Lossless-INR: Lossless Volumetric Implicit Neural Representations

Kaiyuan Tang* Daniel Burke† Chaoli Wang‡

University of Notre Dame

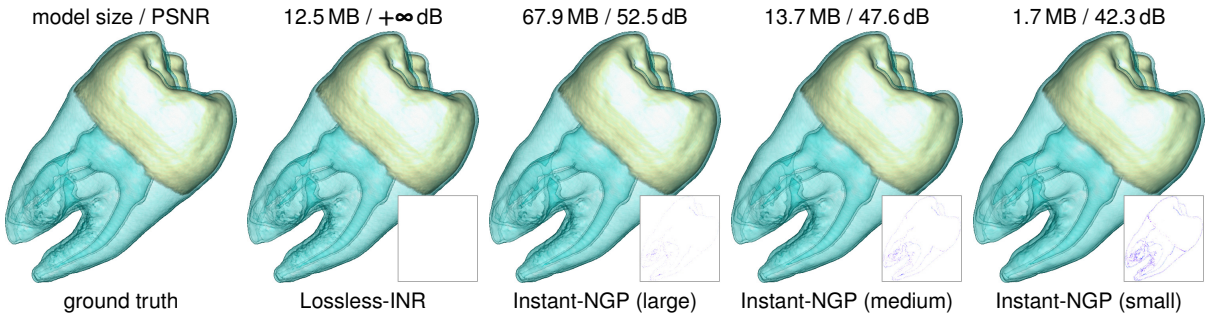


Figure 1: Comparison of the rendering results on the tooth dataset using Instant-NGP [15] under different model sizes and our Lossless-INR. The bottom-right difference image shows pixel-wise error relative to the ground truth in the CIELUV color space. In this paper, we explore and design Lossless-INR, which can represent volumetric data without error while maintaining a relatively compact model size.

Abstract

Implicit neural representation (INR) methods provide continuous coordinate-to-value mappings and integrate naturally with direct volume rendering, making them attractive for representing volumetric data. However, existing INR-based approaches for volumetric data are inherently lossy, and even small reconstruction errors can propagate through rendering and downstream analysis. In this work, we explore Lossless-INR, a lossless INR framework for 3D scientific volumetric data based on bit-plane decomposition. By decomposing each voxel value into binary bit-planes, we reformulate reconstruction as per-bit binary classification, so that exact recovery reduces to predicting every bit correctly. To make this optimization tractable while keeping the representation compact, we combine an octree block-partitioning strategy that adaptively subdivides complex regions with a ternary feature-grid network whose grid entries are parameterized by a ternary set of values. Experiments on diverse volumetric datasets show that this design can achieve zero bit-error rate and bit-exact reconstruction, enabling faithful rendering and downstream analysis with a compact representation. The code is available at https://github.com/TouKaienn/Lossless-INR.

Keywords: Volume visualization; implicit neural representation; lossless volumetric representation

1 Introduction

Implicit neural representation (INR) models volumetric data as continuous coordinate-to-value functions parameterized by neural networks, supporting random-access queries and integrating naturally into direct volume rendering (DVR) pipelines. Because of their simple architecture and flexibility, INRs have become a common choice for volume visualization, with applications spanning compression [4, 6, 13, 20, 30], rendering [26–29], data generation [5, 21], scene representation [22, 31, 32], and ensemble exploration [2].

*e-mail: ktang2@nd.edu
†e-mail: dburke6@nd.edu
‡e-mail: chaoli.wang@nd.edu

Despite their effectiveness, existing INR-based methods for volumetric data are inherently *lossy* [6, 13, 20, 26]. Even small reconstruction errors can propagate through multiple stages of the DVR pipeline, from transfer function lookups to gradient estimation for shading, leading to incorrect colors and lighting during rendering. These errors are also problematic for analysis tasks such as computing derived fields, extracting topology, or comparing simulation ensembles, where small reconstruction errors can significantly bias analysis results. A straightforward solution is to increase model capacity by adding more optimizable parameters to reduce reconstruction error. However, as shown in Figure 1, even a substantial increase in parameter budget does not eliminate the error, leaving a persistent gap from the original data.

To address this limitation, we design Lossless-INR, a lossless INR for 3D scientific volumetric data via *bit-plane decomposition* [8, 9, 16]. Instead of directly regressing continuous voxel values, we decompose each B-bit voxel value into B binary bit-planes and reformulate reconstruction as per-bit binary classification, with network outputs quantized to $\{0,1\}$. If every bit of every voxel is predicted correctly, the original voxel values can be recovered exactly by recomposing the predicted bit-planes, yielding a bit-exact lossless representation. Applying this idea to 3D volumes, however, is non-trivial because driving the bit error to zero still requires substantial model capacity. To keep the representation compact while making the learning problem tractable, we combine an *octree block partitioning* strategy that adaptively divides complex volumes into smaller local blocks with a *ternary feature grid network* whose grid entries are parameterized with a ternary set. This design turns the full reconstruction problem into a collection of manageable local fitting tasks while preserving a compact and lossless overall representation.

The contributions of this paper are summarized as follows: First, to our knowledge, we present the first INR framework that achieves lossless representation of volumetric data. Second, we extend bit-plane decomposition to 3D volumes and combine it with an octree block partitioning strategy and a ternary feature grid network to obtain a storage-efficient lossless representation. Third, through experiments on diverse volumetric datasets, we show that our Lossless-INR achieves zero bit-error rate at every voxel, enabling bit-exact reconstruction and rendering.

2 Related Work

Deep learning for volume representation. As a core DL4SciVis

research task [25], INRs encode a volumetric scalar field as a continuous function learned by a neural network, enabling compressed storage and random-access evaluation. Lu et al. [13] framed volume compression as function approximation, showing that a coordinate-based multilayer perceptron (MLP) with quantized weights can outperform classical codecs while supporting time-varying fields and gradient preservation. Weiss et al. [26] redesigned the network to exploit GPU tensor cores, integrating reconstruction directly into on-chip raytracing kernels for significantly faster decoding and compression-domain volume rendering. Tang and Wang [20] proposed ECNR, which replaces a single large MLP with a Laplacian pyramid of small, parallelized MLPs that fit local blocks, substantially accelerating training and inference for time-varying datasets. Han et al. [6] introduced KD-INR, applying knowledge distillation to transfer temporal redundancy across timesteps for more efficient encoding. Tang and Wang [21] further presented STSR-INR, a method for simultaneous spatiotemporal super-resolution of multivariate volumes via learnable variable embeddings and latent-space interpolation. Yang et al. [30] proposed Meta-INR, a meta-learning pretraining strategy that learns generalizable initial parameters from partial observations, enabling rapid adaptation to unseen volumes. Building on Meta-INR, Son et al. [18] further developed MC-INR to encode multivariate scientific data efficiently. Wurster et al. [29] developed APMGSRN, which adaptively places multi-resolution feature grids for large-scale data visualization. Wu et al. [27] combined multi-resolution hash encoding with neural representations for interactive volume visualization. These hybrid architectures, inspired by the multi-resolution hash encoding of Müller et al. [15], have proven particularly effective in balancing reconstruction quality, model compactness, and rendering speed. Beyond coordinate-based INRs, 3D Gaussian splatting has recently been explored as an alternative for explorable volume visualization [24], scene editing [19], natural-language interaction [1], feature tracking [33], compressive representation [23], and virtual reality application [11].

However, all of the above methods are fundamentally lossy and cannot guarantee exact reconstruction of the original data. Unlike these approaches, we target lossless representation, ensuring that every voxel is reconstructed without any error. Rather than minimizing a regression loss that tolerates residual error, our Lossless-INR reformulates fitting as per-bit binary classification via bit-plane decomposition for lossless volumetric data representation.

Lossless representation. Achieving lossless representation with neural networks requires the predicted output to match the original signal at every sample point with zero error. For example, Punnappurath and Brown [16] showed that operating on individual bit-planes can recover lost bit-depth information, demonstrating the utility of bit-level signal decomposition. Han et al. [9] introduced ABCD, which adds a bit coordinate to the network input for arbitrary bitwise coefficient prediction. Han et al. [8] further presented a bit-plane decomposition method that reformulates signal fitting as per-bit binary classification, thereby reducing the theoretical upper bound on the number of parameters and enabling lossless INR for 2D images and audio, even at 16-bit precision.

However, directly applying lossless INR training to 3D volumetric data poses significant challenges: the substantially larger data size demands prohibitively large models, and convergence to zero error becomes increasingly difficult as the spatial dimensionality grows. In this work, we address these challenges through ternary-set parameterization of feature-grid entries, which drastically reduces per-parameter storage cost, and a block-partitioning strategy that decomposes the volume into manageable subproblems, thereby making lossless convergence tractable for volumetric 3D data.

3 METHOD

Lossless-INR achieves lossless volumetric representation through three components, as illustrated in Figure 2: *bit-plane decomposition*, *octree block partitioning*, and a *ternary feature grid network*. Bit-plane decomposition (Section 3.1) first decomposes volume data into a set of bit-planes, transforming voxel-value regression into binary classification and thereby enabling lossless representation. However, fitting all bit values over an entire volume remains difficult due to limited network capability. We therefore introduce octree block partitioning (Section 3.2) to adaptively divide complex volumes into simpler local blocks. Finally, we leverage a ternary feature grid network (Section 3.3) whose feature values are constrained to $\{-1,0,+1\}$ to encode each block, enabling lossless reconstruction while maintaining a compact model size.

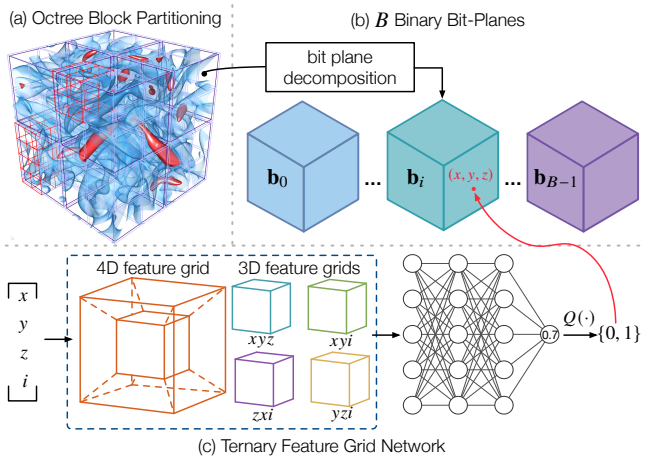


Figure 2: Overview of Lossless-INR. (a) The entire volume is adaptively divided into local blocks with an octree structure, and each block is decomposed into (b) B binary bit-planes, (c) a ternary feature grid network is utilized to losslessly encode each block by classifying bit values.

3.1 Bit-Plane Decomposition

To overcome the inherent precision limitations of continuous voxel value regression, we reformulate the volumetric fitting problem as a binary classification task. Given a volumetric scalar field where each voxel value is represented by B bits of precision, we decompose the voxel value into a set of B binary bit-planes $\{\mathbf{b}_0,\mathbf{b}_1,\ldots,\mathbf{b}_{B-1}\}$. For each bit-plane $\mathbf{b}_i$, we optimize the network f_θ as follows

$$\hat{\theta} = \arg\min_{\theta} \mathscr{L}\left(\mathbf{b}_i(x,y,z), f_\theta(x,y,z,i)\right), \tag{1}$$

where (x,y,z) indicates the spatial position, $\hat{\theta}$ denotes the optimized parameters, and $\mathscr{L}$ corresponds to the binary cross-entropy loss. After training, let $\mathbf{V}(x,y,z)$ denote the voxel value at position (x,y,z). The network reconstructs the volume data as

$$\mathbf{V}(x,y,z) = \frac{1}{2^B-1}\sum_{i=0}^{B-1} 2^i Q\Big(f_\theta(x,y,z,i)\Big), \tag{2}$$

where $Q(\cdot)$ is the quantization function that maps the network's continuous output to the discrete bit value $\{0,1\}$. If f_θ can predict every bit of target volume correctly, it can be considered as a lossless neural representation of that volume.

3.2 Octree Block Partitioning

Fitting an entire scientific volume with a single network f_θ is not trivial; the complexity of optimization can grow quickly with the increase of both spatial resolution and bit precision. We therefore hierarchically divide the volume and encode each block with a separate network. We start from a uniform partition of the volume into non-overlapping blocks and fit each block with its own f_θ for a fixed number of training iterations. If, at the end of training, f_θ predicts all $\mathbf{b}_i(x,y,z)$ correctly for every bit index $i \in \{0,\ldots,B-1\}$, the block is

accepted as an octree leaf. Otherwise, the block is spatially divided into smaller children, each of which becomes a new candidate block. The procedure repeats recursively until every leaf block can be fit losslessly. In practice, we initialize the partition into blocks of size 64^3 and recursively subdivide them down to a minimum block size of 16^3. This adaptive scheme matches model capacity to the complexity of local content. However, because an individual network fits each leaf block, the total number of parameters grows rapidly with the number of leaves; this motivates the ternary feature grid network described next, which keeps each per-block network compact.

3.3 Ternary Feature Grid Network

Our per-block INR consists of multiple feature grids and a lightweight MLP decoder: the feature grids jointly map the input coordinates (x, y, z, i), where i indexes the bit plane, to a continuous spatial feature, and the MLP decodes the feature into a bit value. To keep each per-block model compact, every entry of the feature grids is constrained to a ternary set of parameters, while the MLP decoder stays at full precision.

Feature grids. We build the encoder on the multi-resolution hash-grid design of Instant-NGP [15], in which features are stored on hierarchical grids and queried through linear interpolation. Our encoder consists of one 4D hash grid that operates on the full coordinates (x, y, z, i), together with four 3D hash grids, each operating on a distinct triplet of axes. For each query coordinate, we look up the feature vectors at each grid and concatenate them into a single spatial feature, which is then fed to the MLP for decoding. The 4D grid captures bit-aware spatial structure, while the 3D grids provide complementary lower-dimensional views to improve representation.

Ternary parameterization. Following recent extreme-low-bit network designs [14, 17], we constrain every entry of every hash grid to a ternary set during training. Given a grid weight tensor $\mathbf{W}$, we scale $\mathbf{W}$ by its mean absolute value γ, round each entry to the nearest integer in $\{-1, 0, +1\}$, and rescale back by γ, i.e.,

$$\widetilde{\mathbf{W}} = \gamma \cdot \mathrm{RoundClip}\left(\frac{\mathbf{W}}{\gamma+\varepsilon}, -1, 1\right), \quad \gamma = \frac{1}{n}\sum_j |W_j|, \quad (3)$$

where $\mathrm{RoundClip}(x, a, b) = \max(a, \min(b, \mathrm{round}(x)))$, n is the number of grid entries, and a small value ε for numerical stability. Each entry of resulting weight $\widetilde{\mathbf{W}}$ thus takes one of three values $\{-\gamma, 0, +\gamma\}$, so every hash-grid tensor is fully described by a ternary digit in $\{-1, 0, +1\}$ per entry together with a single per-tensor scale γ. By optimizing and storing ternary weights rather than floating-point weights, our Lossless-INR can achieve a lossless representation while maintaining a relatively compact model size.

Table 1: Volumetric datasets used in our evaluation.

dataset	volume resolution ($x \times y \times z$)	data type
engine	256×256×128	uint8
foot	256×256×256	uint8
MRI-woman	256×256×109	uint16
tooth	103×94×161	uint8
vortex	128×128×128	float32

4 Results

4.1 Datasets, Training, Baselines, and Metrics

Datasets and training. Table 1 lists the volumetric datasets used in our experiments, all of which come from the Open SciVis Dataset repository [12] and cover a range of resolutions and data types (8- and 16-bit unsigned integers and 32-bit floats). Due to page limit, we move the comparison of the engine and tooth datasets to Appendix A. For each block, we use a 4-level multi-resolution hash grid with a hashmap size of 2^{17} and 2-dim features per entry together with four 3D feature grids, followed by a 3-layer, 64-hidden-unit MLP decoder. We optimize with Adam for up to 2,000 iterations per block, using an initial learning rate of 10^{-2}, cosine annealing to 10^{-5}, and a batch size of 32,000. Training stops early once a block reaches zero bit errors. All experiments run on a workstation equipped with an NVIDIA 4090 GPU.

Table 2: PSNR (dB), LPIPS, BER, and training time (TT, in minutes) of Lossless-INR and four INR baselines at matched model size.

dataset	model size	method	PSNR ↑	LPIPS ↓	BER ↓	TT
foot	66.8 MB	ECNR	36.27	0.0710	0.0942	2.07
		fV-SRN	38.01	0.0368	0.0869	0.95
		Instant-NGP	40.54	0.0229	0.0815	0.98
		AMGSRN++	39.41	0.0282	0.0814	4.47
		Lossless-INR	+∞	**0.0000**	**0.0000**	22.20
vortex	91.9 MB	ECNR	57.06	0.0058	0.2692	1.12
		fV-SRN	57.29	0.0114	0.2678	1.55
		Instant-NGP	60.46	0.0087	0.2650	1.63
		AMGSRN++	55.24	0.0178	0.2664	7.83
		Lossless-INR	+∞	**0.0000**	**0.0000**	93.50

Table 3: Ablation study results on the MRI-woman dataset.

variant	lossless	BER	model size
binary grid	×	0.0029	24.6 MB
ternary grid (default)	✓	0.0000	39.2 MB
floating-point grid	✓	0.0000	794.4 MB
w/o blocks + ternary grid	×	0.1989	1.2 MB
w/o blocks + floating-point grid	×	0.1861	24.9 MB

Baselines and metrics. We compare against four representative INRs for volumetric data: ECNR [20], which fits each volume with a Laplacian pyramid of small parallelized block-wise MLPs; fV-SRN [26], which combines a dense latent feature grid with Fourier-feature encoding for fast volume rendering; Instant-NGP [15], which couples a multi-resolution hash grid with a small MLP and shares the same backbone as our Lossless-INR, serving as the direct full-precision counterpart that isolates the effect of our bit-plane formulation and ternary parameterization; and AMGSRN++ [28], which adaptively places multiple feature grids according to local content complexity for large-scale volumetric data. For a fair comparison, we set the baseline methods' model sizes to match those of Lossless-INR on each dataset. We evaluate each method with three quality metrics. We assessed reconstruction quality using three complementary metrics: data-level *peak signal-to-noise ratio* (PSNR), image-level *learned perceptual image patch similarity* (LPIPS) [34], and bit-level *bit-error-rate* (BER), which is the ratio of incorrect bits of reconstructed volume to the total bits.

4.2 Comparison with Baselines

Table 2 reports the quantitative results. The results show that existing lossy representation paradigms cannot achieve lossless reconstruction, even with large parameter budgets. However, because Lossless-INR sequentially encodes each block with the ternary feature grid network, its overall training time is substantially higher than that of the baseline methods. This effect is more pronounced on foot and vortex, where the higher data complexity requires more fine-level blocks to achieve lossless representation. Figure 3 compares the volume-rendering results of different methods. Except for Lossless-INR, all baselines produce nonzero difference images because they do not achieve lossless reconstruction. This suggests that even when reconstruction fidelity is high, small errors can still accumulate during volume visualization, ultimately leading to noticeable deviations from the ground truth. In contrast, our Lossless-INR avoids this issue and is therefore well-suited to volume visualization and analysis tasks with low error tolerance.

4.3 Ablation Studies

In this section, we investigate how the parameter precision of feature grid and octree block partitioning can influence the lossless representation. On the MRI-woman dataset, we compare three feature-grid precisions: binary [17] ($\{-1, +1\}$), ternary ($\{-1, 0, +1\}$), and full

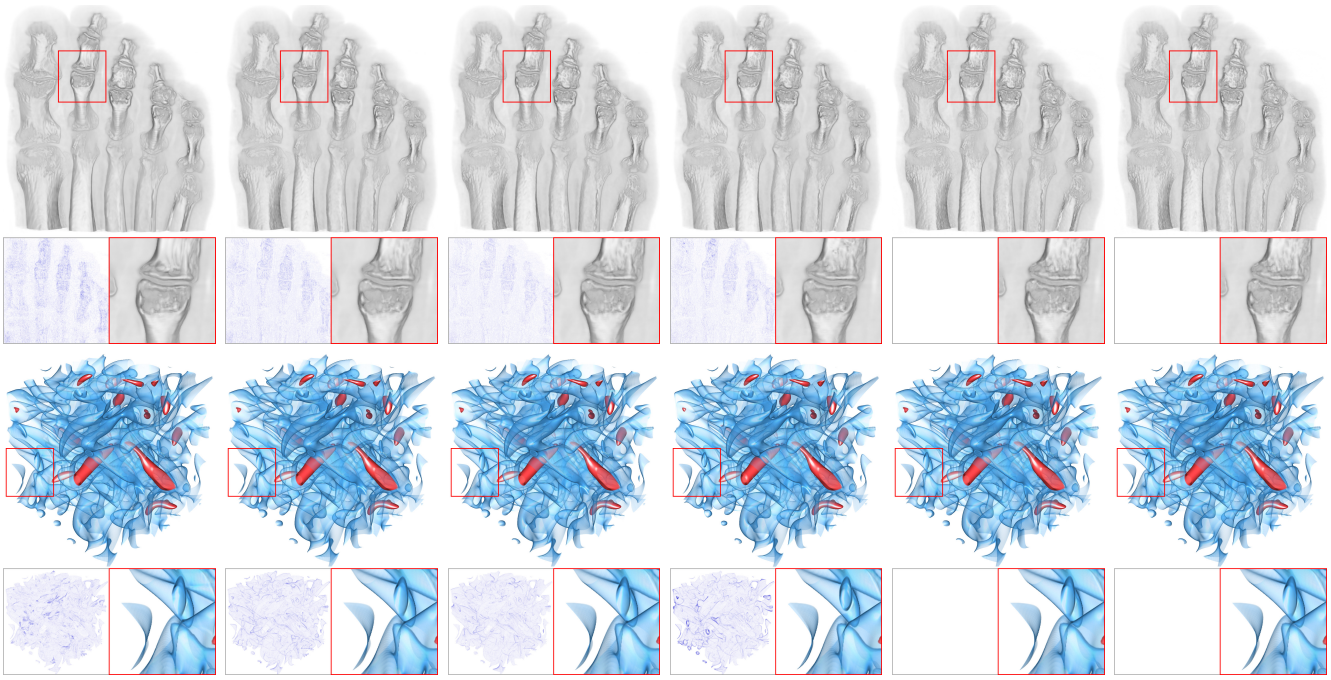


Figure 3: Comparing volume rendering results across different methods with the same model size. Top and bottom: foot and vortex. The bottom-left difference image shows pixel-wise error relative to the ground truth in the CIELUV color space, and the bottom-right inset is a zoom-in of the red-boxed region. Lossless-INR can losslessly reconstruct the ground-truth volume without error.

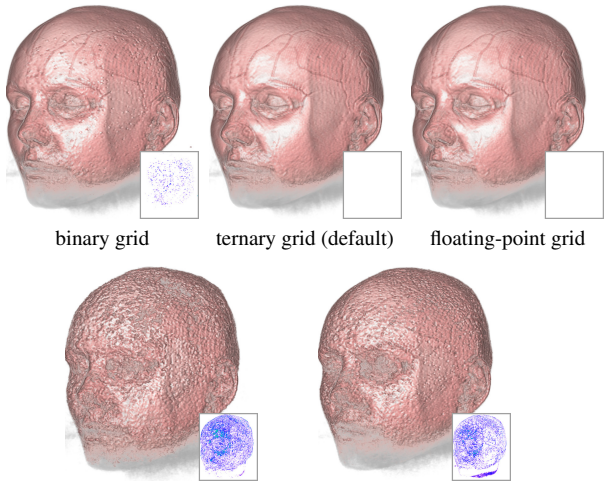


Figure 4: Volume rendering results across different model variants. Our default setting achieves lossless volumetric representation with a model size 20× smaller than its floating-point grid counterpart.

floating-point precision [15], as well as two variants that fit the entire volume with a single INR without block partitioning. Table 3 reports, for each variant, whether it achieves lossless reconstruction, the bit-error rate, and the resulting model size. In addition, Figure 4 provides a qualitative comparison of the reconstructed volumes across all variants. Specifically, the binary grid yields the most compact block-wise model but cannot drive its BER to zero, thereby failing to achieve a lossless representation. The full-precision grid also achieves lossless reconstruction, but its storage cost is approximately 20 times that of the ternary grid. Without block partitioning, neither the floating-point grid nor the ternary grid achieves lossless reconstruction, and both variants exhibit high BER. This highlights the necessity of octree block partitioning for lossless representation.

5 Conclusions and Future Work

We have explored the feasibility of Lossless-INR, a lossless volumetric implicit neural representation. Through bit-plane decomposition and octree block partitioning, Lossless-INR can achieve lossless representation with limited network capacity. To reduce the final model size, we design a ternary feature grid network to encode each partitioned block, whose grid weights are constrained to a ternary set, enabling lossless representation while reducing storage cost by 20× compared with the full-precision counterpart.

However, the model size produced by Lossless-INR after encoding a volume remains relatively large, and can even be larger than the original volume. Although this paper focuses on volume representation rather than compression, our future work will aim to further reduce parameter storage costs while preserving nearly the same representational capacity. One promising direction is to incorporate deep compression techniques [7], such as network pruning [3] and quantization-aware training [10]. Another limitation is the long optimization time of the current framework. Since blocks are optimized sequentially, the training cost grows with the number of blocks, which hinders scaling Lossless-INR to large-scale volumes where optimization could take several days. A straightforward solution is to train all blocks in parallel with multiple GPUs, although this would increase the required computational resources. The per-block optimization can also be accelerated, e.g., by a meta-learned initialization [30] shared across blocks. Hence, each converges in fewer iterations, or by engineering optimizations such as mixed-precision training and fully-fused MLP kernels. Overall, lossless representation is a promising direction for future research, and more compact, efficient models built on a lossless foundation will offer greater practical value.

Acknowledgments

This research was supported in part by the U.S. National Science Foundation through grants IIS-2101696, OAC-2104158, IIS-2401144, and CCF-2550610. The authors thank the anonymous reviewers for their insightful comments.

REFERENCES

[1] K. Ai, K. Tang, and C. Wang. NLI4VolVis: Natural language interaction for volume visualization via LLM multi-agents and editable 3D gaussian splatting. *IEEE Trans. Vis. Comput. Graph.*, 32(1):46–56, 2026. doi: 10.1109/TVCG.2025.3633888 2

[2] Y.-T. Chen, H. Li, N. Shi, X. Luo, W. Xu, and H.-W. Shen. Explorable INR: An implicit neural representation for ensemble simulation enabling efficient spatial and parameter exploration. *IEEE Trans. Vis. Comput. Graph.*, 31(6):3758–3770, 2025. doi: 10.1109/TVCG.2025.3567052 1

[3] H. Cheng, M. Zhang, and J. Q. Shi. A survey on deep neural network pruning: Taxonomy, comparison, analysis, and recommendations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 46(12):10558–10578, 2024. doi: 10.1109/TPAMI.2024.3447085 4

[4] J. Han, K. Tang, and C. Wang. MoE-INR: Implicit neural representation with mixture-of-experts for time-varying volumetric data compression. *IEEE Trans. Vis. Comput. Graph.*, 32(1):254–264, 2026. doi: 10.1109/TVCG.2025.3633893 1

[5] J. Han and C. Wang. CoordNet: Data generation and visualization generation for time-varying volumes via a coordinate-based neural network. *IEEE Trans. Vis. Comput. Graph.*, 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 Trans. Vis. Comput. Graph.*, 30(10):6826–6838, 2024. doi: 10.1109/TVCG.2023.3345373 1, 2

[7] S. Han, H. Mao, and W. J. Dally. Deep compression: Compressing deep neural network with pruning, trained quantization and Huffman coding. In *Proc. ICLR*, 2016. doi: 10.48550/arXiv.1510.00149 4

[8] W. K. Han, B. Lee, H. Cho, S. Im, and K. H. Jin. Towards lossless implicit neural representation via bit plane decomposition. In *Proc. IEEE/CVF CVPR*, pp. 2269–2278, 2025. doi: 10.1109/CVPR52734.2025.00217 1, 2

[9] W. K. Han, B. Lee, S. H. Park, and K. H. Jin. ABCD: Arbitrary bitwise coefficient for de-quantization. In *Proc. IEEE/CVF CVPR*, pp. 5876–5885, 2023. doi: 10.1109/CVPR52729.2023.00569 1, 2

[10] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. G. Howard et al. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proc. IEEE/CVF CVPR*, pp. 2704–2713, 2018. doi: 10.1109/CVPR.2018.00286 4

[11] S. Jeon, K. Tang, C. Wang, and W.-K. Jeong. Super-Gaussian: Interactive scene editing for 3D Gaussian splatting and NLI-based volume visualization in virtual reality. *IEEE Trans. Vis. Comput. Graph.*, 33(1), 2027. Accepted. 2

[12] P. Klacansky. Open SciVis datasets. `https://klacansky.com/open-scivis-datasets/`, 2017. 3

[13] Y. Lu, K. Jiang, J. A. Levine, and M. Berger. Compressive neural representations of volumetric scalar fields. *Comput. Graph. Forum*, 40(3):135–146, 2021. doi: 10.1111/cgf.14295 1, 2

[14] S. Ma, H. Wang, L. Ma, L. Wang, W. Wang, et al. The era of 1-bit LLMs: All large language models are in 1.58 bits. *arXiv preprint arXiv:2402.17764*, 2024. doi: 10.48550/arXiv.2402.17764 3

[15] T. Müller, A. Evans, C. Schied, and A. Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, 2022. doi: 10.1145/3528223.3530127 1, 2, 3, 4

[16] A. Punnappurath and M. S. Brown. A little bit more: Bitplane-wise bit-depth recovery. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(12):9718–9724, 2022. doi: 10.1109/TPAMI.2021.3125692 1, 2

[17] S. Shin and J. Park. Binary radiance fields. In *Proc. NeurIPS*, pp. 55919–55931, 2023. 3

[18] H. Son, J. Noh, S. Jeon, C. Wang, and W.-K. Jeong. MC-INR: Efficient encoding of multivariate scientific simulation data using meta-learning and clustered implicit neural representations. In *Proc. IEEE VIS (Short Papers)*, pp. 206–210, 2025. doi: 10.1109/VIS60296.2025.00047 2

[19] K. Tang, K. Ai, J. Han, and C. Wang. TexGS-VolVis: Expressive scene editing for volume visualization via textured gaussian splatting. *IEEE Trans. Vis. Comput. Graph.*, 32(1):933–943, 2026. doi: 10.1109/TVCG.2025.3634643 2

[20] K. Tang and C. Wang. ECNR: Efficient compressive neural representation of time-varying volumetric datasets. In *Proc. IEEE PacificVis*, pp. 72–81, 2024. doi: 10.1109/PACIFICVIS60374.2024.00017 1, 2, 3

[21] K. Tang and C. Wang. STSR-INR: Spatiotemporal super-resolution for multivariate time-varying volumetric data via implicit neural representation. *Comput. Graph.*, 119:103874, 2024. doi: 10.1016/j.cag.2024.01.001 1, 2

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

[23] K. Tang and C. Wang. ECoNGS: Efficient compressive neural Gaussian splats for volume visualization. *IEEE Trans. Vis. Comput. Graph.*, 33(1), 2027. Accepted. 2

[24] K. Tang, S. Yao, and C. Wang. iVR-GS: Inverse volume rendering for explorable visualization via editable 3D gaussian splatting. *IEEE Trans. Vis. Comput. Graph.*, 31(6):3783–3795, 2025. doi: 10.1109/TVCG.2025.3567121 2

[25] C. Wang and J. Han. DL4SciVis: A state-of-the-art survey on deep learning for scientific visualization. *IEEE Trans. Vis. Comput. Graph.*, 29(8):3714–3733, 2023. doi: 10.1109/TVCG.2022.3167896 2

[26] S. Weiss, P. Hermüller, and R. Westermann. Fast neural representations for direct volume rendering. *Comput. Graph. Forum*, 41(6):196–211, 2022. doi: 10.1111/cgf.14578 1, 2, 3

[27] Q. Wu, D. Bauer, M. J. Doyle, and K.-L. Ma. Interactive volume visualization via multi-resolution hash encoding based neural representation. *IEEE Trans. Vis. Comput. Graph.*, 30(8):5404–5418, 2024. doi: 10.1109/TVCG.2023.3293121 1, 2

[28] S. W. Wurster and H.-W. Shen. AMGSRN++: Improved adaptive SRN for scientific visualization. In *Proc. IEEE PacificVis*, pp. 182–191, 2025. doi: 10.1109/PACIFICVIS64226.2025.00024 1, 3

[29] 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 Trans. Vis. Comput. Graph.*, 30(1):965–974, 2024. doi: 10.1109/TVCG.2023.3327194 1, 2

[30] M. Yang, K. Tang, and C. Wang. Meta-INR: Efficient encoding of volumetric data via meta-learning implicit neural representation. In *Proc. IEEE PacificVis*, pp. 246–251, 2025. doi: 10.1109/PACIFICVIS64226.2025.00030 1, 2, 4

[31] S. Yao, Y. Lu, and C. Wang. ViSNeRF: Efficient multidimensional neural radiance field representation for visualization synthesis of dynamic volumetric scenes. In *Proc. IEEE PacificVis*, pp. 235–245, 2025. doi: 10.1109/PacificVis64226.2025.00029 1

[32] S. Yao and C. Wang. ReVolVE: Neural reconstruction of volumes for visualization enhancement of direct volume rendering. *Comput. Graph.*, 131:104295, 2025. doi: 10.1016/j.cag.2025.104295 1

[33] S. Yao and C. Wang. VolSegGS: Segmentation and tracking in dynamic volumetric scenes via deformable 3D Gaussians. *IEEE Trans. Vis. Comput. Graph.*, 32(1):407–417, 2026. doi: 10.1109/TVCG.2025.3642516 2

[34] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. IEEE/CVF CVPR*, pp. 586–595, 2018. doi: 10.1109/CVPR.2018.00068 3

A Comparing on Additional Datasets

Due to page limit, we do not include all the comparisons in the main paper. In this section, we report the results for the other two datasets, engine and tooth. As in the comparison in the main paper, we keep the model sizes for all methods the same. Table 1 lists the quantitative results, and Figure 1 shows the corresponding volume rendering results. We can observe that for lossy representation methods, even when reconstruction quality is high, the reconstruction still contains errors in a non-blank difference image. In contrast, Lossless-INR is the only method that achieves a zero error rate and produces pixel-wise identical results to the ground truth.

Table 1: PSNR (dB), LPIPS, BER, and training time (TT, in minutes) of Lossless-INR and four INR baselines on the engine and tooth datasets at matched model size.

dataset	model size	method	PSNR ↑	LPIPS ↓	BER ↓	TT
engine	61.0 MB	ECNR	45.67	0.0446	0.0728	1.82
		fV-SRN	47.10	0.0446	0.0682	0.93
		Instant-NGP	49.99	0.0311	0.0606	0.98
		AMGSRN++	48.47	0.0332	0.0647	4.63
		Lossless-INR	$+\infty$	**0.0000**	**0.0000**	2.20
tooth	12.5 MB	ECNR	49.68	0.0443	0.0888	0.91
		fV-SRN	46.76	0.0720	0.1578	0.43
		Instant-NGP	47.36	0.0571	0.1601	0.70
		AMGSRN++	46.02	0.0527	0.1631	1.16
		Lossless-INR	$+\infty$	**0.0000**	**0.0000**	0.94

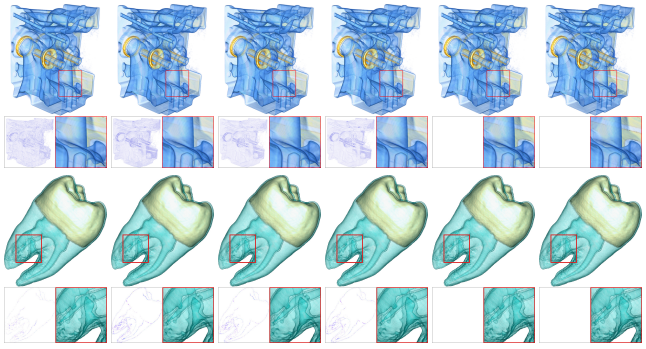
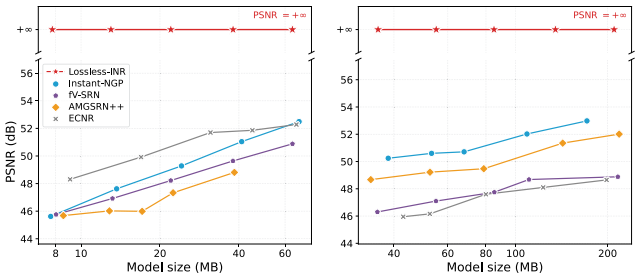


Figure 1: Comparing volume rendering results across different methods with the same model size. Top and bottom: engine and tooth. The bottom-left difference image shows pixel-wise error relative to the ground truth in the CIELUV color space, and the bottom-right inset is a zoom-in of the red-boxed region.

B Rate-Distortion Curves

We report the rate-distortion behavior of each method in Figure 2, where data-level PSNR is plotted with varying model sizes on the tooth and engine datasets. This analysis goes beyond a single comparison at a matched size and provides practical guidance on when a lossless representation becomes necessary. Conventional INR baselines generally improve reconstruction quality by increasing the model size, since additional parameters increase representational capacity. However, these methods still optimize a voxel-value regression objective. As a result, their reconstruction quality may eventually saturate and, in some cases, even degrade once the model size exceeds a certain threshold; adding parameters yields only trivial improvements in quality. Therefore, further improving fidelity requires changing the formulation rather than merely increasing capacity. Lossless-INR addresses this limitation by predicting the per-bit values associated with each voxel, which enables bit-exact reconstruction. From this perspective, the model size at which Lossless-INR achieves lossless reconstruction can be interpreted as



(a) tooth (b) engine

Figure 2: Rate-distortion curves for different methods on two datasets.

an upper bound on the parameter budget that is worth spending on a lossy representation. For example, in Figure 2 (a), Lossless-INR already achieves bit-exact reconstruction at roughly 8 MB. Once the baseline methods exceed this size without achieving lossless fidelity, their voxel-value objective becomes the limiting factor, indicating that one should switch to Lossless-INR rather than further scaling the lossy models. We note that the current Lossless-INR model is still larger than the raw volume. Nevertheless, we believe future designs can substantially reduce this overhead while preserving the random-access advantage of INR representations, ultimately enabling efficient volume visualization.