FCNR: Fast Compressive Neural Representation of Visualization Images

Yunfei Lu* Pengfei Gu† Chaoli Wang‡

University of Notre Dame

Abstract

We present FCNR, a fast compressive neural representation for tens of thousands of visualization images under varying viewpoints and timesteps. The existing NeRVI solution, albeit enjoying a high compression ratio, incurs slow speeds in encoding and decoding. Built on the recent advances in stereo image compression, FCNR assimilates stereo context modules and joint context transfer modules to compress image pairs. Our solution significantly improves encoding and decoding speed while maintaining high reconstruction quality and satisfying compression ratio. To demonstrate its effectiveness, we compare FCNR with state-of-the-art neural compression methods, including E-NeRV, HNeRV, NeRVI, and ECSIC. The source code can be found at `https://github.com/YunfeiLu0112/FCNR`.

1 Introduction

Generating vast datasets has become indispensable for analyzing complex phenomena across diverse fields. As technical advances continue to shine in an era of unprecedented data acquisition, managing and interpreting such enormous amounts of data becomes increasingly challenging. Scientific visualization is important for us to visually comprehend complicated patterns, identify trends, and extract meaningful insights from simulation data, which are often time-varying. When dealing with a time-varying volumetric dataset, we can produce numerous visualization images, including those generated through isosurface rendering (IR) or direct volume rendering (DVR). These images correspond to various parameters, such as timesteps and camera views, offering a thorough representation of the data. We tackle the issue of managing significant quantities of these visualization images, which occupy substantial gigabytes of storage and could surpass the original data size. This scenario poses significant constraints, including storage cost, network transmission, image access, and interactive display when conveying the visualization output. Hence, efficient compression and sharing of these visualization images become a necessity. Recent developments in DL4SciVis [20] provide a viable direction.

To achieve this goal, we present FCNR, a **f**ast **c**ompressive **n**eural **r**epresentation for tens of thousands of visualization images. FCNR takes a pair of visualization images with nearby views as input. It encodes the image pair into quantized bitstreams with entropy coding computed and their similarity exploited. It then decodes them to reconstruct the images. FCNR can efficiently compress a vast array of images derived from time-varying datasets under different viewing and timestep parameters in a relatively short time. We evaluate FCNR on multiple datasets, quantitatively and qualitatively, and compare it with state-of-the-art deep learning compression baselines, including E-NeRV, HNeRV, NeRVI, and ECSIC, to demonstrate its superiority. Our contributions are as follows:

- simultaneously compressing a pair of images with similar views based on *joint context transfer modules* (JCTMs), which extract mutual information from the whole images;
- incorporating viewpoints and timesteps into *stereo context modules* (SCMs) to accommodate our compression scenario while improving entropy estimation of one encoded image using another as the context;
- improving encoding and decoding speeds significantly while maintaining high reconstruction quality and satisfying compression ratio expressed in *bit per pixel* (BPP);
- leveraging the interpolation ability by compressing all the images of the given dataset after training on its subset, which also expedites the encoding process.

2 Related Work

In recent years, implicit neural representation (INR) has been extensively studied for image and video compression [3–5, 11, 12, 14, 25]. NeRV [5] takes as input an image index, generates the image embedding via multilayer perceptrons (MLPs) and convolutional layers, and outputs the whole image. E-NeRV [12] improves the NeRV architecture by identifying the redundant parts and decomposing the image-wise INR into distinct spatial and temporal contexts, accelerating convergence while maintaining high performance. CNeRV [3] enables internal generalization using content-adaptive embedding, which compactly encodes visual information. D-NeRV [25] represents various videos using the same model, which takes sampled key-frames as input for clip-specific content encoding and outputs video frames with a motion-aware decoder. HNeRV [4] resolves the content-agnostic issue and the unbalanced parameter distribution of NeRV by storing videos in small, content-adaptive frame embeddings and utilizing a learned decoder. NIRVANA [14] proposes patch-wise prediction to accommodate videos with varying spatial and temporal resolutions using the same architecture. HiNeRV [11] addresses the limited representation capability of INR and refines the model compression pipeline with adaptive parameter weighting and quantization-aware training.

In scientific visualization, INR has been applied to data generation and compression tasks [9, 18, 19]. Gu et al. [8] extended INR to visualization image compression, which is much more challenging due to the need for accommodating viewpoint and timestep parameters and more significant differences between neighboring rendering images than video frames. The proposed NeRVI achieves neural representations with a high compression ratio and leads to good image fidelity using mask loss. However, the rather slow encoding speed restricts its use in practice, especially when compressing high-resolution visualization images in a large collection.

In contrast, stereo-image compression methods seek to compress image pairs simultaneously by exploiting their similarities (i.e., mutual information) using neural networks. For instance, Liu et al. [13] presented DSIC, which computes a dense warp field and feeds features from the left image after warping into the encoder and decoder of the right image. Deng et al. [7] designed HESIC, which improves DSIC by applying a rigid image-space homography transform. Wödlinger et al. [22] introduced SASIC that enhances a conventional single-image compression backbone model. To accommodate finer local displacements between images, it incorporates latent-domain global shift and subtraction as well as stereo attention modules in the decoder. Based on SASIC, Wödlinger et al. [21] further developed ECSIC that augments the architecture with *stereo*

[*] e-mail: ylu25@nd.edu

[†] e-mail: pgu@nd.edu

[‡] e-mail: chaoli.wang@nd.edu

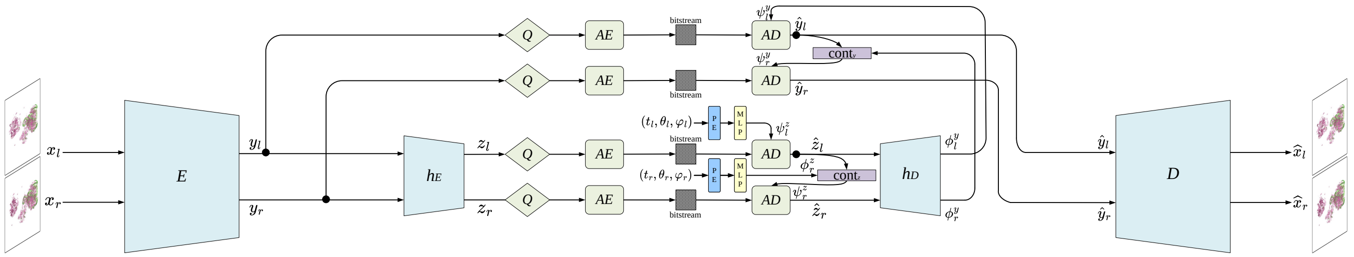
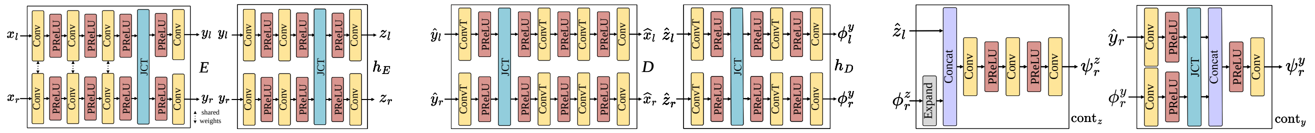


Figure 1: Overview of FCNR. The encoder (E) encodes x_l and x_r to bitstreams with hyper-encoder (h_E), quantization (Q), and arithmetic encoder (AE). The decoder (D) then reconstructs $\hat{x}_l$ and $\hat{x}_r$ through quantized latents ($\hat{y}_l$ and $\hat{y}_r$) with arithmetic decoder (AD) and hyper-decoder (h_D).



(a) encoder (E) and hyper-encoder (h_E) (b) decoder (D) and hyper-decoder (h_D) (c) SCM (cont$_z$) and SCM (cont$_y$)

Figure 2: The detailed structure of each module.

cross-attention modules (SCAMs) and SCMs. Zhang et al. [24] proposed LDMIC, a simple and effective cross-attention-based JCTM utilizing the decoder's cross-attention mechanism to capture global inter-view correlations efficiently. In this work, we assimilate the notion of jointly compressing image pairs by exploiting their mutual information. We design our network based on ECSIC, incorporate the JCTM from LDMIC for the more complicated visualization image compression task, and present FCNR, a fast solution for compressive neural representation.

3 FCNR

Given a time-varying dataset $Y = \{Y_1, Y_2, \ldots, Y_T\}$, where T is the number of timesteps, with a predefined isovalue or transfer function, we produce a set of visualization images. Each volume Y_t, $t \in [1, T]$, is represented as a subset of images X_t associated with different camera views. We aim to learn a mapping that encodes the input images into latent representations, which are quantized, compressed, and decoded to reconstruct the images.

Networks for stereo image compression often consist of the main autoencoder and hyperprior autoencoder. We adapt this structure to compress IR or DVR images under different views and timesteps. As shown in Figure 1, the input to our model is a pair of images x_l and x_r (l and r denote left and right) from X_t, with the two neighboring views (θ, φ_l) and (θ, φ_r). We first use the encoder E to encode x_l and x_r into the latents y_l and y_r. Then, we estimate the latent entropy parameters ψ_l^y and ψ_r^y using the hyper-encoder h_E and hyper-decoder h_D to produce quantized latents $\hat{y}_l$ and $\hat{y}_r$. Finally, we utilize the decoder D to reconstruct the images $\hat{x}_l$ and $\hat{x}_r$ from $\hat{y}_l$ and $\hat{y}_r$. We improve the structure with SCMs [21] (which aid in the prediction of ψ_r^y from $\hat{y}_l$ and hyper-latent entropy parameters ψ_r^z from hyper-latent $\hat{z}_l$) and JCTMs [24] (which exploits the feature-space inter-view correlations brought by overlapping viewpoints in visualization images for generating more informative representations).

Encoding modules and quantization. We develop our E and h_E based on the structure proposed in ECSIC. Each consists of several convolutional (Conv) layers, a JCTM, and parametric ReLU (PReLU) activation functions [10]. Their detailed structures are shown in Figure 2 (a). Unlike the epipolar assumption in ECSIC, the transformations between two visualization images in a pair are much more complex, involving the 3D rotation of the volume. Since SCAM [21] only computes cross-attention between the corresponding epipolar lines, it fails to fully capture the inter-view information brought by the more complicated transformations. Therefore, we replace SCAMs with JCTMs to exploit mutual information globally. Given x_l and x_r, E computes the main latent representations y_l and y_r by

$$y_i = E(x_i),\ i \in \{l, r\}. \tag{1}$$

h_E accepts y_l and y_r and generates the hyper-latents z_l and z_r

$$z_i = h_E(x_i),\ i \in \{l, r\}. \tag{2}$$

Introducing z for entropy estimation is effective to model the dependencies between y, which are assumed to be independently conditioned on z [2]. y_l, y_r, z_l, and z_r then go through a quantization process. For example,

$$\hat{y}_i = \text{round}(y_i - \mu_i^y) + \mu_i^y,\ i \in \{l, r\}, \tag{3}$$

where μ_i^y is the estimated mean of the disrtibution of y_i. A similar process is applied to z_l and z_r to generate quantized hyper-latents $\hat{z}_l$ and $\hat{z}_r$. To make the process differentiable, we use approximate quantization by adding the uniform noise $\varepsilon \sim \mathscr{U}(-0.5, 0.5)$ for the rate loss [1].

$$\tilde{y}_i = y_i + \varepsilon,\ i \in \{l, r\}. \tag{4}$$

Thus, the density function of $\hat{y}_i$ is a continuous relaxation of the probability mass function of y_i, allowing the differential entropy of $\hat{y}_i$ to approximate the entropy of $\hat{y}$. Additionally, independent uniform noise approximates the quantization error, modeling its marginal moments for distortion measurement. z_l and z_r follow a similar process. For the distortion loss, we employ a straight-through-estimation quantization [15].

Decoding modules and entropy model. As shown in Figure 2 (b), D and h_D each have Conv layers, a JCTM, PReLU activation functions, and transposed convolutional (ConvT) layers for upsampling. $\hat{z}_l$ and $\hat{z}_r$ are stored as side information to help predict ψ_l^y and ψ_r^y of $\hat{y}_l$ and $\hat{y}_r$. The distribution of each latent representation is modeled by a Laplacian distribution with parameters $\psi = (\mu, b)$. The process of distribution modeling and parameter estimation corresponds to the arithmetic encoders (AEs) and arithmetic decoders (ADs) shown in Figure 1. We model the distribution of $\hat{z}_l$ by a channel-wise Laplacian distribution with parameters ψ_l^z computed from visualization parameters $(t_l, \theta_l, \varphi_l)$

$$\psi_l^z = \text{MLP}(\text{PE}(t_l, \theta_l, \varphi_l)), \tag{5}$$

where PE denotes positional encoding which projects $(t_l, \theta_l, \varphi_l)$ into a higher-dimensional space

$$\text{PE}(u) = (\sin(b^0 \pi u), \cos(b^0 \pi u), \ldots, \sin(b^{L-1} \pi u), \cos(b^{L-1} \pi u)), \tag{6}$$

$$\mathrm{PE}(t,\theta,\varphi) = (\mathrm{PE}(t), \mathrm{PE}(\theta), \mathrm{PE}(\varphi)). \tag{7}$$

Here, we set $b = 1.25$ and $L = 8$. Since our visualization image pairs have greater variations than the stereo image pairs compressed by ECSIC, computing distribution parameters from visualization parameters can help mitigate the gap by allowing for detailed quantitative differences between images and improve model performance. The distributions of $\hat{z}_r$, $\hat{y}_l$, and $\hat{y}_r$ are modeled by factorized Laplacian distributions. We further reduce the bitrate by conditioning the distributions of $\hat{y}_r$ and $\hat{z}_r$ on the information from $\hat{y}_l$ and $\hat{z}_l$ with SCMs cont_y and cont_z, and their structures are shown in Figure 2 (c). In this way, ψ_r^z are predicted from $\hat{z}_l$ and parameters ϕ_r^z

$$\psi_r^z = \mathrm{cont}_z(\hat{z}_l, \phi_r^z), \tag{8}$$

where, likewise, ϕ_r^z is generated with $(t_r, \theta_r, \varphi_r)$

$$\phi_r^z = \mathrm{MLP}(\mathrm{PE}(t_r, \theta_r, \varphi_r)), \tag{9}$$

Then h_D computes ϕ_l^y and ϕ_r^y from quantized hyper-latents

$$\phi_i^y = h_D(\hat{z}_i),\ i \in \{l, r\}. \tag{10}$$

In the main branch, we set $\psi_l^y = \phi_l^y$. For ψ_r^y, we similarly apply the SCM

$$\psi_r^y = \mathrm{cont}_y(\hat{y}_l, \phi_r^y). \tag{11}$$

Finally, D reconstructs $\hat{x}_l$ and $\hat{x}_r$ from $\hat{y}_l$ and $\hat{y}_r$

$$\hat{x}_i = D(\hat{y}_i),\ i \in \{l, r\}. \tag{12}$$

Loss functions. Image compression models can be optimized for a weighted sum of the rate and distortion losses [1]

$$L_{\text{total}} = L_R + \lambda L_D, \tag{13}$$

where L_R is the rate loss, L_D is the distortion loss, and $\lambda \in \mathbb{R}$ is a trade-off weight. L_D is the expectation of the mean squared errors between the input and reconstructed images

$$L_D = \mathbb{E}_{x_l, x_r \sim p_x}\left[\|x_l - \hat{x}_l\|_2^2 + \|x_r - \hat{x}_r\|_2^2\right]. \tag{14}$$

Following the rate loss in ECSIC, L_R is the expected sum of the cross entropy between the predicted distribution of our entropy model and the true distribution of the latents or hyper-latents

$$\begin{aligned} L_R = \mathbb{E}_{x_l, x_r \sim p_x}\big[& -\log_2 p(\hat{z}_l \mid \phi_l^z) \\ & -\log_2 p(\hat{z}_r \mid \Phi_{\mathrm{cont}_z}, \phi_r^z, \hat{z}_l) \\ & -\log_2 p(\hat{y}_l \mid \Phi_{h_D}, \hat{z}_r, \hat{z}_l) \\ & -\log_2 p(\hat{y}_r \mid \Phi_{\mathrm{cont}_y}, \Phi_{h_D}, \hat{y}_l, \hat{z}_r, \hat{z}_l)\big], \end{aligned} \tag{15}$$

where $\Phi_{\mathrm{cont}_z}, \Phi_{\mathrm{cont}_y}, \Phi_{h_D}$ denote the parameters of cont_z, cont_y, and h_D, respectively.

4 Results and Discussion

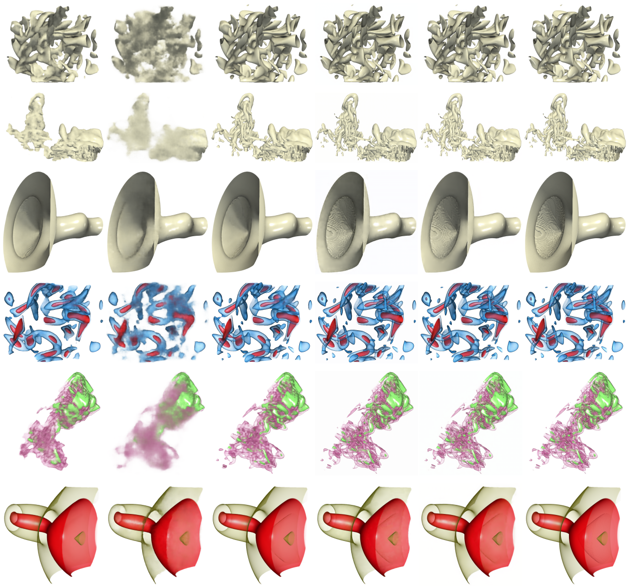
Datasets, setting, and training. Table 1 shows the three datasets we experimented with. We picked 30 consecutive timesteps for each dataset. To ensure an even distribution of viewpoints across the volume data, we determined camera positions using the vertices of an icosphere, approximating a sphere using equilateral triangles. We selected a subdivision level resulting in 812 vertices to generate the training set for each timestep, producing a good quantity of images for compression. The image resolutions were all set to 1024×1024. For ECSIC and FCNR, which require image pairs, we sorted the images at each timestep first by θ and then by φ if there is a tie. We then chose the image with an even index j (starting from 0) as the left image and the image with index $j+1$ as its right counterpart. Since ECSIC and FCNR possess generalization ability, we decreased the number of training images to $1/6$ (evenly selected $1/2$ of the sampled views and $1/3$ of the timesteps), i.e., 4060 images. We evaluated these two models on all 24360 images during inference.

Table 1: The resolution and total sampled images of each dataset. "# st" denotes the number of timesteps we subsample from the dataset.

dataset	resolution ($x \times y \times z \times t$)	# views	# st	# images
vortex [17]	$128 \times 128 \times 128 \times 90$	812	30	24360
Tangaroa [16]	$300 \times 180 \times 120 \times 150$	812	30	24360
tornado [6]	$128 \times 128 \times 128 \times 48$	812	30	24360

Table 2: Average PSNR (dB) and LPIPS, BPP, and total ET (hours) DT (seconds). Each case has 24360 images with a resolution of 1024×1024. The best ones are highlighted in bold.

		IR images					DVR images				
dataset	method	PSNR↑	LPIPS↓	BPP↓	ET↓	DT↓	PSNR↑	LPIPS↓	BPP↓	ET↓	DT↓
vortex	E-NeRV	27.17	0.1432	0.0031	149.46	1925.47	21.77	0.1154	0.0031	151.58	1939.19
	HNeRV	21.15	0.2701	**0.0019**	72.72	391.25	20.17	0.2022	**0.0019**	69.63	916.33
	NeRVI	26.80	0.1386	0.0356	251.46	3965.49	24.30	0.0602	0.0356	255.56	2982.23
	ECSIC	36.27	**0.0980**	0.0915	1.20	**259.90**	34.77	0.0139	0.1437	1.20	**250.66**
	FCNR	**37.47**	0.1025	0.0693	**1.18**	269.67	**34.85**	**0.0132**	0.1212	**1.19**	296.70
Tangaroa	E-NeRV	25.96	0.0093	0.0031	147.52	4362.98	25.43	0.1103	0.0031	147.13	4203.47
	HNeRV	23.91	0.1690	**0.0015**	57.07	4872.01	24.17	0.1759	**0.0015**	71.50	1239.88
	NeRVI	28.16	0.0750	0.0356	181.13	2015.16	26.39	0.0964	0.0359	247.33	2137.02
	ECSIC	37.82	0.0149	0.0895	**1.18**	**306.94**	**34.61**	**0.0153**	0.1405	1.20	246.52
	FCNR	**38.12**	**0.0145**	0.0709	**1.18**	319.84	34.45	0.0177	0.1109	**1.17**	**211.73**
tornado	E-NeRV	36.72	0.1389	0.0031	50.14	4643.82	35.09	0.0038	0.0031	50.34	5157.63
	HNeRV	34.53	0.0544	**0.0015**	28.54	5359.20	31.97	0.0506	**0.0016**	30.75	1804.20
	NeRVI	**38.21**	**0.0359**	0.0356	90.56	9609.40	36.27	0.0336	0.0356	88.99	2030.20
	ECSIC	36.30	0.0700	0.0580	0.40	**231.71**	36.51	0.0501	0.0982	0.42	**180.17**
	FCNR	38.07	0.0685	0.0280	**0.39**	326.91	**37.35**	**0.0386**	0.0359	**0.39**	352.00



(a) E-NeRV (b) HNeRV (c) NeRVI (d) ECSIC (e) FCNR (f) GT

Figure 3: Decompressed IR and DVR images. The datasets are vortex, Tangaroa, and tornado, respectively.

We implemented FCNR using PyTorch. We chose the number of channels in the encoding and decoding modules to be 192, the number of latent channels to be 48, and the number of attention heads in JCTM to be 2. All experiments were run on an NVIDIA A40 GPU. Adam optimizer was utilized for gradient descent (β_1=0.9, β_2=0.999), and the learning rate was set as 10^{-4}. We trained FCNR with a batch size of 1. The number of training epochs was 3 for the vortex and Tangaroa datasets and 1 for the tornado dataset.

Baselines and evaluation metrics. We compared FCNR with three state-of-the-art INR-based methods, including E-NeRV [12], HNeRV [4], and NeRVI [8], and one stereo image compression method, ECSIC [21]. We extended E-NeRV by feeding all (t, θ, φ) to the model, first normalized to $[0, 1]$ and then input to the network after PE. All INR-based methods were trained until convergence for a fair comparison. The number of training epochs for ECSIC was the same as FCNR for each dataset. For quantitative evaluation in the image space, we employed peak signal-to-noise ratio (PSNR) and learned perceptual image patch similarity (LPIPS) [23]. Besides, encoding time (ET) and decoding time (DT) are also recorded.

Table 3: Average PSNR (dB) and LPIPS, BPP, and total ET (hours) DT (seconds) of ECSIC, its varying modifications, and FCNR on the tornado IR dataset.

method	PSNR↑	LPIPS↓	BPP↓	ET↓	DT↓
ECSIC	36.30	0.0700	0.0580	0.40	231.71
JCT-Only	37.81	**0.0685**	0.0666	**0.39**	270.15
PE-Only	37.19	0.0708	0.0370	0.40	**228.91**
FCNR	**38.07**	**0.0685**	**0.0280**	**0.39**	326.91

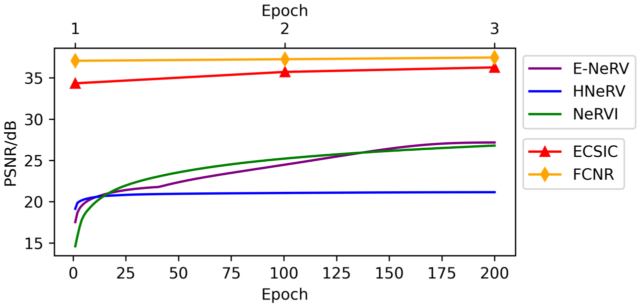


Figure 4: PSNR comparison of all methods on the vortex IR dataset. E-NeRV, HNeRV, and NeRVI were all trained for 200 epochs. Both ECSIC and FCNR were trained for 3 epochs.

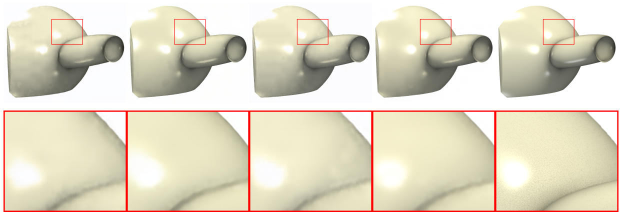
Results. Table 2 compares FCNR with state-of-the-art baselines quantitatively. Figure 3 shows the decompressed rendering images for all datasets under chosen (t,θ,φ). All images are cropped for closer comparison.

FCNR achieves the highest PSNR and lowest LPIPS for most cases, even on images unseen during training, demonstrating its interpolation ability. Although INR-based methods (E-NeRV, HNeRV, and NeRVI) lead to lower BPP, they have limited reconstruction quality as depicted in Figure 3. The vortex and Tangaroa datasets vary greatly at different timesteps, posing greater reconstruction challenges. HNeRV produces the most blurry images. The results of E-NeRV and NeRVI are much clearer. However, they still yield artifacts and distortions when multiple components overlap in rendering images and miss some components when images become complex. For the tornado dataset, all methods generate clear images, but FCNR generates images with better PSNR and LPIPS than E-NeRV and HNeRV. Though NeRVI achieves the highest PSNR and the lowest LPIPS in the tornado IR dataset, it fails to reconstruct high-frequency details as shown in Figure 3. By contrast, FCNR generates the best visual quality images, with the clearest high-frequency details and the fewest artifacts.

Moreover, the significant encoding time (48.36× to 232.21×) and decoding time (1.45× to 29.39×) over FCNR put INR-based baselines at a disadvantage. This is because E-NeRV and NeRVI need substantial training and more parameters to restore the lost information and reconstruct images from the rather limited and low-dimensional input and the necessity of learning both input embedding and decoder weights in HNeRV leads to a more complex design. In contrast, like ECSIC, FCNR drastically enhances encoding and decoding speed by fully utilizing the images for direct compression and reconstruction and exploiting mutual information between them. Compared with ECSIC, FCNR achieves very close, high-quality results for all datasets, with slight improvements in PSNR and LPIPS for the majority of cases. While ECSIC performs similarly to FCNR in image quality and encoding and decoding speed, its BPP is from 18.56% (vortex DVR dataset) to 173.54% (tornado DVR dataset) higher than FCNR across all cases. Such improvements make FCNR stand out from ECSIC.

Figure 4 compares FCNR and baseline methods in PSNR during training on the vortex IR dataset. It shows that all methods have been trained until convergence to ensure a fair comparison. The comparison highlights the effectiveness and efficiency of FCNR.

Ablation study. We performed an ablation study on the tornado IR dataset to show FCNR's differences from ECSIC. We compared FCNR and ECSIC with two architectural modifications: JCT-only and PE-only. For the JCT-only case, we modified ECSIC by changing all SCAMs to JCTMs. For the PE-only case, we extended ECSIC by transforming (t_l,θ_l,φ_l) to the input ψ_l^z of x_l's AD and (t_r,θ_r,φ_r) to the corresponding parameter ϕ_r^z of cont$_z$ through PE and MLP.



(a) ECSIC (b) JCT-only (c) PE-only (d) FCNR (e) GT

Figure 5: First row: decompressed images of FCNR and ECSIC with variations. Second row: zoom-ins for closer examination.

Table 3 shows that both modifications lead to improvements in image quality measured by PSNR (JCT-only and PE-only) and LPIPS (JCT-only). Moreover, though JCT-only leads to higher BPP than ECSIC, PE-only lowers BPP. By incorporating both modifications, FCNR achieves even lower BPP with the best PSNR and LPIPS. Figure 5 demonstrates visual improvements in image quality. JCT-only yields images with smoother surfaces and more natural lighting, and PE-only reduces visual artifacts to some extent and enhances lighting concentration. As the zoom-ins indicate, FCNR further improves border clearness, lighting concentration, and color consistency, generating an image closest to GT.

Discussion. Our results demonstrate that FCNR can compress a large collection of visualization images in high fidelity within a short time. It is much more promising than INR-based methods when image quality and encoding and decoding speed are of greater importance. Though ECSIC can achieve high-quality compression in a similar timeframe, FCNR leads to a higher compression ratio.

5 Conclusions and Future Work

We present FCNR, a novel method for neural compression of visualization images borrowing insights from stereo image compression frameworks. The model of ECSIC reduces the bitrate with distributions of the right image learned from the left image using SCMs. We integrate this model with JCTMs to extract mutual information globally and incorporate visualization parameters to allow for more detailed quantitative differences between images, further improving image quality and compression ratio. Compared with state-of-the-art INR-based methods, FCNR provides previously unavailable interpolation ability and demonstrates improved encoding and decoding time. Compared with ECSIC, FCNR achieves a higher compression ratio and slightly better reconstruction quality.

The future work of FCNR can be summarized as follows. First, given the substantial differences between stereo images and visualization images, designing a more tailored model architecture for visualization images is necessary for further gains in quality and speed. Second, FCNR lags behind INR-based methods in terms of compression ratio. Our method will be more promising if BPP can be reduced to the same level as INR-based methods. Finally, more visualization parameters, such as isovalues and transfer functions, may be included, and a better fusion of these parameters with images is worthy of exploration.

Acknowledgments

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

References

[1] J. Ballé, V. Laparra, and E. P. Simoncelli. End-to-end optimized image compression. In *Proceedings of International Conference on Learning Representations*, 2017. 2, 3

[2] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston. Variational image compression with a scale hyperprior. In *Proceedings of International Conference on Learning Representations*, 2018. 2

[3] H. Chen, M. Gwilliam, B. He, S.-N. Lim, and A. Shrivastava. CNeRV: Content-adaptive neural representation for visual data. In *Proceedings of British Machine Vision Conference*, pp. 510:1–510:20, 2022. 1

[4] H. Chen, M. Gwilliam, S.-N. Lim, and A. Shrivastava. HNeRV: A hybrid neural representation for videos. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10270–10279, 2023. doi: 10.1109/CVPR52729.2023.00990 1, 3

[5] H. Chen, B. He, H. Wang, Y. Ren, S.-N. Lim, and A. Shrivastava. NeRV: Neural representations for videos. In *Proceedings of Advances in Neural Information Processing Systems*, pp. 21557–21568, 2021. 1

[6] R. A. Crawfis and N. Max. Texture splats for 3D scalar and vector field visualization. In *Proceedings of IEEE Visualization Conference*, pp. 261–267, 1993. doi: 10.1109/VISUAL.1993.398877 3

[7] X. Deng, W. Yang, R. Yang, M. Xu, E. Liu, Q. Feng, and R. Timofte. Deep homography for efficient stereo image compression. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1492–1501, 2021. doi: 10.1109/CVPR46437.2021.00154 1

[8] P. Gu, D. Z. Chen, and C. Wang. NeRVI: Compressive neural representation of visualization images for communicating volume visualization results. *Computers & Graphics*, 116:216–227, 2023. doi: 10.1016/J.CAG.2023.08.024 1, 3

[9] 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

[10] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *Proceedings of IEEE International Conference on Computer Vision*, pp. 1026–1034, 2015. doi: 10.1109/ICCV.2015.123 2

[11] H. M. Kwan, G. Gao, F. Zhang, A. Gower, and D. Bull. HiNeRV: Video compression with hierarchical encoding based neural representation. In *Proceedings of Advances in Neural Information Processing Systems*, 2023. 1

[12] Z. Li, M. Wang, H. Pi, K. Xu, J. Mei, and Y. Liu. E-NeRV: Expedite neural video representation with disentangled spatial-temporal context. In *Proceedings of European Conference on Computer Vision*, pp. 267–284, 2022. doi: 10.1007/978-3-031-19833-5_16 1, 3

[13] J. Liu, S. Wang, and R. Urtasun. DSIC: Deep stereo image compression. In *Proceedings of IEEE International Conference on Computer Vision*, pp. 3136–3145, 2019. doi: 10.1109/ICCV.2019.00323 1

[14] S. R. Maiya, S. Girish, M. Ehrlich, H. Wang, K. S. Lee, P. Poirson, P. Wu, C. Wang, and A. Shrivastava. NIRVANA: Neural implicit representations of videos with adaptive networks and autoregressive patch-wise modeling. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 14378–14387, 2023. doi: 10.1109/CVPR52729.2023.01382 1

[15] D. Minnen and S. Singh. Channel-wise autoregressive entropy models for learned image compression. In *Proceedings of IEEE International Conference on Image Processing*, pp. 3339–3343, 2020. doi: 10.1109/ICIP40778.2020.9190935 2

[16] 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 3

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

[18] 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

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

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

[21] M. Wödlinger, J. Kotera, M. Keglevic, J. Xu, and R. Sablatnig. ECSIC: Epipolar cross attention for stereo image compression. In *Proceedings of IEEE Winter Conference on Applications of Computer Vision*, pp. 3436–3445, 2024. 1, 2, 3

[22] M. Wödlinger, J. Kotera, J. Xu, and R. Sablatnig. SASIC: Stereo image compression with latent shifts and stereo attention. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 661–670, 2022. doi: 10.1109/CVPR52688.2022.00074 1

[23] 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 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018. doi: 10.1109/CVPR.2018.00068 4

[24] X. Zhang, J. Shao, and J. Zhang. LDMIC: Learning-based distributed multi-view image coding. In *Proceedings of International Conference on Learning Representations*, 2023. 2

[25] Q. Zhao, M. S. Asif, and Z. Ma. DNeRV: Modeling inherent dynamics via difference neural representation for videos. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2031–2040, 2023. doi: 10.1109/CVPR52729.2023.00202 1