Efficient Compression of Sparse Accelerator Data Using Implicit Neural Representations and Importance Sampling

Anonymous Author(s)
Affiliation
Address
email

Abstract

High-energy, large-scale particle colliders in nuclear and high-energy physics generate data at extraordinary rates, reaching up to 1 terabyte and several petabytes per second, respectively. The development of real-time, high-throughput data compression algorithms capable of reducing this data to manageable sizes for permanent storage is of paramount importance. A unique characteristic of the tracking detector data is the extreme sparsity of particle trajectories in space, with an occupancy rate ranging from approximately 10^{-6} to 10%. Furthermore, for downstream tasks, a continuous representation of this data is often more useful than a voxel-based, discrete representation due to the inherently continuous nature of the signals involved. To address these challenges, we propose a novel approach using implicit neural representations for data learning and compression. We also introduce an importance sampling technique to accelerate the network training process. Our method matches traditional compression algorithms while offering significant speed-up and maintaining negligible accuracy loss, leveraging the proposed importance sampling strategy.

1 Introduction

High-energy particle accelerators, such as the Large Hadron Collider (LHC) and the Relativistic Heavy Ion Collider (RHIC), represent pinnacle achievements in modern physics, enabling profound investigations into the fundamental particles and forces of the universe. The operation of these colossal machines generates an immense volume of data, necessitating efficient compression methods to manage and analyze the deluge of information effectively. Traditionally, data compression for such accelerators has been challenging due to the sheer scale and complexity of the data involved [12, 17, 20, 21].

Recent advances in deep learning have introduced novel approaches to data compression that surpass traditional methods in both efficiency and effectiveness [5, 15, 33, 37]. Unlike conventional techniques (e.g., SC and ZFP), deep learning models are adept at capturing intricate and non-linear patterns in large datasets. However, most current deep learning-based compression models rely on grid-based, resolution-fixed representations that are ill-suited for scientific applications where a continuous representation of data is crucial [23, 29, 31].

Implicit Neural Representations (INRs) have recently emerged as a promising method in the field of machine learning, offering a pathway to efficient compression and continuous representation learning [35]. These models excel in representing data in a continuous form, enabling more flexible and granular analysis and reconstruction [16, 18, 28, 36]. Nevertheless, the typical deployment of INRs has primarily focused on dense datasets, such as images, and not on the sparse, irregularly distributed data characteristic of particle accelerators.

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

One of the unique challenges in accelerator data is its sparsity–the trajectories of particles are extremely sparse in space, presenting a significant obstacle to traditional INR applications. Moreover, the need for speed in compression is paramount. Traditional INR training methods, which involve processing all data points, become inefficient in the context of accelerator data, where a substantial proportion of the data points are zero-valued. This characteristic significantly impedes the training process, necessitating a novel approach to manage and exploit the sparse nature of the data effectively.

This work first investigates the adaptability of INRs to learn from sparse data typical of accelerator outputs. We propose an innovative importance sampling-based strategy that selectively trains on the most informative data points rather than the entire dataset. This approach not only preserves the integrity and granularity of the data but also significantly accelerates the training process.

2 Method

The objectives of this study are twofold: firstly, to assess the capability of INRs in representing sparse data; and secondly, to explore a novel importance sampling technique designed to significantly expedite training on sparse datasets by adapting to the differential importance of grid points.

Problem Setup We analyze 3D time projection chamber (TPC) data from a minipad array with $r = 48$ cylindrical layers divided into three groups of $r = 16$ layers each–inner, middle, and outer. When unwrapped, these layers form a rectangular grid with z rows along the axial dimension and c columns along the azimuthal dimension of the TPC. Despite consistent row numbers (z) across all groups, the column numbers (c) vary by layer group. We focus on the *outer* layer group for this study. The full 3D data volume for an outer layer group is $(c, z, r) = (2304, 498, 16)$. For synchronization with the TPC's data concentrator, the data is segmented into 12 non-overlapping azimuthal sections and the horizontal dimension is halved, resulting in a processed data shape of $(c, z, r) = (192, 249, 16)$. Figure 1 showcases the data utilized in this study. Additional data details are provided in Appendix A.

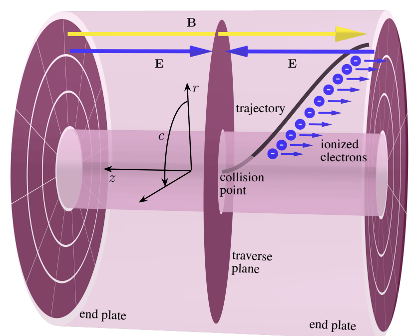

Figure 1: Illustration of the working principal for the time projection chamber (TPC) of sPHENIX Experiment [1, 22]. For simplicity, a single charge particle is visualized, as it is produced at the collision point, and traverses through the TPC leaving ion-electron pairs along its trajectory. These ionization electrons drift along an electrical field to the end plate for amplification and readout. During experiment, thousands of particle can be produced at a single collision and tracks from multiple collisions can pile up onto each other in the TPC data.

Objective The model input, denoted as $\mathbf{x} = (c, z, r)$, maps to the signal intensity y at these coordinates through the function:

$$f(\mathbf{x}; \mathbf{\Theta}) = y \tag{1}$$

Here, f represents the INR function, parameterized by weights $\mathbf{\Theta}$. In this study, we employ three distinct models of INRs to address the challenges of compressing sparse accelerator data: Sinusoidal Representation Networks (SIREN) [30], Fourier Feature Networks (FFNet) [32], and Wavelet Implicit Neural Representations (WIRE) [27]. Network details are available in Appendix B.

2.1 Sampling Strategies

Importance Sampling The core of the importance sampling lies in assigning sampling probabilities based on the informativeness of the data points. Specifically, data points with non-zero values are deemed more informative for the model and are consequently assigned higher sampling probabilities. Formally, the sampling weights are calculated as follows:

$$\text{weights} = \frac{w(y_i)}{\sum w(y_i)} \quad \text{where} \quad w(y_i) = \begin{cases} |y_i| & \text{if } y_i \neq 0, \\ \epsilon & \text{if } y_i = 0. \end{cases} \tag{2}$$

with ϵ denoting a small number. This formulation ensures that non-zero data points are more likely to be selected during the sampling process, thereby reducing the disproportionate influence of zero-value data in the training set.

Entropy-based Sampling Data points with low-probability values are often more informative for visualization and discovery, as established in the literature [7, 8, 9]. For instance, in image analysis, foreground pixels, though rarer, hold more significance than the abundant background pixels. Here, entropy-based sampling is a method that prioritizes rare data points to enhance INR training. The key idea is to overrepresent rare values while retaining a representative sample. The importance function (IF) assigns lower priorities to common points and higher priorities to rare ones. Formally, let y denote a data point's field value and $p(y)$ the probability density function of these values. Rare values correspond to low $p(y)$, and common values to high $p(y)$. The importance function is defined as: $\mathrm{IF}(y) \propto \frac{1}{p(y)}$ on the support of y. To achieve a uniform target distribution over field values y, bounded by ℓ and u, we adjust the sampled data to upweight rare values and downweight common ones. This process resembles rejection sampling [6, 26], where acceptance is governed by the ratio: $\frac{f(y)}{C \cdot p(y)}$, with C ensuring $f(y) \leq C \cdot p(y)$ for all y. Here, $p(y)$ represents the dataset's PDF, and $f(y)$ the target uniform distribution. The importance function $\mathrm{IF}(y)$ thus guides sample selection, overrepresenting rare values. More implementation details are available at Appendix C.

3 Experiments

3.1 Continuous Reconstruction

Task. The goal of this task is to evaluate the ability of INRs to learn continuous patterns from sparse 3D data from an example collider tracking detector, the TPC of sPHENIX Experiment [1, 22] as illustrated in Figure 1. Our focus is on determining whether these models can effectively reconstruct data at arbitrary resolutions, which is crucial for enhancing the flexibility and utility of INRs in physics.

Setup. All models are initially trained on data at its original resolution (super-resolution scale $S = 1$) to establish a performance benchmark. In the subsequent experiments, data is progressively downsampled to lower resolutions (e.g., $y_{\text{half}} = \text{downsample}(y_{\text{orig}}, 4)$) and then reconstructed back to the original resolution (e.g., $S = 4$).

Results summary. Figure 2 provides a qualitative comparison of reconstruction and super-resolution results across different INR models. The results indicate that while all models achieve faithful data reconstruction, SIREN consistently outperforms the others, with FFNet showing the least effective performance. In the context of super-resolution, all models maintain high-quality outputs even when the vertical and azimuthal dimensions are downsampled by a factor of 2, resulting in a total super-resolution scale of $S = 4$ ($\times 2$ in z and $\times 2$ in c). Impressively, the models continue to deliver accurate continuous reconstructions at higher scales, such as $S = 16$ ($\times 4$ in z and $\times 4$ in c). However, a significant performance drop is observed when the number of layers is halved, that is, ($\times 2$ in r). This decline could be attributed to the relatively small number of layers (16 in total) compared to the other dimensions, where the resolution is much higher (2304 for c and 498 for r). Therefore, it may be more advantageous to reduce resolution in the z and c dimensions rather than r. Due to space constraints, a more detailed summary of the results is provided in the Appendix D.

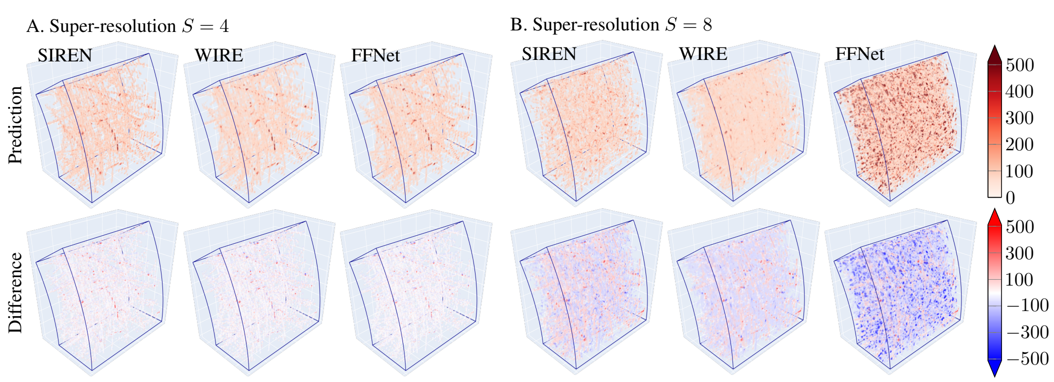

Figure 2: Qualitative results of continuous reconstruction with super-resolution scales of $\times 4$ and $\times 8$. The $\times 4$ super-resolution is trained on $(96, 125, 16)$ and and the $\times 8$, on $(96, 125, 8)$. Both are evaluated on the full resolution $(192, 249, 16)$.

3.2 Compression

Task. The goal of this task is to assess the performance of INRs in compressing TPC data. Notably, the entire dataset is compressed into the neural network, with no latent space required—only the network itself needs to be stored.

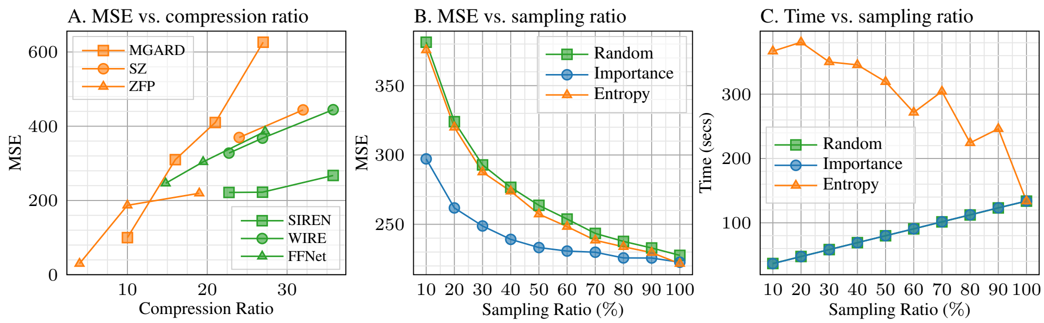

Figure 3: Panel A. MSE vs. compression ratio for conventional method (MGARD, SZ, and ZFP) nd INR approaches (SIREN, WIRE, and FFNet). Panel B. MSE vs. sampling ratio for different sampling methods based on the SIREN algorithm. Panel C. time vs. sampling ratio for different sampling methods.

Setup. We will compare the proposed INR models with three traditional compression algorithms: MGARD [2, 10], a multilevel lossy compression technique based on multigrid methods; ZFP [25], a compressed format for multidimensional arrays with spatial correlation; and SZ [14, 24], an error-controlled lossy algorithm optimized for high compression ratios.

Results summary. For traditional methods, ZFP operates by transforming, quantizing, and entropy coding blocks of data. However, if the target bit rate per value is too low, the algorithm struggles to represent the transformed and quantized data within the specified bit budget. In our experiments, we observed that ZFP's compression threshold is around 20; beyond this point, it begins to lose a significant amount of information in the compressed data (See Figure 3A). Similarly, SZ can be configured to target a specific compression ratio, but it generally focuses on controlling error bounds rather than directly targeting a compression ratio. When SZ is used in iterative computational processes, extreme compression can lead to non-convergence or instability, as excessive compression degrades data accuracy. On the other hand, INR-based methods demonstrate competitive performance, surpassing traditional methods like MGARD and SZ. Notably, SIREN excels in this regard, outperforming both traditional and other INR-based methods when the compression ratio exceeds 20. For instance, SIREN achieves comparable MSE to ZFP while delivering higher compression efficiency.

3.3 Efficiency

Task. The goal of this task is to evaluate the speed-up achieved using different sampling methods. Additionally, we will examine the trade-off between accuracy and speed.

Setup. We explored three sampling methods: Importance Sampling (IS), random sampling, and entropy-based sampling. All experiments were conducted using SIREN as the baseline model.

Results summary. From Figure 3B, we can observer that IS consistently achieves the lowest MSE across all sampling ratios, highlighting its effectiveness in capturing the most critical data points for training. In contrast, Entropy-based sampling initially produces higher MSE values than IS but demonstrates gradual improvement as the sampling ratio increases. While IS is the most effective method for minimizing MSE, it shows a linear increase in computation time as the sampling ratio grows (See Figure 3C). Rand sampling has nearly identical computational time to IS, also increasing linearly with the sampling ratio. It also offers significant speed-ups compared to the full sampling, though at the cost of slightly higher MSE. Entropy-based sampling, although effective at full data usage, is the most computationally expensive, particularly at lower sampling ratios. Overall, IS offers the best balance between accuracy and computational efficiency, making it the most suitable method for applications where high accuracy is critical, even at the cost of moderately higher computational demands.

4 Conclusion

In this work, we address the challenge of compressing the vast, sparse data generated by high-energy particle colliders using implicit neural representations combined with importance sampling. Our method achieves comparable compression performance to traditional algorithms while providing a significant speedup in training time with minimal accuracy loss. This approach offers a scalable and efficient solution for real-time data processing in physics research.

References

[1] A. Adare et al. An Upgrade Proposal from the PHENIX Collaboration. 1 2015.

[2] Mark Ainsworth, Ozan Tugluk, Ben Whitney, and Scott Klasky. Multilevel techniques for compression and reduction of scientific data—the univariate case. *Computing and Visualization in Science*, 19(5):65–76, 2018.

[3] John Allison, Katsuya Amako, John Apostolakis, Pedro Arce, Makoto Asai, Tsukasa Aso, Enrico Bagli, A Bagulya, S Banerjee, GJNI Barrand, et al. Recent developments in geant4. *Nuclear instruments and methods in physics research section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 835:186–225, 2016.

[4] Babak Azmoun, P Garg, TK Hemmick, M Hohlmann, A Kiselev, ML Purschke, C Woody, and A Zhang. Design studies for a tpc readout plane using zigzag patterns with multistage gem detectors. *IEEE Transactions on Nuclear Science*, 65(7):1416–1423, 2018.

[5] Daniel Becking, Paul Haase, Heiner Kirchhoffer, Karsten Müller, Wojciech Samek, and Detlev Marpe. Nncodec: An open source software implementation of the neural network coding iso/iec standard. In *ICML 2023 Workshop Neural Compression: From Information Theory to Applications*, 2023.

[6] Christopher M Bishop. Pattern recognition and machine learning. *Springer google schola*, 2:1122–1128, 2006.

[7] Ayan Biswas, Soumya Dutta, Earl Lawrence, John Patchett, Jon C Calhoun, and James Ahrens. Probabilistic data-driven sampling via multi-criteria importance analysis. *IEEE Transactions on Visualization and Computer Graphics*, 27(12):4439–4454, 2020.

[8] Roger Bramon, Imma Boada, Anton Bardera, Joaquim Rodriguez, Miquel Feixas, Josep Puig, and Mateu Sbert. Multimodal data fusion based on mutual information. *IEEE Transactions on Visualization and Computer Graphics*, 18(9):1574–1587, 2011.

[9] Stefan Bruckner and Torsten Möller. Isosurface similarity maps. In *Computer Graphics Forum*, volume 29, pages 773–782. Wiley Online Library, 2010.

[10] Jieyang Chen, Lipeng Wan, Xin Liang, Ben Whitney, Qing Liu, David Pugmire, Nicholas Thompson, Jong Youl Choi, Matthew Wolf, Todd Munson, et al. Accelerating multigrid-based hierarchical scientific data refactoring on gpus. In *2021 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 859–868. IEEE, 2021.

[11] Kai Chen, Hucheng Chen, Jin Huang, Francesco Lanni, Shaochun Tang, and Weihao Wu. A generic high bandwidth data acquisition card for physics experiments. *IEEE Transactions on Instrumentation and Measurement*, 69(7):4569–4577, 2019.

[12] Abhik Datta, Kian Fong Ng, Deepan Balakrishnan, Melissa Ding, See Wee Chee, Yvonne Ban, Jian Shi, and N Duane Loh. A data reduction and compression description for high throughput time-resolved electron microscopy. *Nature communications*, 12(1):664, 2021.

[13] Cameron Thomas Dean. The sphenix experiment at rhic. *PoS-Proceedings of Science*, 390(LA-UR-20-30202), 2021.

[14] Sheng Di and Franck Cappello. Fast error-bounded lossy hpc data compression with sz. In *2016 ieee international parallel and distributed processing symposium (ipdps)*, pages 730–739. IEEE, 2016.

[15] Yann Dubois, Benjamin Bloem-Reddy, Karen Ullrich, and Chris J Maddison. Lossy compression for lossless prediction. *Advances in Neural Information Processing Systems*, 34:14014–14028, 2021.

[16] Emilien Dupont, Adam Goliński, Milad Alizadeh, Yee Whye Teh, and Arnaud Doucet. Coin: Compression with implicit neural representations. *arXiv preprint arXiv:2103.03123*, 2021.

[17] Ekaterina Govorkova, Ema Puljak, Thea Aarrestad, Thomas James, Vladimir Loncar, Maurizio Pierini, Adrian Alan Pol, Nicolò Ghielmetti, Maksymilian Graczyk, Sioni Summers, et al. Autoencoders on field-programmable gate arrays for real-time, unsupervised new physics detection at 40 mhz at the large hadron collider. *Nature Machine Intelligence*, 4(2):154–161, 2022.

[18] Zongyu Guo, Gergely Flamich, Jiajun He, Zhibo Chen, and José Miguel Hernández-Lobato. Compression with bayesian implicit neural representations. *Advances in Neural Information Processing Systems*, 36:1938–1956, 2023.

[19] Hugo Hernandez, Bruno Cavalcante de Souza Sanches, Dionisio Carvalho, Marco Bregant, Armando Ayala Pabon, Ronaldo Wilton da Silva, Raul Acosta Hernandez, Tiago Oliveira Weber, Andre Luis do Couto, Arthur Campos, et al. A monolithic 32-channel front end and dsp asic for gaseous detectors. *IEEE Transactions on Instrumentation and Measurement*, 69(6):2686–2697, 2019.

[20] Yi Huang, Yihui Ren, Shinjae Yoo, and Jin Huang. Fast 2d bicephalous convolutional autoencoder for compressing 3d time projection chamber data. In *Proceedings of the SC'23 Workshops of The International Conference on High Performance Computing, Network, Storage, and Analysis*, pages 298–305, 2023.

[21] LC Jarrott, MS Wei, C McGuffey, AA Solodov, W Theobald, B Qiao, C Stoeckl, R Betti, H Chen, J Delettrez, et al. Visualizing fast electron energy transport into laser-compressed high-density fast-ignition targets. *Nature Physics*, 12(5):499–504, 2016.

[22] Henry T. Klest. A Compact TPC for the sPHENIX Experiment. *J. Phys. Conf. Ser.*, 2374(1):012147, 2022.

[23] Théo Ladune, Pierrick Philippe, Wassim Hamidouche, Lu Zhang, and Olivier Déforges. Conditional coding for flexible learned video compression. In *Neural Compression: From Information Theory to Applications–Workshop@ ICLR 2021*.

[24] Xin Liang, Sheng Di, Dingwen Tao, Sihuan Li, Shaomeng Li, Hanqi Guo, Zizhong Chen, and Franck Cappello. Error-controlled lossy compression optimized for high compression ratios of scientific datasets. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 438–447. IEEE, 2018.

[25] Peter Lindstrom. Fixed-rate compressed floating-point arrays. *IEEE transactions on visualization and computer graphics*, 20(12):2674–2683, 2014.

[26] Radford M Neal. Slice sampling. *The annals of statistics*, 31(3):705–767, 2003.

[27] Vishwanath Saragadam, Daniel LeJeune, Jasper Tan, Guha Balakrishnan, Ashok Veeraraghavan, and Richard G Baraniuk. Wire: Wavelet implicit neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18507–18516, 2023.

[28] Jonathan Richard Schwarz, Jihoon Tack, Yee Whye Teh, Jaeho Lee, and Jinwoo Shin. Modality-agnostic variational compression of implicit neural representations. In *International Conference on Machine Learning*, pages 30342–30364. PMLR, 2023.

[29] Tong Shao, Jay N Shingala, Ajay Shyam, Peng Yin, Arjun Arora, and Sean McCarthy. Low complexity neural network-based in-loop filtering with decomposed split luma-chroma model for video compression. In *ICML 2023 Workshop Neural Compression: From Information Theory to Applications*, 2023.

[30] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33:7462–7473, 2020.

[31] Yang Sui, Zhuohang Li, Ding Ding, Xiang Pan, Xiaozhong Xu, Shan Liu, and Zhenzhong Chen. Reconstruction distortion of learned image compression with imperceptible perturbations. In *ICML 2023 Workshop Neural Compression: From Information Theory to Applications*.

[32] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in neural information processing systems*, 33:7537–7547, 2020.

[33] James Townsend and Iain Murray. Lossless compression with state space models using bits back coding. In *Neural Compression: From Information Theory to Applications–Workshop@ ICLR 2021*.

[34] Xin-Nian Wang and Miklos Gyulassy. Hijing: A monte carlo model for multiple jet production in pp, pa, and aa collisions. *Physical Review D*, 44(11):3501, 1991.

[35] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. In *Computer Graphics Forum*, volume 41, pages 641–676. Wiley Online Library, 2022.

[36] Yibo Yang, Robert Bamler, and Stephan Mandt. Improving inference for neural image compression. *Advances in Neural Information Processing Systems*, 33:573–584, 2020.

[37] Yibo Yang, Stephan Mandt, Lucas Theis, et al. An introduction to neural data compression. *Foundations and Trends® in Computer Graphics and Vision*, 15(2):113–200, 2023.

A Data Configuration

We analyze simulated data from 200 GeV Au+Au collisions detected by the sPHENIX TPC, leveraging the HIJING event generator [34] and the Geant4 Monte Carlo detector simulation package [3], integrated within the sPHENIX software framework. The sPHENIX TPC is designed to detect thousands of charged particles produced in high-energy Au+Au collisions at the Relativistic Heavy Ion Collider (RHIC), operating at collision rates of approximately 100 kHz. The ionization charges generated by these collisions are captured within the TPC gas volume, drifted, amplified, and collected by 160,000 mini pads [4], and subsequently digitized using the SAMPA v5 application-specific integrated circuit at a 20 MHz rate [19, 13].

As the ionization charge drifts along the z-axis at approximately 8 cm/μs, the corresponding ADC (Analog-Digital Converter) time series data provides a measure of the z-location dependent ionization charge density. These ADC values are quantified as 10-bit unsigned integers, ranging from 0 to 1023, which represent the initial ionization charge density. Spatial interpolation of the trajectory location between neighboring pads is derived from the ADC amplitudes, emphasizing the need to maintain relative ADC ratios in lossy compression strategies. Prior to data readout, zero suppression is implemented on the SAMPA chips, setting ADC values below a threshold of 64 to zero, simplifying the data stream.

Data from the SAMPA chips are then transmitted via 960 6-Gbps optical fibers through the FELIX interfaces [11] to a network of commodity computing servers, where the potential for on-the-fly compression by algorithms embedded in field-programmable gate arrays or directly on the servers exists. The detector's TPC minipad array consists of 48 cylindrical layers, categorized into three sets (inner, middle, and outer), each containing 16 layers. When expanded, each layer forms a rectangular grid with consistent rows in the z-direction across all groups but varying column counts in the azimuthal direction due to different layer group configurations. The 3D data volume for an outer layer group, for instance, takes the form of $(2304, 498, 16)$ across azimuthal, horizontal, and radial dimensions. To align with the segmentation protocols of the TPC's readout data concentrator, we segment a full data frame into 12 distinct non-overlapping sections along the azimuth and reduce the horizontal dimension by half, resulting in a processed data shape of $(192, 249, 16)$.

B Model Architecture

We begin by leveraging a standard Multi-Layer Perceptron (MLP) for modeling Implicit Neural Representations (INRs). MLPs, composed of multiple fully connected layers with nonlinear activation functions, are known for their universal approximation capabilities across various tasks. Mathematically, an MLP with L layers is expressed as:

$$f(x) = W_L\sigma(W_{L-1}\sigma(\cdots\sigma(W_1x + b_1)\cdots) + b_{L-1}) + b_L$$

where W_i and b_i represent the weights and biases of the i-th layer, respectively, and σ denotes the activation function.

Despite their versatility, MLPs exhibit an issue called **spectral bias**, favoring the learning of low-frequency components and struggling with high-frequency content. This limitation significantly hinders their performance in tasks requiring fine-grained detail, such as scientific data modeling and signal reconstruction. To mitigate the spectral bias, we explore three advanced methods designed to enhance the ability of MLPs to capture high-frequency information: **FFNet** [32], **SIREN** [30], and **WIRE** [27].

FFNet addresses the spectral bias by introducing Fourier features that map the input x into a higher-dimensional space. This mapping is defined as:

$$\gamma(x) = [\sin(2\pi Bx), \cos(2\pi Bx)]$$

where B is a matrix of frequencies drawn from a Gaussian distribution. The transformed features $\gamma(x)$ enrich the input with high-frequency components, enabling the MLP to model complex functions more effectively.

The FFNet model is therefore formulated as:

$$f(x) = \text{MLP}(\gamma(x))$$

This approach significantly improves the network's ability to capture detailed and high-frequency variations in the data.

SIREN mitigates spectral bias by employing sinusoidal activation functions. These periodic functions, such as the sine function, naturally encode high-frequency information. The SIREN model is given by:

$$f(x) = W_L \sin(W_{L-1} \sin(\cdots \sin(W_1 x + b_1) \cdots) + b_{L-1}) + b_L$$

By replacing traditional activations with sine functions, SIREN effectively captures high-frequency details, making it particularly suitable for applications in neural rendering and signal processing.

WIRE introduces wavelet transforms into the INR framework to capture multi-scale information. Wavelets provide a powerful means to decompose signals into different frequency bands, enabling the model to capture both local and global features. The WIRE model is expressed as:

$$f(x) = \sum_{j,k} c_{j,k} \Psi_{j,k}(x)$$

where $c_{j,k}$ are the wavelet coefficients and $\Psi_{j,k}(x)$ represents the wavelet basis functions. By integrating wavelet transforms with neural networks, WIRE efficiently models both high-frequency and low-frequency components.

C Sampling Strategies

Importance Sampling. Both the data and their corresponding weights are flattened into one-dimensional arrays, enabling simplified index management. The stochastic selection of data points is then performed using the 'torch.multinomial' function, which allows for weighted sampling with replacement:

$$\text{indices} = \text{torch.multinomial(weights\_flat, num\_samples, replacement=True)}$$

The sampled data points are subsequently retrieved based on these indices:

$$\text{sampled\_data} = \text{data\_flat[indices]}$$

By prioritizing non-zero data points, the importance sampling approach aims to more effectively allocate computational resources, thereby enhancing the training dynamics of our INR model. This strategy is particularly suited to handling the inherent sparsity in TPC data and aligns with the goal of accurately capturing the significant physical phenomena represented by non-zero values.

Entropy-based Sampling. We approximate $p(y)$ using a histogram $P(y)$, where $P(x_i)$ represents the count of data points near the value x_i. The importance function is then computed as:

$$\text{IF}(x_i) \propto \frac{C}{P(x_i)},$$

where C is a proportionality constant chosen such that $\text{IF}(x_i) \cdot P(x_i) = C$ across all bins. This approach results in a new histogram $P_{\text{Samp}}(x_i)$ that is as uniform as possible given the constraints of the dataset.

For a given sampling ratio ρ and a dataset containing N data points, let B represent the number of histogram bins. The constant C is determined by:

$$C = \frac{N \cdot \rho}{B}.$$

If C is smaller than the smallest count across all bins in $P(y)$, the algorithm samples C points from each bin. Otherwise, the sampling adjusts to allocate the deficit among bins with counts exceeding C, ensuring that the overall distribution remains as uniform as possible.

This entropy-based sampling framework not only prioritizes rare values but also maximizes the information content of the sampled data, making it particularly well-suited for INR training where the goal is to capture complex and subtle features within scientific datasets.

D Results

D.1 Task 1 Additional Results

In this section, we demonstrate the influence of sub-sampling on super-resolution accuracy. In Figure 4 to 6, we show the reconstruction with input from sub-sampling $192 \times 249 \times 16$ ($S = 4$), $96 \times 125 \times 16$ ($S = 4$), and $48 \times 63 \times 16$ ($S = 16$). All the reconstructions are then evaluated on full resolution $192 \times 249 \times 16$ with L_1 errors listed on the differences. We can see that the reconstruction quality decreases as S increases.

Since the TPC data has the lowest resolution in the layer dimension r, sub-sampling along this dimension affect the reconstruction quality in the most obvious way. As we can see by comparing Figure 6 and 7, because the $S = 8$ reconstruction sub-sample the layer dimension, it quality is lower than the $S = 16$ super-resolution.

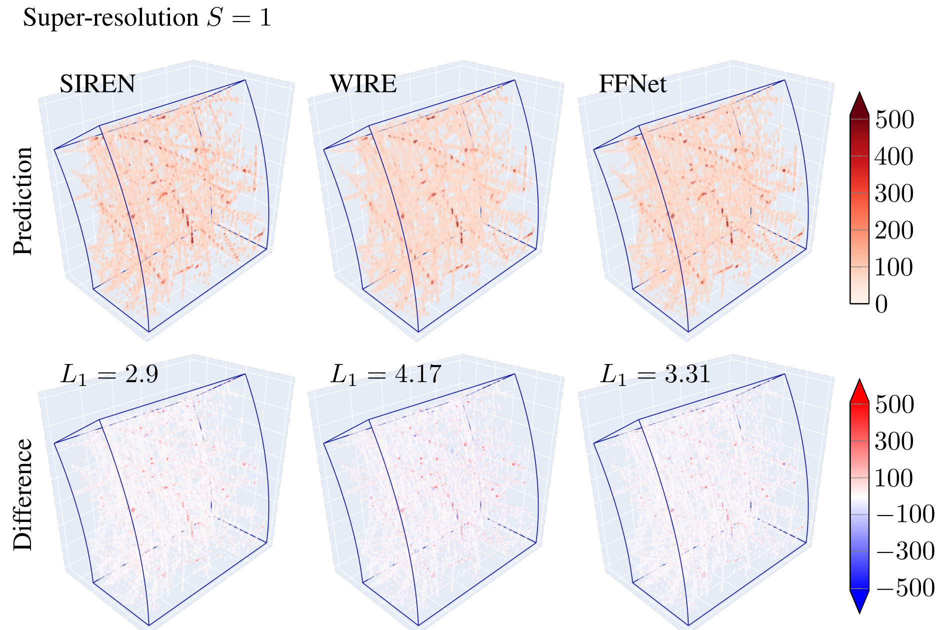

Figure 4: Qualitative results of continuous reconstruction with super-resolution scales of $\times 1$. All INR models were trained on data with dimensions $192 \times 249 \times 16$ and evaluated on datasets of the same size.

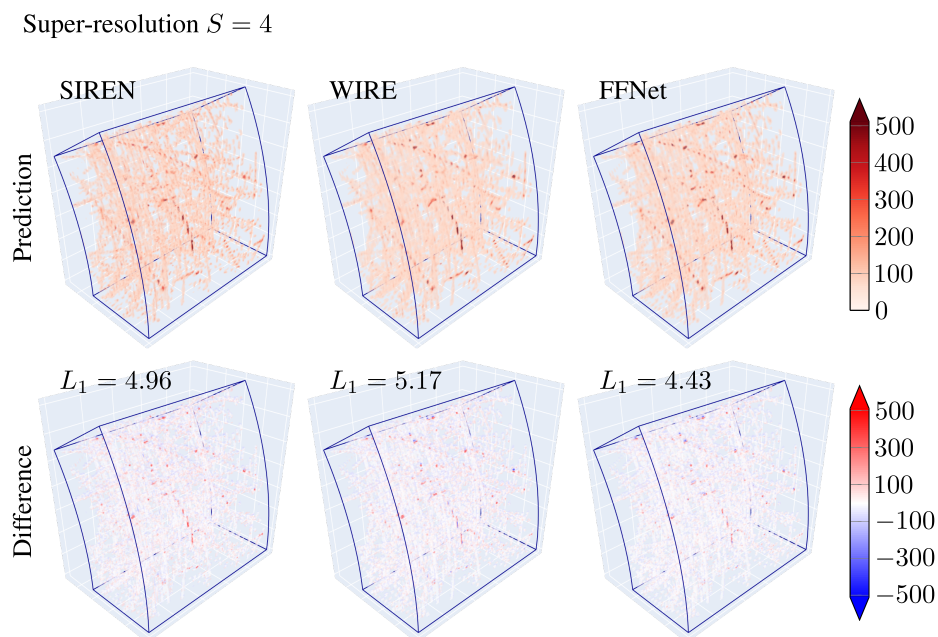

Figure 5: Qualitative results of continuous reconstruction with super-resolution scales of $\times 4$. All INR models were trained on data with dimensions $96 \times 125 \times 16$ and evaluated on datasets with dimensions $192 \times 249 \times 16$.

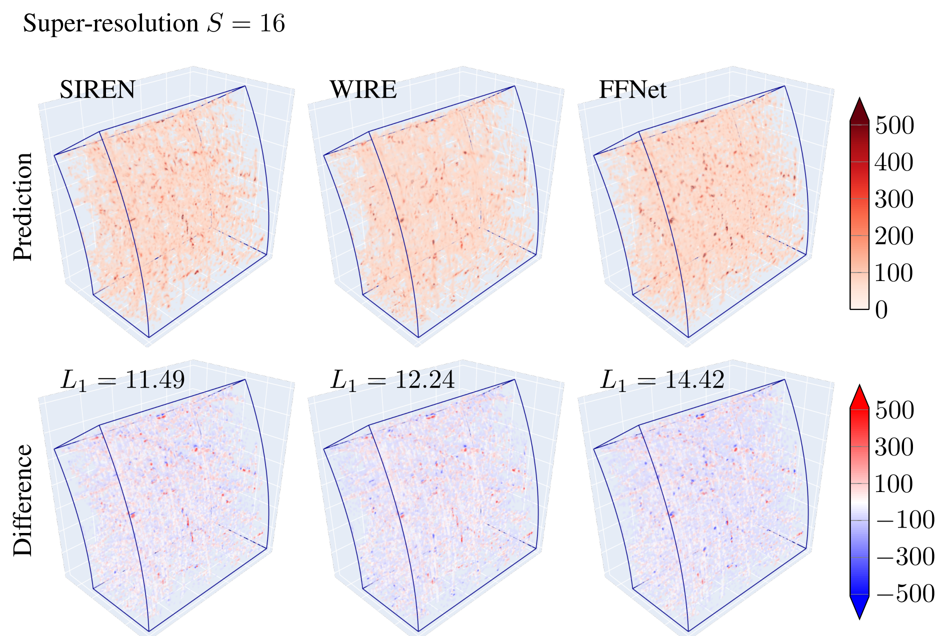

Figure 6: Qualitative results of continuous reconstruction with super-resolution scales of $\times 16$. All INR models were trained on data with dimensions $48 \times 63 \times 16$ and evaluated on datasets with dimensions $192 \times 249 \times 16$.

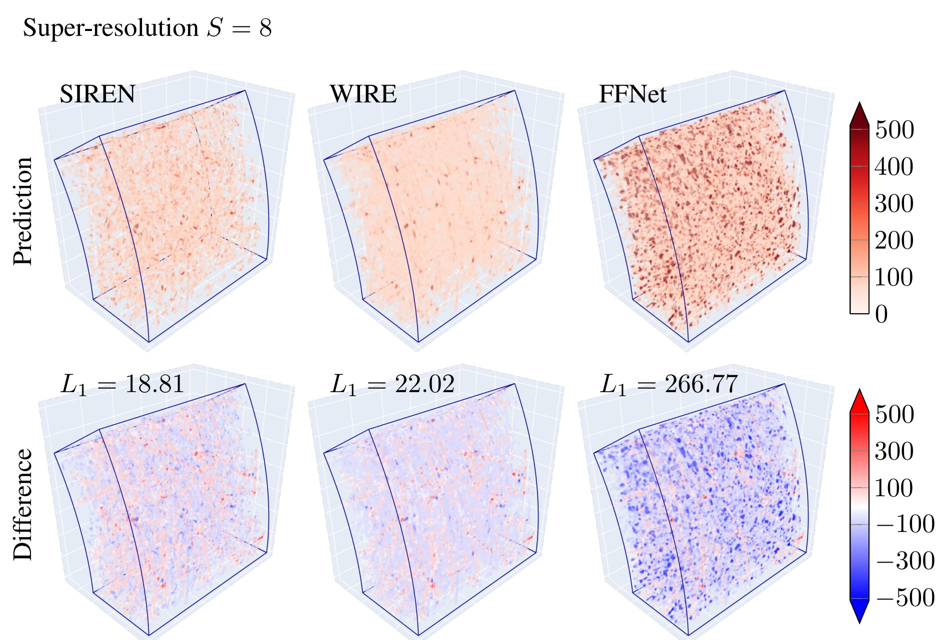

Figure 7: Qualitative results of continuous reconstruction with super-resolution scales of $\times 8$. All INR models were trained on data with dimensions $96 \times 125 \times 8$ and evaluated on datasets with dimensions $192 \times 249 \times 16$. This is only the super-resolution that sub-sample the layer dimension. Since the layer dimension has the lowest resolution in TPc data, sub-sampling along this dimension affect the super-resolution accuracy significantly.