Few-shot Implicit Function Generation via Equivariance

Supplementary Material

1. Detailed Problem Formalization

Following the definition we gave in the *Preliminary*, we now formulate a more theoretical and official definition.

We aim to generate diverse neural network weights that encode different instances within the same signal class (*e.g.*, different styles of digit "7", different shapes of planes) as implicit neural representations, given only a few example weights as reference.

Let $\mathcal{W} \subseteq \mathbb{R}^d$ denote the weight space of MLPs with a fixed architecture, where d is the total number of parameters when all weights are flattened into a single vector. Consider a class space $\mathcal{F}$ where each $f \in \mathcal{F}$ represents a class of signals (*e.g.*, a specific digit, a category of shapes). Let $\Phi : \mathcal{W} \to (\mathcal{X} \to \mathcal{Y})$ map weights to their corresponding implicit functions, where $\mathcal{X}$ is the coordinate space (e.g., 2D/3D coordinates) and $\mathcal{Y}$ is the target space (e.g., RGB values, occupancy).

Definition 1 (Class-Induced Weight Distribution)**.** *For any class $f \in \mathcal{F}$ with corresponding coordinate space $\mathcal{X}$, we define its corresponding weight distribution as:*

$$p_f(w) := P(w|d(\Phi(w), f) \leq \epsilon) \tag{1}$$

where validity is measured by a distance metric:

$$d(\Phi(w), f) = \mathbb{E}_{x\sim\mathcal{X}} \left[|\Phi(w)(x) - y_f(x)|_2^2\right]^{1/2}, \tag{2}$$

where $y_f(x)$ represents the target values for class f at coordinate x, and $\epsilon > 0$ is a tolerance parameter that determines the acceptable variation within the class.

Definition 2 (Few-shot Weight Generation)**.** *Given:*

- *A set of training classes $\{f_1, ..., f_N\} \subset \mathcal{F}$ (e.g., different digits, different shape categories) with their corresponding weight distributions $\{p_{f_1}(w), ..., p_{f_N}(w)\}$*
- *For each class f_i, a support set $S_{f_i} = \{w_1^i, ..., w_k^i\}$ where $w_j^i \overset{i.i.d.}{\sim} p_{f_i}(w)$ represents different MLP weights that encode valid instances of that class*

The Few-shot Class-based INR Weight Generation problem is to learn a generator G that can estimate the weight distribution of any class $f \in \mathcal{F}$ given only its k-shot support set:

$$G : S_f \mapsto \tilde{p}_f(w), \tag{3}$$

such that the generated distribution $\tilde{p}_f(w) \approx p_f(w)$ when $S_f = \{w_1, ..., w_k\} \overset{i.i.d.}{\sim} p_f(w)$. The final objective is to minimize the expected distribution distance over the space of possible test classes:

$$\min_\theta \mathbb{E}_{f\sim\mathcal{F}}[\mathcal{D}(G(S_f), p_f(w))|S_f \overset{i.i.d.}{\sim} p_f(w)]. \tag{4}$$

2. Additional Related Work

2.1. Implicit Neural Representation

Implicit Neural Representations (INRs) have demonstrated remarkable efficacy in representing diverse forms of complex signals, including spatial occupancy [25, 35, 37], 3D geometric morphology [2, 4, 10], signed distance functions [17, 30], 3D scene appearence [5, 16, 21, 26, 42] and some other complex signals [3, 22, 23, 43] with the help of a small neural network, usually a MLP with few layers.

Within this domain, several methodological innovations have emerged as particularly noteworthy. The SIREN architecture employs sinusoidal activation functions to capture high-frequency spatial details with unprecedented fidelity [37]. Subsequent research has introduced Gaussian activation functions, offering enhanced initialization stability and parameter efficiency compared to their predecessors [32]. The WIRE framework represents a significant advancement through its implementation of continuous complex Gabor wavelets, enabling robust and high-precision representation of natural images [35]. Furthermore, LIIF has introduced novel approaches to continuous representation, specifically addressing the challenge of arbitrary resolution representation [3].

2.2. Generative Models

For image generation task, diffusion probabilistic models have revolutionized generative modeling [6, 14, 39], emerging as a powerful alternative to traditional approaches like GANs [11] and energy-based models (EBMs) [7]. Their success in image synthesis stems from their ability to produce higher quality outputs with improved fidelity [33, 39], while also enabling effective text conditioning [28, 34]. A notable advancement came with Latent Diffusion, which achieved efficient high-resolution image synthesis by applying the diffusion process in the latent space of pretrained autoencoders [1, 33].

The extension of diffusion models to 3D domains has followed several paths. Early attempts to directly apply 2D generation techniques to 3D voxel grids encountered computational limitations at higher resolutions due to the inherent complexity of 3D convolution networks [20, 38, 41]. This led researchers to explore alternative 3D representations, including point clouds [44, 47] and implicit fields [4, 24]. A significant breakthrough came with MeshDiffusion [18], which pioneered the unconditional generation of 3D shapes using diffusion models with direct 3D shape supervision.

Recent work has expanded diffusion models beyond tra-

Figure 1. Visualization of the proposed INR-based data augmentation including color jitter (middle) and bias disturbance (right).

ditional data types to address the challenge of generating neural network parameters. HyperNetworks[12] introduced the concept of using one network to learn the parameters of another. Model-Agnostic Meta-Learning (MAML) developed methods for learning parameter initializations that enable efficient fine-tuning [9]. The G.pt model proposed predicting parameter update distributions based on initial parameters and prompted losses [31]. Other approaches have included using autoencoders trained on model zoos to learn hyper-representations for parameter generation [36, 40], GNN-based parameter sampling [15], and direct MLP weight generation for neural implicit fields [8].

Most recently, diffusion models have been successfully applied to generate high-performing neural network parameters across various architectures and datasets, demonstrating their potential for weight generation tasks.

3. INR-based Data Augmentation

For the INR-based data augmentation part. In our framework, we incorporate several established augmentation methods for INRs. Consider an INR that maps an image, defined as $f : \mathbb{R}^2 \to \mathbb{R}^3$, where x represents a normalized coordinate grid in $[0, 1]^2$ serving as the input to the INR. Through manipulating the weight vector, we can simulate various image augmentations on the INR's representation. For instance, given a rotation matrix $R \in \mathbb{R}^{2\times2}$, multiplication of the first layer weights W_1 with R effectively rotates the image represented by the INR. Following similar principles, we can implement both translation and scaling operations through appropriate weight transformations. For INRs, we apply rotation, translation, and scaling augmentations. Besides all these, we have proposed a more fine-grained augmentation named bias perturbation. Different to directly adding Gaussian noise to both weights and biases, we simple implement the disturbance to the biases. In this way, a more fine-grained augmentation is applied due to different activation value. In addition, for RGB images, we have proposed a series of color jitter augmentations including brightness adjustment, contrast adjustment and saturation adjustment. See Fig. 1 for the example of our proposed data augmentations.

4. Smooth Augmentation

This part mainlu derived from our previous work [45], here we offer a condensed version of it. Neural networks require smooth signals for effective convergence, yet weight matrices typically exhibit non-smooth characteristics. To enable our Implicit Neural Representation (INR) to accurately reconstruct these weights, we implement a smoothness-preserving strategy through weight permutation.

Traditional approaches to weight permutation often treat it as a Traveling Salesman Problem (TSP), which oversimplifies the complex interconnections within neural networks. We identify three key issues with this approach:

- Weight matrix smoothness is better modeled as a Shortest Hamiltonian Path (SHP) problem rather than a TSP, as returning to the starting point is unnecessary.
- Permutations in one layer require corresponding adjustments in connected layers to maintain functional equivalence. For activation function $\sigma(\cdot)$ and permutation pair P and P^{-1}:

$$W_i P \sigma(P^{-1} W_{i-1} X) = W_i \sigma(W_{i-1} X) \tag{5}$$

- Improving smoothness along one dimension may degrade it in another dimension.

We leverage a key insight: permutation in one dimension does not affect total variation in the orthogonal dimension. For a weight matrix W and permutation P:

$$TV(WP) = TV_{in}(WP) + TV_{out}(W) \tag{6}$$

$$TV(PW) = TV_{in}(W) + TV_{out}(PW) \tag{7}$$

We formulate the neural network as a dependency graph $G = (V, E)$ where operations with weights form nodes and their connections form edges. We optimize permutations within each fully-connected clique $C = (V_C, E_C) \subset G$ through a multi-objective Shortest Hamiltonian Path (mSHP) problem:

$$\arg\min_{P} \sum_{e_{ij} \in E_C} \left(TV_{out}(PW_i) + TV_{in}(W_j P^{-1}) \right) \tag{8}$$

By converting this to a TSP with a dummy node and solving via 2.5-opt local search, we efficiently determine optimal permutations that preserve network functionality while enhancing weight smoothness across the entire model. With approximately 20 cliques per network, the total computation time remains under 4 seconds.

5. Equivariant Architecture

To construct our equivariant encoder, we follow the framework established in [27]. The weight space V is decomposed into $\mathcal{V} = \mathcal{W} \oplus \mathcal{B}$, where $\mathcal{W} := \bigoplus_{m=1}^{M} \mathcal{W}_m$ and $\mathcal{B} := \bigoplus_{m=1}^{M} \mathcal{B}_m$ represent the weight and bias spaces, respectively. The equivariant layer L is partitioned into four mappings: $L_{\text{ww}} : \mathcal{W} \to \mathcal{W}$, $L_{\text{wb}} : \mathcal{W} \to \mathcal{B}$, $L_{\text{bw}} : \mathcal{B} \to \mathcal{W}$, and $L_{\text{bb}} : \mathcal{B} \to \mathcal{B}$. These mappings are implemented using standard equivariant operations (pooling, broadcast, and linear layers) as described in [13, 46]. Our implementation follows the block matrix structure from [27], with modifications to the layer configuration and an additional projection layer.

6. Implementation Details

Our equivariant architecture is implemented with four hidden equivariant layers followed by a final invariant layer, with the output equivariant feature dimension set to 128. The training process consists of three distinct stages: First, during the equivariant encoder pre-training stage, we employ the AdamW [19] optimizer with a weight decay of 5e-4 and a learning rate of 5e-3. The encoder is trained for 500 epochs with a batch size of 512. Second, in the equivariance-guided diffusion stage, we utilize a squared cosine beta scheduler [29] to modulate noise injection across 1000 timesteps. For MLP sampling from the diffusion process, we implement the Denoising Diffusion Implicit Models (DDIM) [39]. For the transformer architecture, we follow the modified version of minGPT [31] which has 12 layers, 16 self-attention heads and 2880 hidden size. The input MLP is flattened into a 1D vector and projects to 6 tokens (weight and bias from each layer form 2 distinct tokens, our MLP has 3 layers, so that's 6 in total). We also incorporate an Exponential Moving Average (EMA) strategy with $\beta = 0.99$ to enhance convergence stability. This stage comprises 5000 epochs of training using the AdamW optimizer with a batch size of 32 and an initial learning rate of 2e-4, employing a decay schedule that reduces the learning rate by 10% every 250 epochs. The equivariance loss proportion parameter λ is set to 0.1 unless otherwise specified. The process of fine-tuning λ is demonstrated in subsequent sections Finally, during the few-shot fine-tuning stage, we maintain identical configurations to the previous stage while training for 250 epochs. At last, the subspace disturbance parameter γ in the generation process is set to 0.3 by default. All experiments are conducted on the Linux server with two L20 GPUs.

7. Additional Experimental Results

Impact of few-shot sample numbers. As demonstrated in Tables 1 and 2, our experimental results validate the consistency of both quality and diversity metrics across varying quantities of few-shot support samples. The inverse relationship between sample size and learning complexity manifests in the evaluation metrics: specifically, the FID for 2D scenarios and MMD for 3D scenarios exhibit elevated values, indicating degraded generation quality with reduced sample sizes. Similarly, for generative diversity metrics, LPIPS for 2D and COV for 3D scenarios demonstrate performance deterioration proportional to the reduction in support samples.

Table 1. Quantitative evaluation of different shot generation on both MNIST-INRs, CIFAR-10-INRs.

Shots	MNIST-INRs		CIFAR10-INRs	
	FID↓	LPIPS↑	FID↓	LPIPS↑
1-Shot	185.80	0.2169	202.70	0.2561
3-Shot	143.55	0.2907	180.72	0.3319
5-Shot	<u>127.39</u>	<u>0.3355</u>	<u>165.43</u>	<u>0.3702</u>
10-Shot	**121.24**	**0.4133**	**164.14**	**0.4926**

Table 2. Quantitative evaluation of different shot generation on ShapeNet-INRs.

Category	Shots	MMD↓	COV(%)↑	1-NNA(%)↓
Airplane	1-shot	4.8	20	85.2
	3-shot	4.1	26	79.5
	5-shot	<u>3.7</u>	<u>31</u>	<u>75.8</u>
	10-shot	**3.4**	**35**	**73.0**
Car	1-shot	5.0	18	88.3
	3-shot	4.2	23	82.4
	5-shot	<u>3.8</u>	<u>27</u>	<u>79.1</u>
	10-shot	**3.5**	**31**	**76.5**
Chair	1-shot	5.8	25	79.5
	3-shot	4.9	32	73.2
	5-shot	<u>4.5</u>	<u>37</u>	<u>69.8</u>
	10-shot	**4.2**	**41**	**67.1**

Additional ablation study. Similar results to 3D scenario in the main paper are obtained. As illustrated in Tab. 3, we conducted a detailed ablation analysis to evaluate the distinct effects of weight space smooth augmentation and equivariant subspace disturbance in 2D scenarios. The experimental results demonstrate that smooth augmentation independently improves both generation quality and diversity metrics. This finding corroborates that initialized from an optimized state, the equivariant encoder achieves enhanced representational capacity through contrastive learning. In contrast, the application of subspace disturbance alone yields higher diversity scores but exhibits a slight degradation in generation quality when implemented without smooth augmentation. This performance trade-off can

Table 3. Ablation study of weight space smooth augmentation and equivariant space disturbance on MNIST-INRs and CIFAR-10-INRs. Combined usage achieves optimal performance, while individual modules demonstrate distinct contributions to quality and diversity.

Dataset	Smooth	Disturbance	FID↓	LPIPS↑
MNIST-INRs	✗	✗	137.54	0.3357
	✓	✗	**120.15**	0.3466
	✗	✓	139.65	0.3988
	✓	✓	121.24	**0.4133**
CIFAR10-INRs	✗	✗	187.09	0.3974
	✓	✗	**157.83**	0.4175
	✗	✓	192.57	**0.4841**
	✓	✓	164.14	0.4822

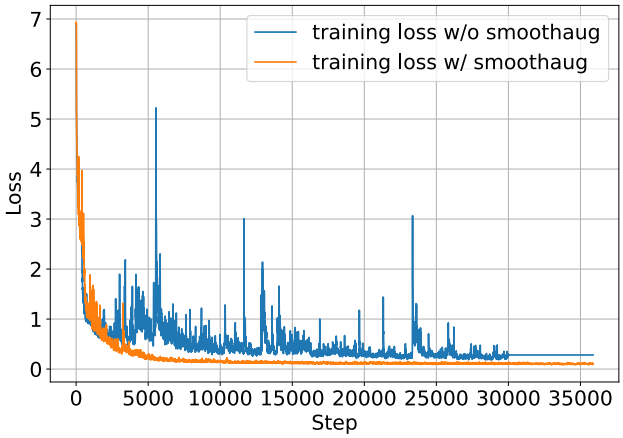


Figure 2. The training loss of the equivariant encoder pre-training. With and without smooth augmentation show notable differences in terms of the converging speed and the final converged loss.

be attributed to the divergent categorical clustering of equivariant features under identical disturbance conditions, contingent upon the presence or absence of smooth augmentation. Notably, the concurrent implementation of both components—smooth augmentation and subspace disturbance—yields optimal performance across all metrics.

Additional exploration of equivariant subspace. As a complementary to the 2D t-SNE visualization of the equivariant space, we offer the training loss both with and without the smooth augmentation on the MNIST-INRs in Fig Fig. 2. A comparatively faster converging speed and an obvious better final loss is obtained if the weights have been smooth augmented.

Impact of subspace disturbance. Similar to 3D scenario in the main paper, through quantitative assessment of equivariant subspace disturbance magnitude and its resultant effects on generative performance, as depicted in Fig. 6, we identify a significant inverse relationship between sample diversity and functional fidelity. Our analysis reveals that

Figure 3. Qualitative comparison of 3-shot generation results on the plane category of ShapeNet-INRs dataset. Notable geometric variations are exhibited across structural components, particularly in the configuration of wings, engine placements, and nose cone morphologies. Left: Support samples. Right: Generated samples.

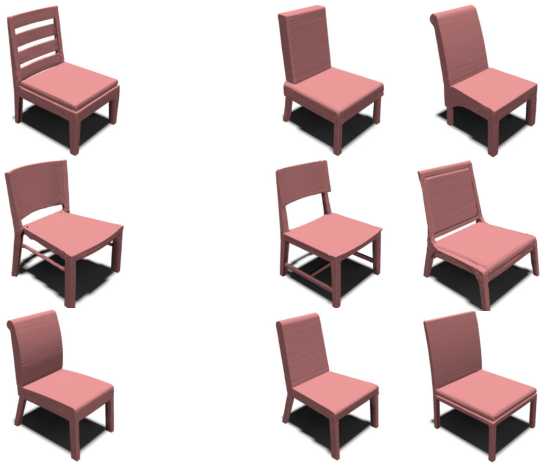


Figure 4. Qualitative comparison of 3-shot generation results on the chair category of ShapeNet-INRs dataset. Notable geometric variations are exhibited across structural components, particularly in the morphology of backrests, leg configurations, and cushion designs. Left: Support samples. Right: Generated samples.

the magnitude of subspace disturbance demonstrates a positive correlation with generation variance. Empirical evidence from Fig. 6(a) indicates that elevated disturbance magnitudes correspond to increased LPIPS scores, signifying enhanced inter-sample diversity. However, as evidenced in Fig. 6(b), this diversification manifests a concomitant degradation in generation quality, quantified by elevated FID metrics.

Qualitative results. To comprehensively evaluate the efficacy of our EQUIGEN, we present additional examples from the ShapeNet-INRs dataset, specifically demonstrating 3-

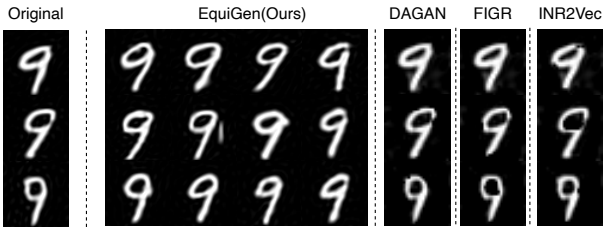
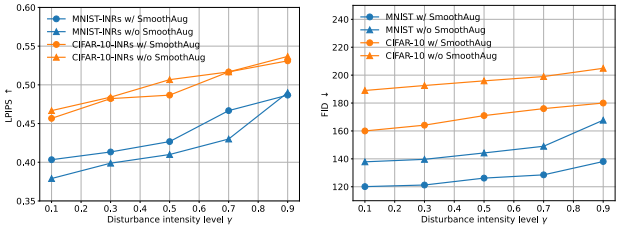


Figure 5. Qualitative results of 3-shot generation on MNIST-INRs. Compared to DAGAN, FIGR and INR2Vec, our method could provide obvious more diverse results.



(a) COV with different γ (b) MMD with different γ

Figure 6. The evaluation on MNIST-INRs and CIFAR-10-INRs with respect to different subspace disturbance intensity. (a) Higher disturbance leads to increased LPIPS, indicating greater sample diversity. (b) However, larger disturbances result in higher FID, reflecting decreased generation quality.

shot generation capabilities across both chair and plane categories. In the context of 3D shape, geometric diversity serves as a critical evaluation criterion. Our analysis reveals obvious structural variations across key components: in the plane category, we observe distinct modifications in wing geometry, engine placement configurations, and nose cone morphologies. Similarly, the chair category exhibits diverse architectural features, manifested through variations in backrest design, leg structure configurations, and cushion geometries.

To intuitively compare the proposed EQUIGEN with existing methods, we visualize the 3-shot generation on MNIST-INRs. As demonstrated in Fig. 5, our approach generates substantially more diverse outputs compared to baseline methods, which suffer from mode collapse, exhibiting minimal inter-sample variation in their generated results. To facilitate detailed analysis, we rendered our INR-based results at 56×56 resolution, leveraging INR's capacity for infinite-resolution rendering. Other baseline methods are rendered at the standard 28×28 resolution.

Evaluation of equivariant subspace. To further assess the success of mapping original weights into the equivariant subspace, we conducted additional evaluations beyond the qualitative visualizations in **??**. Specifically, we computed a Silhouette Score of 0.85 on CIFAR-10-INRs, indicating strong clustering behavior in the projected subspace. Furthermore, we trained a simple 1-layer MLP classifier on the equivariant subspace representation, achieving a classification accuracy of 92.01%, demonstrating that the learned subspace effectively preserves task-relevant information. These quantitative results confirm that the transfer to the equivariant space is well-structured and meaningful.

Impact of INR-based data augmentation on contrastive learning. Given the potential sensitivity of SimCLR-based contrastive learning to data augmentation, we conducted an ablation study on the influence of color jitter and bias disturbance on model performance. By evaluating classification accuracy, we observed that applying color jitter augmentation led to a 7.08% increase, while bias disturbance augmentation improved accuracy by 2.44%. However, our initial findings suggest that excessive augmentation—beyond 15% color jitter and 20% bias disturbance—induces mode collapse, leading to degraded feature separability. These results highlight the importance of carefully tuning augmentation parameters to balance diversity and stability in contrastive training.

References

[1] Andreas Blattmann, Robin Rombach, Kaan Oktay, Jonas Müller, and Björn Ommer. Retrieval-augmented diffusion models. *Advances in Neural Information Processing Systems*, 35:15309–15324, 2022. 2

[2] Rohan Chabra, Jan E Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard Newcombe. Deep local shapes: Learning local sdf priors for detailed 3d reconstruction. In *ECCV*, pages 608–625. Springer, 2020. 2

[3] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *CVPR*, pages 8628–8638, 2021. 2

[4] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *CVPR*, pages 5939–5948, 2019. 2

[5] Zhang Chen, Zhong Li, Liangchen Song, Lele Chen, Jingyi Yu, Junsong Yuan, and Yi Xu. Neurbf: A neural fields representation with adaptive radial basis functions. In *CVPR*, pages 4182–4194, 2023. 2

[6] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 2

[7] Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[8] Ziya Erkoç, Fangchang Ma, Qi Shan, Matthias Nießner, and Angela Dai. Hyperdiffusion: Generating implicit neural fields with weight-space diffusion. In *CVPR*, pages 14300–14310, 2023. 3

[9] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, pages 1126–1135. PMLR, 2017. 3

[10] Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T Freeman, and Thomas Funkhouser. Learning shape templates with structured implicit functions. In *CVPR*, pages 7154–7164, 2019. 2

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 2

[12] David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016. 3

[13] Jason Hartford, Devon Graham, Kevin Leyton-Brown, and Siamak Ravanbakhsh. Deep models of interactions across sets. In *ICLR*, pages 1909–1918. PMLR, 2018. 4

[14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2

[15] Boris Knyazev, Michal Drozdzal, Graham W Taylor, and Adriana Romero Soriano. Parameter prediction for unseen deep architectures. In *NeurIPS*, pages 29433–29448, 2021. 3

[16] Zhengqi Li, Qianqian Wang, Forrester Cole, Richard Tucker, and Noah Snavely. Dynibar: Neural dynamic image-based rendering. In *CVPR*, pages 4273–4284, 2023. 2

[17] Shaohui Liu, Yinda Zhang, Songyou Peng, Boxin Shi, Marc Pollefeys, and Zhaopeng Cui. Dist: Rendering deep implicit signed distance function with differentiable sphere tracing. In *CVPR*, pages 2019–2028, 2020. 2

[18] Zhen Liu, Yao Feng, Michael J Black, Derek Nowrouzezahrai, Liam Paull, and Weiyang Liu. Meshdiffusion: Score-based generative 3d mesh modeling. *arXiv preprint arXiv:2303.08133*, 2023. 2

[19] I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 4

[20] Sebastian Lunz, Yingzhen Li, Andrew Fitzgibbon, and Nate Kushman. Inverse graphics gan: Learning to generate 3d shapes from unstructured 2d data. *arXiv preprint arXiv:2002.12674*, 2020. 2

[21] Qi Ma, Danda Pani Paudel, Ajad Chhatkuli, and Luc Van Gool. Deformable neural radiance fields using rgb and event cameras. In *CVPR*, pages 3590–3600, 2023. 2

[22] Qi Ma, Danda Pani Paudel, Ajad Chhatkuli, and Luc Van Gool. Continuous pose for monocular cameras in neural implicit representation. In *CVPR*, pages 5291–5301, 2024. 2

[23] Julian McGinnis, Suprosanna Shit, Hongwei Bran Li, Vasiliki Sideri-Lampretsa, Robert Graf, Maik Dannecker, Jiazhen Pan, Nil Stolt-Ansó, Mark Mühlau, Jan S Kirschke, et al. Single-subject multi-contrast mri super-resolution via implicit neural representations. In *MICCAI*, pages 173–183. Springer, 2023. 2

[24] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4460–4470, 2019. 2

[25] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *CVPR*, pages 4460–4470, 2019. 2

[26] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022. 2

[27] Aviv Navon, Aviv Shamsian, Idan Achituve, Ethan Fetaya, Gal Chechik, and Haggai Maron. Equivariant architectures for learning in deep weight spaces. In *ICML*, pages 25790–25816. PMLR, 2023. 4

[28] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 2

[29] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *ICLR*, pages 8162–8171. PMLR, 2021. 4

[30] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *CVPR*, pages 165–174, 2019. 2

[31] William Peebles, Ilija Radosavovic, Tim Brooks, Alexei Efros, and Jitendra Malik. Learning to learn with generative models of neural network checkpoints. *arXiv preprint arXiv:2209.12892*, 2022. 3, 4

[32] Sameera Ramasinghe and Simon Lucey. Beyond periodicity: Towards a unifying framework for activations in coordinate-mlps. In *ECCV*, pages 142–158. Springer, 2022. 2

[33] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 2

[34] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022. 2

[35] Vishwanath Saragadam, Daniel LeJeune, Jasper Tan, Guha Balakrishnan, Ashok Veeraraghavan, and Richard G Baraniuk. Wire: Wavelet implicit neural representations. In *CVPR*, pages 18507–18516, 2023. 2

[36] Konstantin Schürholt, Boris Knyazev, Xavier Giró-i Nieto, and Damian Borth. Hyper-representations as generative models: Sampling unseen neural network weights. In *NeurIPS*, pages 27906–27920, 2022. 3

[37] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *NeurIPS*, pages 7462–7473, 2020. 2

[38] Edward J Smith and David Meger. Improved adversarial systems for 3d object generation and reconstruction. In *Conference on Robot Learning*, pages 87–96. PMLR, 2017. 2

[39] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. 2, 4

[40] Kai Wang, Zhaopan Xu, Yukun Zhou, Zelin Zang, Trevor Darrell, Zhuang Liu, and Yang You. Neural network diffusion. *arXiv preprint arXiv:2402.13144*, 2024. 3

[41] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. *Advances in neural information processing systems*, 29, 2016. 2

[42] 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*, pages 641–676. Wiley Online Library, 2022. 2

[43] Junshen Xu, Daniel Moyer, Borjan Gagoski, Juan Eugenio Iglesias, P Ellen Grant, Polina Golland, and Elfar Adalsteinsson. Nesvor: implicit neural representation for slice-to-volume reconstruction in mri. *IEEE TMI*, 42(6):1707–1719, 2023. 2

[44] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4541–4550, 2019. 2

[45] Xingyi Yang and Xinchao Wang. Neural metamorphosis. In *ECCV*, pages 1–19. Springer, 2025. 3

[46] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. In *NeurIPS*, 2017. 4

[47] Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. In *CVPR*, pages 5826–5835, 2021. 2