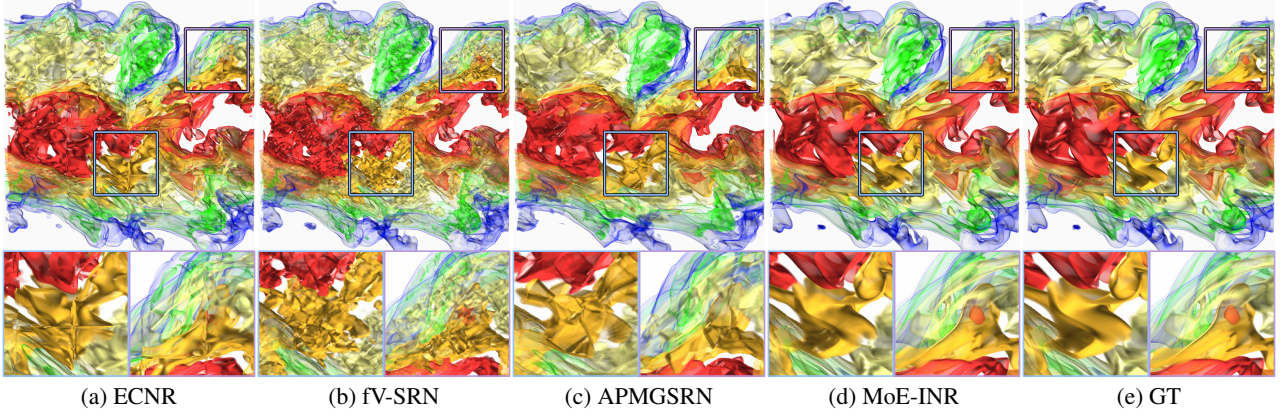


Fig. 5: Comparison of volume rendering results between MoE-INR and grid-based INRs using the combustion (MF) dataset. The CR is 1,951.

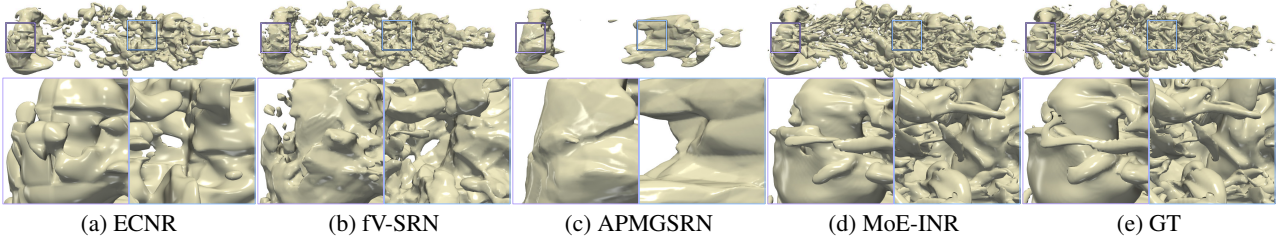


Fig. 6: Comparison of isosurface rendering results between MoE-INR and grid-based INRs using the argon bubble dataset. The CR is 3,470. The chosen isovalue is -0.45 .

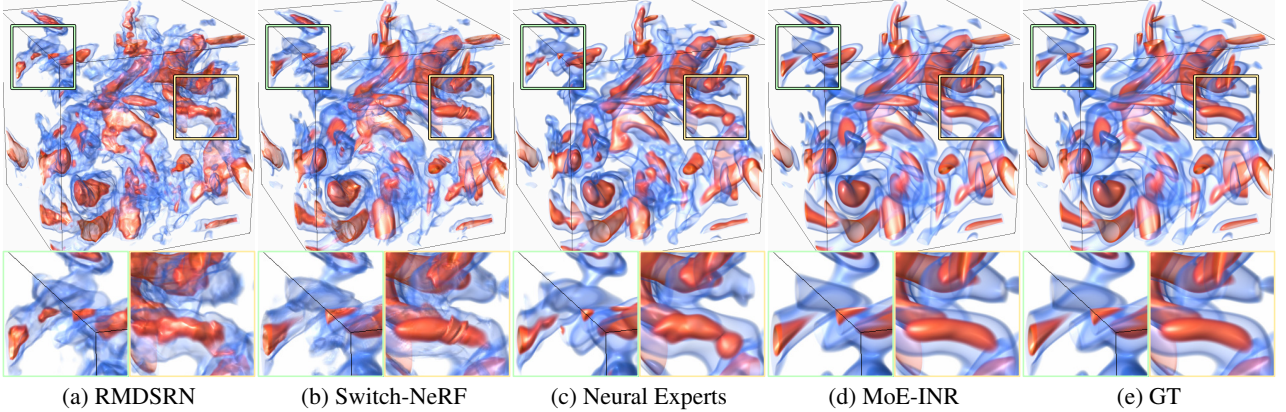


Fig. 7: Comparison of volume rendering results between MoE-INR and other INRs using the vortex dataset. The CR is 2,510.

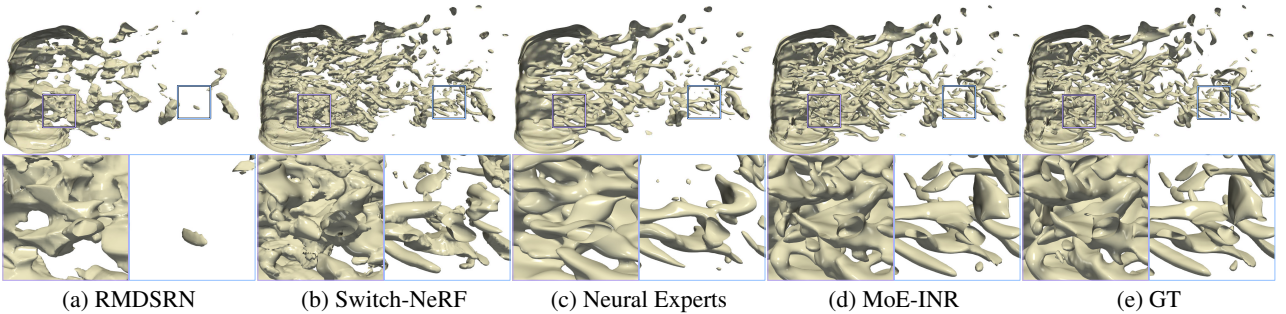


Fig. 8: Comparison of isosurface rendering results between MoE-INR and other INRs using the Tangaroa dataset. The CR is 7,510. The chosen isovalue is -0.72 .

Experts) under the same CR. Figure 7 shows the volume rendering images using the vortex dataset. RMDSRN produces patterns with jagged artifacts, while Switch-NeRF and Neural Experts display noticeable artifacts along the partition boundaries. These issues arise because the policy network in both Switch-NeRF and Neural Experts is designed

for a single dataset and lacks the capability to assign coordinates to different experts for time-varying data. Additionally, Figure 8 presents the isosurface rendering images using the Tangaroa dataset. RMDSRN completely fails to extract the isosurface with a coherent shape, while the isosurfaces from Switch-NeRF and Neural Experts capture fewer

Table 7: Comparison of average PSNR, LPIPS, total CT in hours, and total inference time (IT) in minutes under the same CR. The best result is highlighted in bold.

dataset	CR	method	PSNR (dB) \uparrow	LPIPS \downarrow	CT \downarrow	IT \downarrow	dataset	CR	method	PSNR (dB) \uparrow	LPIPS \downarrow	CT \downarrow	IT \downarrow
argon bubble	3,470	CoordNet	46.90	0.016	91.63	30.05	combustion (MF)	1,951	CoordNet	47.32	0.078	61.78	18.33
		NeurComp	43.12	0.012	105.67	32.16			NeurComp	51.77	0.030	74.03	20.04
		SIREN	41.83	0.030	78.23	28.75			SIREN	42.76	0.126	53.12	17.56
		ECNR	34.68	0.107	22.24	4.05			ECNR	30.79	0.271	30.07	2.64
		fV-SRN	35.11	0.095	3.83	3.84			fV-SRN	34.55	0.257	2.51	2.43
		APMGSRN	30.75	0.200	6.82	3.95			APMGSRN	30.17	0.285	6.45	2.47
		RMDSRN	35.32	0.093	4.02	3.87			RMDSRN	34.21	0.260	2.76	2.44
		Switch-NeRF	40.40	0.042	40.83	24.49			Switch-NeRF	38.34	0.162	27.06	12.76
		Neural Experts	39.38	0.044	60.50	25.34			Neural Experts	37.02	0.180	43.99	12.69
		MoE-INR	48.97	0.012	42.67	24.85			MoE-INR	54.26	0.030	29.09	13.78
ionization (H+)	2,862	CoordNet	55.29	0.014	50.89	17.56	vortex	2,510	CoordNet	40.97	0.057	14.12	6.34
		NeurComp	49.00	0.017	64.56	19.24			NeurComp	46.80	0.024	16.82	6.78
		SIREN	50.87	0.024	44.56	16.78			SIREN	38.63	0.082	10.83	5.89
		ECNR	33.54	0.201	15.99	2.11			ECNR	23.23	0.440	10.99	1.05
		fV-SRN	40.06	0.123	1.43	1.96			fV-SRN	30.07	0.288	0.57	0.86
		APMGSRN	33.27	0.195	6.02	2.01			APMGSRN	25.93	0.344	0.68	0.93
		RMDSRN	40.20	0.120	1.50	1.97			RMDSRN	29.37	0.292	0.61	0.89
		Switch-NeRF	43.26	0.092	24.02	12.48			Switch-NeRF	33.67	0.201	8.45	4.46
		Neural Experts	44.88	0.061	38.56	13.65			Neural Experts	31.25	0.196	14.04	4.92
		MoE-INR	60.24	0.007	24.22	13.11			MoE-INR	52.37	0.011	8.96	4.68

Table 8: Comparison of average CD values under the same CR. The best result is highlighted in bold.

dataset	CR	method	isovalues			dataset	CR	method	isovalues		
argon bubble	3,470		$v = -0.82$	$v = -0.68$	$v = -0.45$	combustion (MF)	1,951		$v = -0.15$	$v = 0$	$v = 0.25$
		CoordNet	0.53	0.55	0.72			CoordNet	0.59	0.61	0.61
		NeurComp	0.56	0.53	0.76			NeurComp	0.36	0.38	0.38
		SIREN	1.05	0.98	1.18			SIREN	0.91	0.94	0.96
		ECNR	4.07	3.09	4.09			ECNR	3.84	3.88	3.97
		fV-SRN	3.46	2.86	3.71			fV-SRN	2.51	2.52	2.59
		APMGSRN	30.61	14.73	19.30			APMGSRN	4.40	4.46	4.67
		RMDSRN	3.38	2.73	3.59			RMDSRN	2.62	2.65	2.74
		Switch-NeRF	1.19	1.22	1.48			Switch-NeRF	1.57	1.62	1.64
		Neural Experts	1.53	1.53	1.95			Neural Experts	1.89	1.92	1.96
ionization (H+)	2,862	MoE-INR	0.41	0.44	0.62	vortex	2,510	MoE-INR	0.30	0.31	0.31
			$v = -0.98$	$v = -0.75$	$v = -0.4$				$v = -0.32$	$v = -0.05$	$v = 0.17$
		CoordNet	1.23	0.21	0.13			CoordNet	0.74	0.72	0.73
		NeurComp	2.24	0.36	0.21			NeurComp	0.53	0.55	0.61
		SIREN	1.79	0.31	0.18			SIREN	0.93	0.87	0.88
		ECNR	11.14	2.89	1.67			ECNR	6.97	9.87	15.12
		fV-SRN	6.96	1.41	0.75			fV-SRN	2.55	2.46	2.72
		APMGSRN	17.27	3.34	1.97			APMGSRN	4.80	5.83	8.97
		RMDSRN	6.49	1.39	0.71			RMDSRN	2.75	2.64	2.92
		Switch-NeRF	2.58	0.71	0.52			Switch-NeRF	1.70	1.59	1.71
		Neural Experts	3.43	0.72	0.43			Neural Experts	2.30	2.18	2.40
		MoE-INR	0.76	0.13	0.08			MoE-INR	0.27	0.26	0.26

details compared to those produced by MoE-INR.

Quantitative Analysis. Table 7 reports the average PSNR and LPIPS values. In terms of PSNR, MoE-INR demonstrates superior performance compared to all baselines. For image-level evaluation, MoE-INR achieves the best quality in most cases, except for the argon bubble and combustion (MF) datasets, where MoE-INR and NeurComp achieve the same performance. Additionally, the total CT and inference time for each approach is listed in Table 7. While RMDSRN, APMGSRN, and fV-SRN require the shortest time for compression, they exhibit the worst performance in terms of PSNR and LPIPS. Following Xiong et al. [39], fV-SRN is used as the backbone model in RMDSRN, which explains why RMDSRN also requires a short time to model a field. Conversely, ECNR requires significant training time due to its multiscale design, which is optimized sequentially from the coarsest scale to the finest scale. Compared to conventional INRs, NeurComp requires more time for optimization, as it computes an additional gradient loss and performs an extra step for weight quantization. Table 8 reports the average CD values under three different isovalues. Evidently, MoE-INR outperforms all state-of-the-art INRs across all cases, demonstrating its superior capability in preserving geometric information. Refer to Section 5.5 for the evaluation under different CRs.

5.3 Uncertainty Estimation

Like RMDSRN, MoE-INR can also estimate uncertainty in the predicted voxel values. Specifically, MoE-INR utilizes more than two experts to process a single input coordinate for voxel prediction and is optimized by incorporating an additional variance regularization loss, following Xiong et al. [39]. The uncertainty for each voxel can be obtained by calculating the variance across these predictions. Figure 9 compares the renderings of error, variance, and reconstructed volumes as approximated by MoE-INR and RMDSRN under the same CR. For

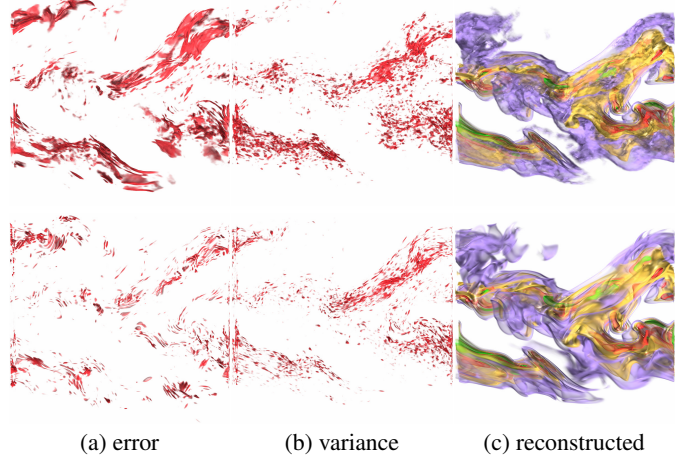


Fig. 9: Rendering of (a) error, (b) variance, and (c) reconstructed volumes of RMDSRN (top) and MoE-INR (bottom) using the combustion (HR) dataset. The CR is 1,951.

the reconstructed volume, MoE-INR demonstrates superior quality, as RMDSRN distorts the features in the rendering image. In terms of uncertainty, both MoE-INR and RMDSRN exhibit a correlation between the variance and error volumes. Additionally, we leverage Pearson correlation (PC) and Jaccard index with spatial tolerance (JI-ST) [39] to quantify the relationship between variance and error. The quantitative scores are shown in Table 10. The results show that under the same CR, MoE-INR can achieve better PC and JI-ST values compared to RMDSRN. These results emphasize that MoE-INR not only achieves better

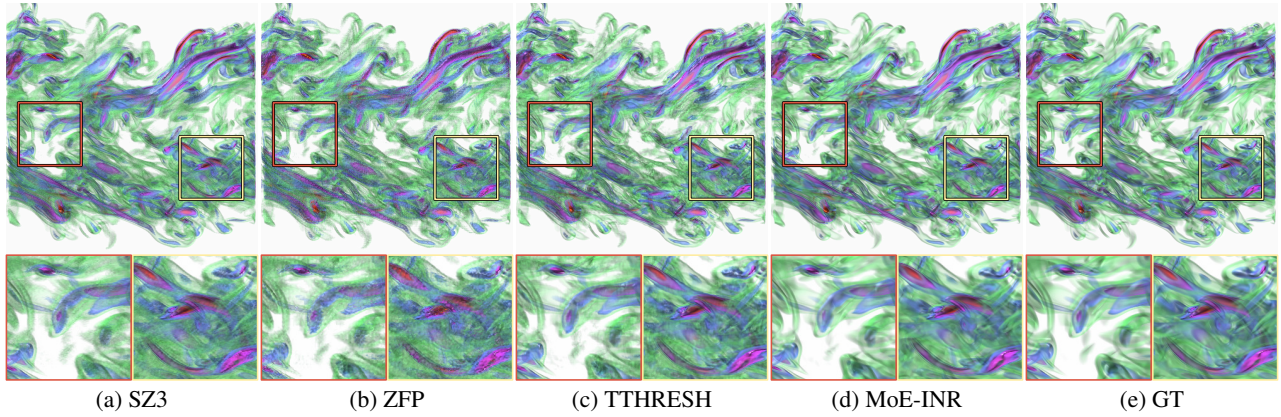


Fig. 10: Comparison of volume rendering between MoE-INR and traditional lossy compressors using the combustion (VORT) dataset. The average PSNR is 39.73 (dB).

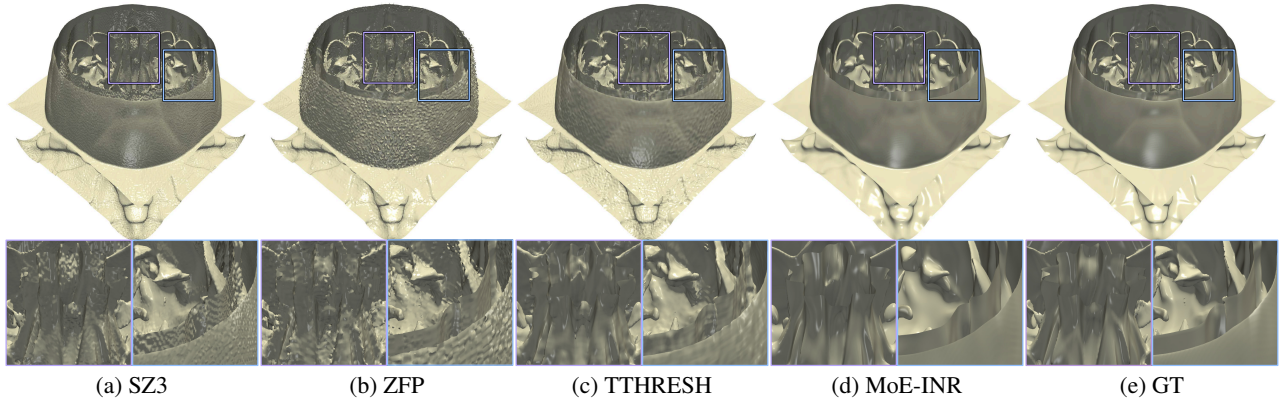


Fig. 11: Comparison of volume rendering between MoE-INR and traditional lossy compressors using the ionization (PD) dataset. The average PSNR is 50.73. The chosen isovalue is -0.83 .

Table 9: Average CR, LPIPS, and total CT in hours under the same PSNR. The best result is highlighted in bold.

dataset	PSNR (dB) \uparrow	method	CR \uparrow	LPIPS \downarrow	CT (hours) \downarrow
combustion (HR)	42.77	SZ3	38	0.163	0.02
		ZFP	64	0.229	0.06
		TTHRESH	1,465	0.177	0.22
		MoE-INR	10,991	0.075	24.22
combustion (VORT)	39.73	SZ3	47	0.116	0.02
		ZFP	64	0.224	0.05
		TTHRESH	1,328	0.151	0.21
		MoE-INR	5,053	0.075	26.28
ionization (PD)	50.73	SZ3	32	0.092	0.02
		ZFP	34	0.123	0.01
		TTHRESH	1,562	0.109	0.17
		MoE-INR	24,878	0.073	21.25
Tangaroa	44.41	SZ3	101	0.082	0.04
		ZFP	79	0.138	0.06
		TTHRESH	4,005	0.071	0.36
		MoE-INR	50,898	0.057	43.64

Table 10: Comparison of average PC and JI-ST for uncertainty quantification using the combustion (HR) dataset. The CR is 1,951. The best result is highlighted in bold.

method	PC \uparrow	JI-ST \uparrow
RMDSRN	0.539	0.545
MoE-INR	0.589	0.583

reconstruction performance but also maintains a comparable correlation between uncertainty and error when compared to RMDSRN.

5.4 Comparison against Traditional Lossy Compressors

We evaluate the performance of MoE-INR by comparing it to three traditional lossy compressors, i.e., SZ3, ZFP, and TTHRESH, under the same PSNR. Figure 10 shows volume rendering images for the combustion (VORT) dataset. While all compressors preserve the overall texture,

the images produced by the lossy compressors exhibit noticeable artifacts and noise. Similarly, Figure 11 presents isosurface rendering images for the ionization (PD) dataset. Although all methods maintain geometric shapes, the lossy compressors generate visibly non-smooth surfaces. Table 9 summarizes the CR, LPIPS, and total CT in hours. MoE-INR significantly outperforms the lossy compressors in terms of CR and LPIPS. However, similar to other INR approaches, MoE-INR incurs higher computational costs, resulting in slower runtimes than the lossy compressors due to its iterative learning process for compression. For the comparison between MoE-INR and TTHRESH under different CRs, refer to Section 5.5.

5.5 Performance Evaluation under Various CRs

To evaluate the performance of different compressors under various CRs, we compare PSNR and LPIPS values of CoordNet, NeurComp, SIREN, fV-SRN, APMGSRN, RMDSRN, Switch-NeRF, Neural Experts, TTHRESH, and MoE-INR across five CRs. We exclude ECNR as it cannot achieve large CRs due to its multiscale design. Specifically, ECNR learns data representations from coarse to fine, and at each scale, multiple INRs are utilized to model different spatiotemporal blocks. Similarly, ZFP and SZ3 are excluded due to their limited compression capabilities. Figure 12 illustrates the performance trends of these compressors as the CR increases from thousands to tens of thousands. Clearly, MoE-INR achieves the best performance in both PSNR and LPIPS across all CRs. Additionally, Figure 13 compares volume rendering results using the ionization (H2) dataset under a CR of 50,000. TTHRESH and grid-based INRs totally fail to reconstruct the volume with meaningful structure. NeurComp can only retain the bottom part of the ionization (H2), while the patterns at the top of the ionization (H2) are completely lost. CoordNet, SIREN, Switch-NeRF, and Neural Experts preserve only some patterns at the top of ionization (H2). Only MoE-INR can keep features at different regions well under

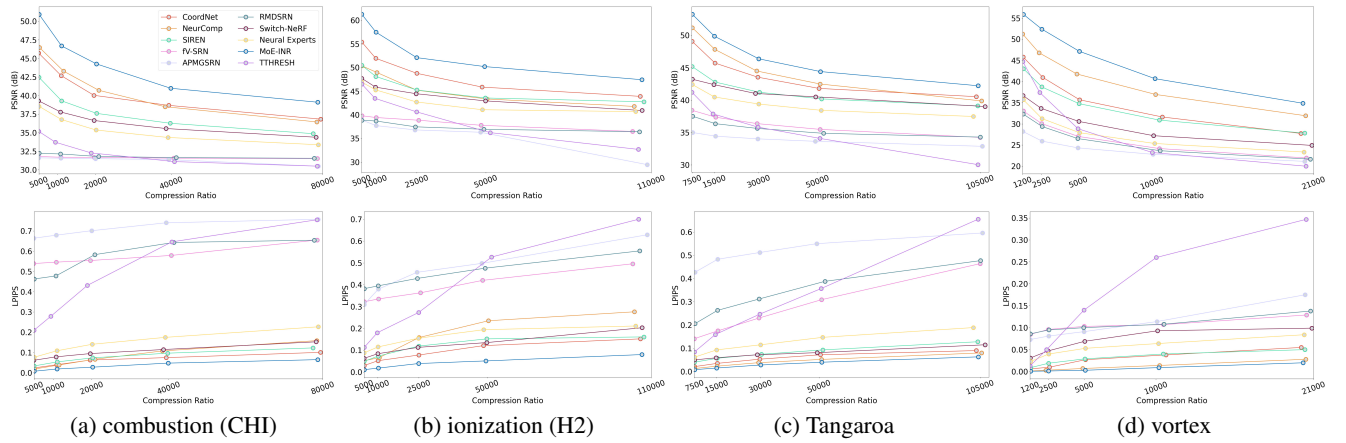


Fig. 12: Comparison of average PSNR (top) and LPIPS (bottom) values under various CRs among different compressors.

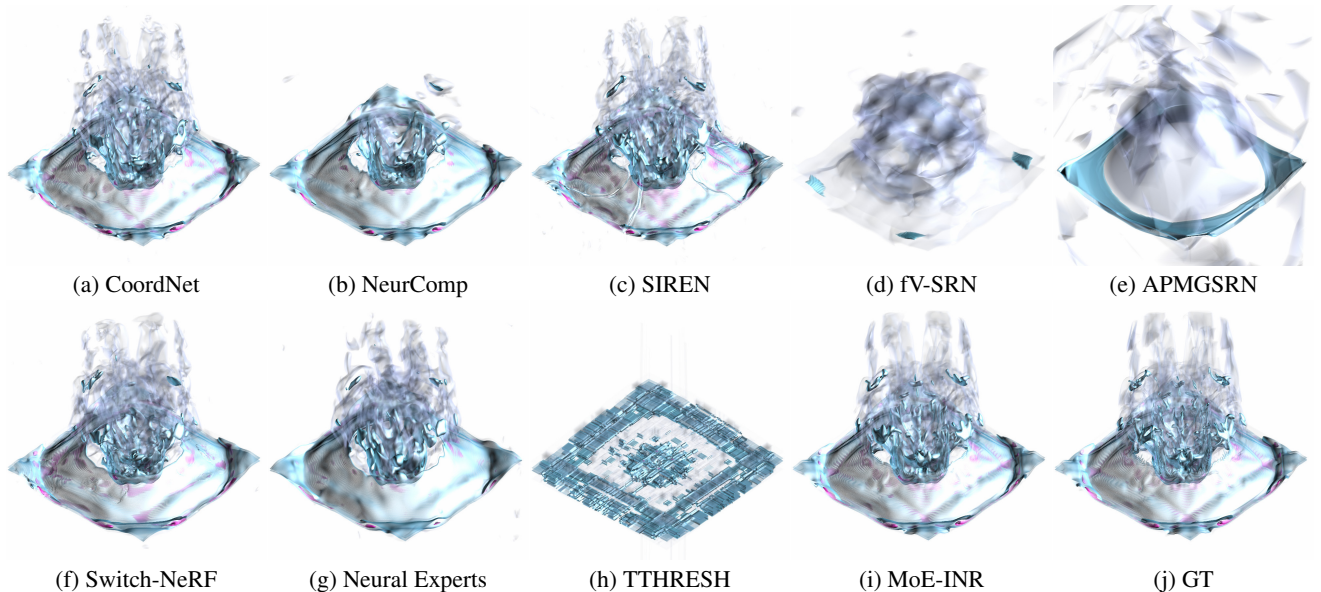


Fig. 13: Comparison of volume rendering among different compressors using the ionization (H2) dataset. The CR is around 50,000.

a large CR. Refer to the Appendix for additional visual comparisons.

6 LIMITATION AND FUTURE WORK

We acknowledge two limitations of MoE-INR and discuss future directions. First, as a MoE architecture, MoE-INR requires optimizing multiple expert decoders during each epoch. Compared to non-MoE INRs, such as APMGSRN and fV-SRN, this results in longer training time. Such a limitation may restrict the application of MoE-INR in certain data compression scenarios, such as online compression. In the future, we aim to address this challenge by designing advanced training techniques, such as data and model parallelism, to significantly reduce compression time. Second, although MoE-INR outperforms state-of-the-art compressors, its current expert assignment strategy and the number of experts are static. This fixed allocation does not account for the varying prediction difficulty across voxels, as each input is always processed by a predetermined number of experts. As a future direction, we aim to explore dynamic MoE architectures in which a policy network can adaptively select the number of experts to activate based on the input coordinates. Additionally, the number of experts could be adjusted during optimization to better reflect the underlying data complexity. In this work, we leverage MoE to unify various INR architectures for time-varying volumetric data compression, providing a flexible framework that supports both regular and irregular field decomposition. This approach enables more effective grouping of patterns into clusters, allowing expert decoders to focus and specialize on

distinct data regions.

7 CONCLUSIONS

We have presented MoE-INR, a unified, flexible, and effective INR framework that leverages a novel MoE architecture to automatically partition spatiotemporal fields and assign each partition to specialized experts. By incorporating a policy network, a shared encoder, and expert decoders, MoE-INR can efficiently handle regular and irregular subdivisions, avoid boundary artifacts across partition edges, and independently learn complex patterns with high fidelity. The pre-training scheme for the policy network enhances generalization, enabling compatibility with diverse data decomposition algorithms. Quantitative and qualitative evaluations demonstrate that MoE-INR outperforms state-of-the-art non-MoE INRs, MoE INRs, and lossy compression methods across diverse time-varying volumetric datasets under varying CRs. Furthermore, MoE-INR can also function as an error-aware INR by (1) enabling multiple decoders to process a single input coordinate and (2) calculating the variance across multiple predictions.

ACKNOWLEDGEMENTS

This research was supported in part by the National Natural Science Foundation of China under Grant No. 62302422 and CORE, a joint research center for ocean research between Laoshan Laboratory and The Hong Kong University of Science and Technology. The authors thank the anonymous reviewers for their insightful comments.

REFERENCES

- [1] M. Ainsworth, O. Tugluk, B. Whitney, and S. Klasky. Multilevel techniques for compression and reduction of scientific data-quantitative control of accuracy in derived quantities. *SIAM Journal on Scientific Computing*, 41(4):A2146–A2171, 2019. doi: 10.1137/18M1208885 2
- [2] M. Atzmon and Y. Lipman. SAL: Sign agnostic learning of shapes from raw data. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2565–2574, 2020. doi: 10.48550/arXiv.1911.10414 3
- [3] R. Ballester-Ripoll, P. Lindstrom, and R. Pajarola. TTHRESH: Tensor compression for multidimensional visual data. *IEEE Transactions on Visualization and Computer Graphics*, 26(9):2891–2903, 2019. doi: 10.1109/TVCG.2019.2904063 2, 5
- [4] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *Proceedings of International Joint Conference on Artificial Intelligence*, pp. 659–663, 1977. doi: 10.5555/1622943.1622971 4
- [5] Y. Ben-Shabat, C. H. Koneputugodage, S. Ramasinghe, and S. Gould. Neural Experts: Mixture of experts for implicit neural representations. In *Proceedings of Advances in Neural Information Processing Systems*, 2024. doi: 10.48550/arXiv.2410.21643 2, 5
- [6] B. Cao, Y. Sun, P. Zhu, and Q. Hu. Multi-modal gated mixture of local-to-global experts for dynamic image fusion. In *Proceedings of IEEE International Conference on Computer Vision*, pp. 23555–23564, 2023. doi: 10.48550/arXiv.2302.01392 2
- [7] S. Devkota and S. Pattanaik. Efficient neural representation of volumetric data using coordinate-based networks. *Computer Graphics Forum*, 42(7):e14955, 2023. doi: 10.1111/cgf.14955 1, 2
- [8] W. Fedus, B. Zoph, and N. Shazeer. Switch Transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022. doi: 10.48550/arXiv.2101.03961 2
- [9] J. Han and C. Wang. SSR-TVD: Spatial super-resolution for time-varying data analysis and visualization. *IEEE Transactions on Visualization and Computer Graphics*, 28(6):2445–2456, 2022. doi: 10.1109/TVCG.2020.3032123 4
- [10] 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, 2, 4, 5
- [11] J. Han and F. Yang. DCINR: A divide-and-conquer implicit neural representation for compressing time-varying volumetric data in hours. *IEEE Transactions on Visualization and Computer Graphics*, 2025. accepted. doi: 10.1109/TVCG.2025.3564255 1, 4
- [12] 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, 2, 4
- [13] J. Han, H. Zheng, and J. Tao. A study of data augmentation for learning-driven scientific visualization. *IEEE Transactions on Visualization and Computer Graphics*, 2025. accepted. doi: 10.1109/TVCG.2025.3587685 4
- [14] J. Han, H. Zheng, Y. Xing, D. Z. Chen, and C. Wang. V2V: A deep learning approach to variable-to-variable selection and translation for multivariate time-varying data. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1290–1300, 2021. doi: 10.1109/TVCG.2020.3030346 4
- [15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90 3
- [16] A. Q. Jiang, A. Sablayrolles, A. Roux, A. Mensch, B. Savary, C. Bamford, D. S. Chaplot, D. d. l. Casas, E. B. Hanna, F. Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024. doi: 10.48550/arXiv.2401.04088 1
- [17] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proceedings of International Conference for Learning Representations*, 2015. doi: 10.48550/arXiv.1412.6980 4
- [18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of Advances in Neural Information Processing Systems*, pp. 84–90, 2012. doi: 10.1145/3065386 5
- [19] D. Lepikhin, H. Lee, Y. Xu, D. Chen, O. Firat, Y. Huang, M. Krikun, N. Shazeer, and Z. Chen. GShard: Scaling giant models with conditional computation and automatic sharding. In *Proceedings of International Conference on Learning Representations*, 2021. doi: 10.48550/arXiv.2006.16668 4
- [20] X. Liang, K. Zhao, S. Di, S. Li, R. Underwood, A. M. Gok, J. Tian, J. Deng, J. C. Calhoun, D. Tao, Z. Chen, and F. Cappello. SZ3: A modular framework for composing prediction-based error-bounded lossy compressors. *IEEE Transactions on Big Data*, 9(2):485–498, 2023. doi: 10.1109/TBDATA.2022.3201176 2, 5
- [21] P. Lindstrom. Fixed-rate compressed floating-point arrays. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2674–2683, 2014. doi: 10.1109/TVCG.2014.2346458 2, 5
- [22] 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, 2, 4, 5
- [23] Z. Mi and D. Xu. Switch-NeRF: Learning scene decomposition with mixture of experts for large-scale neural radiance fields. In *Proceedings of International Conference on Learning Representations*, 2023. 2, 4, 5
- [24] 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. In *Proceedings of European Conference on Computer Vision*, pp. 405–421, 2020. doi: 10.48550/arXiv.2003.08934 3
- [25] B. Mustafa, C. Riquelme, J. Puigcerver, R. Jenatton, and N. Houlsby. Multimodal contrastive learning with LIMoE: the language-image mixture of experts. In *Proceedings of Advances in Neural Information Processing Systems*, vol. 35, pp. 9564–9576, 2022. doi: 10.48550/arXiv.2206.02770 2
- [26] S. Nowlan and G. E. Hinton. Evaluation of adaptive mixtures of competing experts. In *Proceedings of Advances in Neural Information Processing Systems*, 1990. doi: 10.1007/s10462-009-9124-7 2
- [27] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. In *Proceedings of Advances in Neural Information Processing Systems*, pp. 8024–8035, 2019. doi: 10.48550/arXiv.1912.01703 4
- [28] 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 4
- [29] 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 2, 3, 4, 5
- [30] M. Soler, M. Plainchault, B. Conche, and J. Tierny. Topologically controlled lossy compression. In *Proceedings of IEEE Pacific Visualization Symposium*, pp. 46–55, 2018. doi: 10.1109/PacificVis.2018.00015 2
- [31] M. Tancik, P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. Barron, and R. Ng. Fourier features let networks learn high frequency functions in low dimensional domains. pp. 7537–7547, 2020. doi: 10.48550/arXiv.2006.10739 3
- [32] 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, 2, 4, 5
- [33] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. doi: 10.48550/arXiv.2302.13971 1
- [34] 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 2
- [35] 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, 2, 4, 5
- [36] 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 4
- [37] Q. Wu, J. A. Insley, V. A. Mateevitsi, S. Rizzi, M. E. Papka, and K.-L. Ma. Distributed neural representation for reactive in situ visualization. *IEEE Transactions on Visualization and Computer Graphics*, 2024. accepted. doi: 10.1109/TVCG.2024.3432710 1, 2
- [38] 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, 2023. doi: [10.1109/TVCG.2023.3327194](https://doi.org/10.1109/TVCG.2023.3327194) 1, 2, 4, 5
- [39] 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](https://doi.org/10.1109/TVCG.2024.3456357) 1, 2, 4, 5, 7
- [40] L. Yan, X. Liang, H. Guo, and B. Wang. TopoSZ: Preserving topology in error-bounded lossy compression. *IEEE Transactions on Visualization and Computer Graphics*, 30(1):1302–1312, 2023. doi: [10.1109/TVCG.2023.3326920](https://doi.org/10.1109/TVCG.2023.3326920) 2
- [41] J. Yu, Y. Zhuge, L. Zhang, P. Hu, D. Wang, H. Lu, and Y. He. Boosting continual learning of vision-language models via mixture-of-experts adapters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 23219–23230, 2024. doi: [10.48550/arXiv.2403.11549](https://doi.org/10.48550/arXiv.2403.11549) 2
- [42] 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*, pp. 586–595, 2018. doi: [10.1109/CVPR.2018.00068](https://doi.org/10.1109/CVPR.2018.00068) 4
- [43] J. Zhao, C.-C. Tseng, M. Lu, R. An, X. Wei, H. Sun, and S. Zhang. MoEC: Mixture of experts implicit neural compression. *arXiv preprint arXiv:2312.01361*, 2023. doi: [10.48550/arXiv.2312.01361](https://doi.org/10.48550/arXiv.2312.01361) 2
- [44] J. Zheng, S. Ramasinghe, and S. Lucey. Rethinking positional encoding. *arXiv preprint arXiv:2107.02561*, 2021. doi: [10.48550/arXiv.2107.02561](https://doi.org/10.48550/arXiv.2107.02561) 3

APPENDIX

1 MODEL CONFIGURATIONS

Table 1 reports each dataset’s model configurations and total CT under different CRs.

Table 1: Model configurations and total CT under different CRs, where M is the initial number of neurons in MoE-INR.

dataset	M	CR	CT (hours)
argon bubble	207	3,470	42.67
	234	1,951	29.09
	145	5,053	26.28
combustion	98	10,991	24.23
	72	20,131	23.56
	51	39,127	23.17
	36	78,858	22.96
	182	2,862	24.22
ionization	137	5,033	23.25
	96	10,078	21.59
	61	24,878	21.25
	43	48,741	20.96
	29	106,630	20.83
Tangaroa	163	7,510	49.79
	115	15,011	46.86
	81	30,041	44.64
	62	50,898	43.64
	43	104,370	43.46
vortex	176	1,252	9.57
	124	2,510	8.96
	87	5,065	8.72
	61	10,205	8.59
	43	20,267	8.27

2 VISUALIZATION OF EXPERT ASSIGNMENTS

Figure 1 shows the expert assignment statistics for different pre-training schemes. Without pre-training, the policy network only favors a single expert, which becomes a conventional INR architecture. When pre-training is leveraged, most partitions can enable the policy network to assign experts in balance. Since k-means clustering produces uneven allocation to different clusters, an imbalanced expert assignment occurs when voxel clustering is applied in pre-training.

Figure 2 illustrates the voxel assignment for one expert using the ionization (T) dataset after pre-training and training. Without pre-training, MoE-INR assigns all coordinates to a single expert for processing all coordinates. The remaining experts are inactive, leading to no assignment, as shown in Figure 2 (a). Neither random partitioning (Figure 2 (b)) nor load balancing (Figure 2 (d)) successfully divides the field meaningfully after pre-training. Thus, during training, the policy network fails to fine-tune the subdivision and only assigns all coordinates to one expert. However, pre-training MoE-INR with voxel clustering (Figure 2 (c)) yields a semantic partition, leading to the best compression performance among all investigated subdivisions. In addition, after voxel clustering pre-training, MoE-INR partitions the field in a finer structure. For example, the bottom feature, distinguished from the top feature, is assigned to another expert.

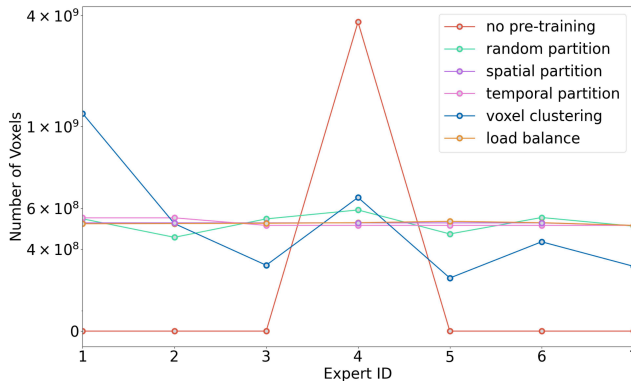
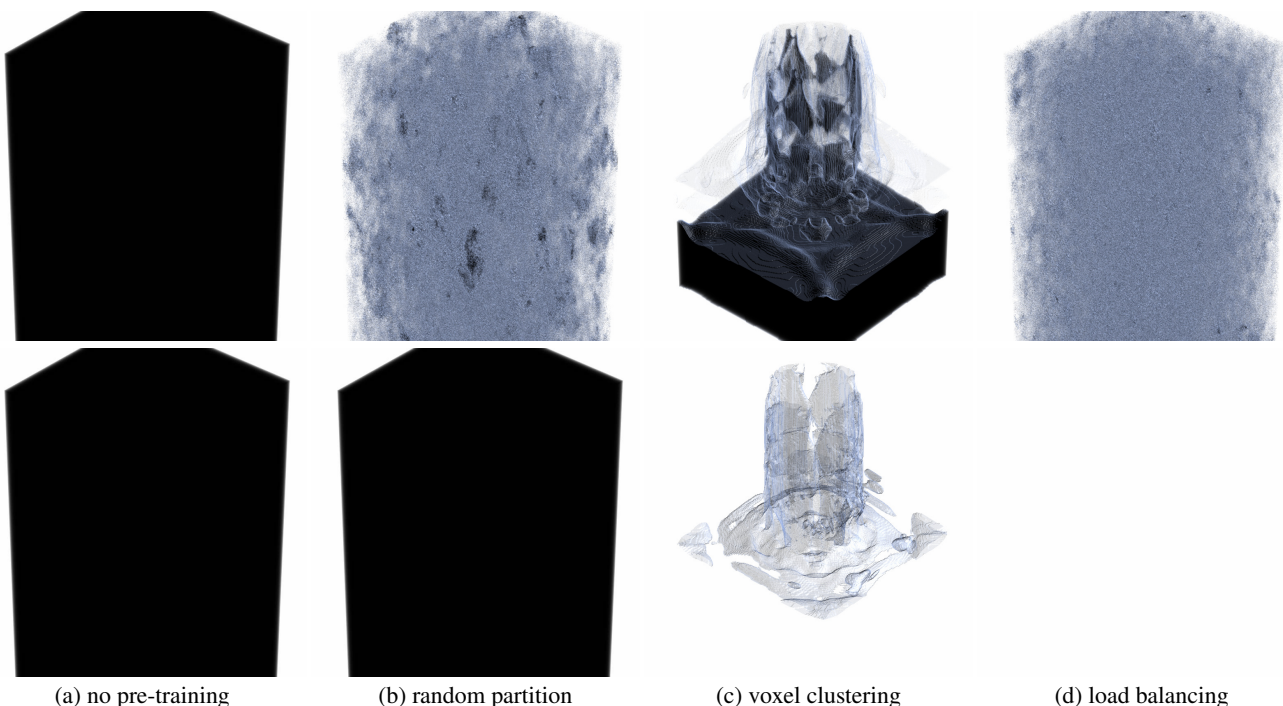


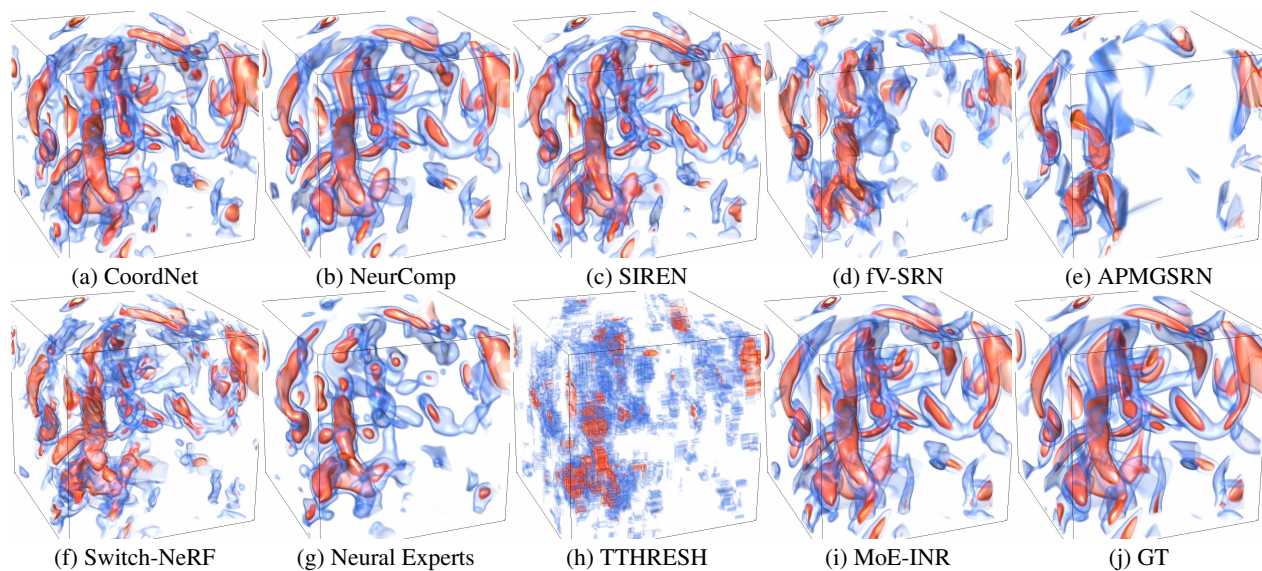
Fig. 1: Expert assignment statistics under different pre-training schemes using the ionization (T) dataset.

3 PERFORMANCE EVALUATION UNDER DIFFERENT CRs

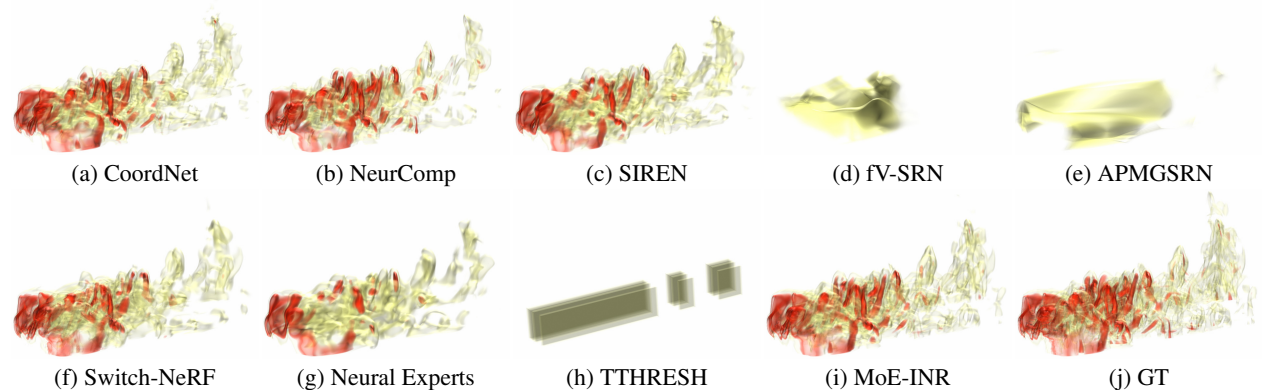
Figures 3 and 4 show the volume rendering results among different compressors using the vortex and Tangaroa datasets, respectively. In addition, Figures 5 and 6 present the isosurface rendering results among different compressors using the combustion (CHI) and ionization (H2) datasets, respectively. Since APMGSRN, fV-SRN, and TTHRESH are unable to reconstruct the combustion (CHI) dataset at a CR of 78,858, failing to extract isosurfaces from the reconstructed volumes on the specific isovalue. Similarly, for the ionization (H2) dataset, TTHRESH cannot extract isosurfaces from the decompressed volume. All those visual results further demonstrate that MoE-INR can achieve superior quality compared with state-of-the-art compressors.



(a) no pre-training (b) random partition (c) voxel clustering (d) load balancing
 Fig. 2: Visualization of the learned assignments for the 4th expert using the ionization (T) dataset. Top: after pre-training, and bottom: after training.



(a) CoordNet (b) NeurComp (c) SIREN (d) fV-SRN (e) APMGSRN
 (f) Switch-NeRF (g) Neural Experts (h) TTHRESH (i) MoE-INR (j) GT
 Fig. 3: Comparison of volume rendering among different compressors using the vortex dataset. The CR is around 20,267.



(a) CoordNet (b) NeurComp (c) SIREN (d) fV-SRN (e) APMGSRN
 (f) Switch-NeRF (g) Neural Experts (h) TTHRESH (i) MoE-INR (j) GT
 Fig. 4: Comparison of volume rendering among different compressors using the Tangaroa dataset. The CR is around 104,370.

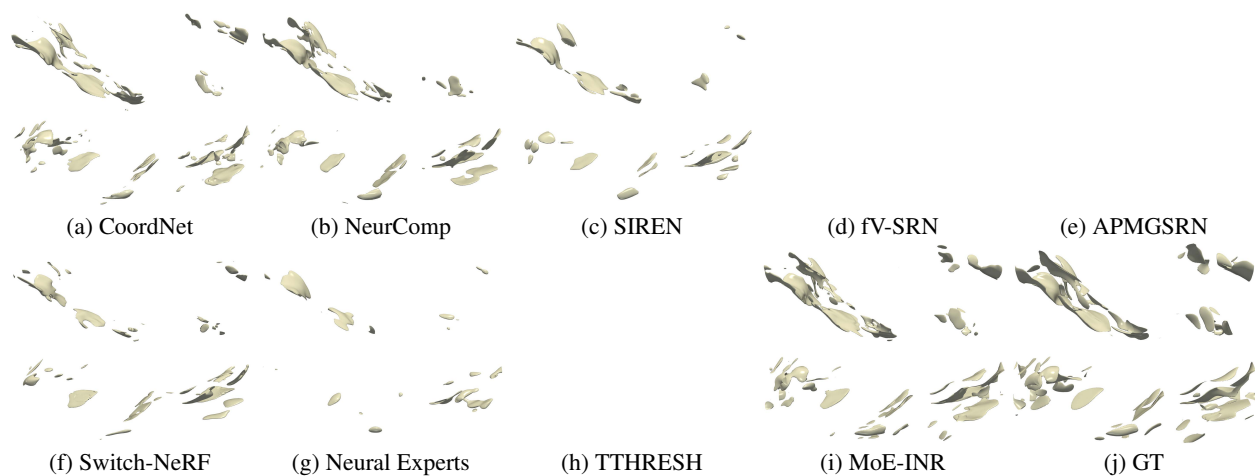


Fig. 5: Comparison of isosurface rendering among different compressors using the combustion (CHI) dataset. The CR is around 78,858. The chosen isovalue is -0.7 .

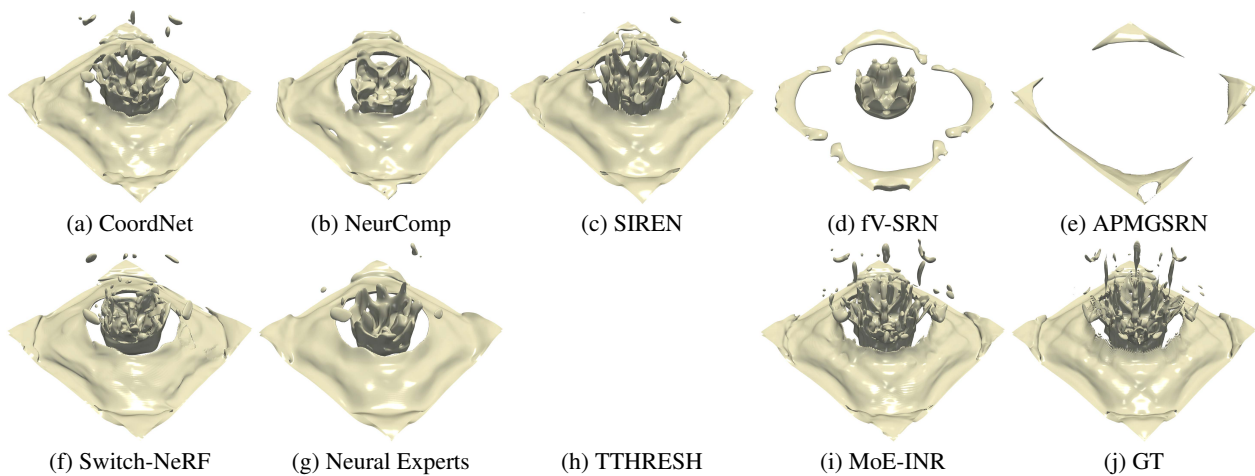


Fig. 6: Comparison of isosurface rendering among different compressors using the ionization (H2) dataset. The CR is around 106,630. The chosen isovalue is -0.86 .