FewShotNeRF: Meta-Learning-based Novel view Synthesis for Rapid Scene-Specific Adaptation

Paul Janson
University of Moratuwa
paul.18@cse.mrt.ac.lk

Piraveen Sivakumar
University of Moratuwa
piraveen.18@cse.mrt.ac.lk

Jathushan Rajasegaran
UC Berkeley
jathushan@berkeley.edu

Thanuja Ambegoda
University of Moratuwa
thanuja@cse.mrt.ac.lk

Abstract

In this paper, we address the challenge of generating novel views of real-world objects with limited multi-view images through our proposed approach, ***FewShotNeRF****. Our method utilizes meta-learning to acquire an optimal initialization, facilitating rapid adaptation of a Neural Radiance Field (NeRF) to specific scenes. The focus of our meta-learning process is on capturing shared geometry and textures within a category, embedded in the weight initialization. This approach expedites the learning process of NeRFs and leverages recent advancements in positional encodings to reduce the time required for fitting a NeRF to a scene, thereby accelerating the inner loop optimization of meta-learning. Notably, our method enables meta-learning on a large number of 3D scenes to establish a robust 3D prior for various categories. Through extensive evaluations on the Common Objects in 3D open source dataset[27], we empirically demonstrate the efficacy and potential of meta-learning in generating high-quality novel views of objects.*

1. Introduction

Neural radiance fields (NeRF) [19] have emerged as a transformative technology in the realm of novel view synthesis [15, 29, 42], particularly in the context of posed multiview images. This advancement is attributed to the utilization of a coordinate-based representation [18, 22, 30], wherein a three-dimensional coordinate system is efficiently mapped to its corresponding color and density [19]. By adopting this approach, the representation of a three-dimensional scene becomes more compact and memory-efficient [7, 8, 32]. However, it is important to acknowledge that this enhancement comes at the expense of increased computational costs [19, 20].

To construct a NeRF capable of generating novel views, each scene necessitates the initialization of a new model from scratch, followed by training via volume rendering [13] using 2D supervision provided in the form of multi-view images. Nonetheless, the efficacy of this approach is subject to certain limitations [38]. Firstly, it relies on large datasets containing hundreds of images capturing a single scene, which may pose challenges in terms of data acquisition and storage [42]. Additionally, the computational demands associated with this methodology may be restrictive and require substantial computing resources [20]. Thus, while neural radiance fields offer considerable benefits in terms of compact representation and memory efficiency, their practical implementation is hindered by the reliance on extensive datasets and the computational burdens they entail.

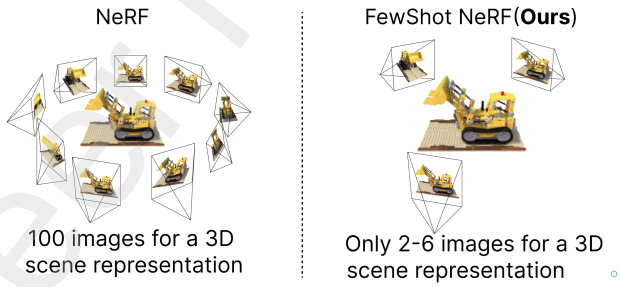

Figure 1. FewShot-NeRF: Learning Rich 3D Scenes from Minimal Camera Poses. Conventional NeRF training demands nearly 100 camera poses per scene. Our approach reduces this requirement by harnessing meta-learning to acquire an optimized initialization for NeRF. By incorporating a 3D prior into the parameter initialization, FewShot-NeRF learns a 3D scene with a minimal set of camera poses, effectively reducing frame requirements

As researchers delved into the realm of Neural Radiance Fields (NeRFs), they recognized the importance of addressing the challenge of generalization [31, 42, 44]. Several studies have emerged, each attempting to tackle this issue from various angles. The key idea behind these attempts

is to incorporate prior knowledge about the world into the initial NeRF models, enabling the learning of scene representations from a few views.

To inject these models with this prior knowledge, researchers have explored three main approaches [42]. The first involves conditioning the NeRF model on a latent code [4, 44], while the second entails learning a prior initialization [31, 38] that facilitates rapid convergence to a scene with limited views. The third way is to use diffusion generative models to generate views and use them to train a NeRF[45]. However, it is worth noting that models conditioned on a latent code may suffer from limited expressivity due to the constraints imposed by the code's size [42]. That means, once developed, the restriction caused by the latent code will remain as a constraint. Diffusion-based methods rely on a 2D prior. On the other hand, gradient-based meta-learning approaches inherently maintain the full expressivity of NeRF models, thereby enabling the representation of any scene that can be captured by per-scene optimized NeRF models. The prior knowledge learned is inherently 3D.

The first work investigated the usage of gradient-based meta-learning is [38], specifically using Model-Agnostic Meta-Learning(MAML) [9] framework, to learn improved initializations for NeRFs. However, their study focused on a simplified version of NeRF that lacks view dependence, and the generalization achieved by their method was limited to three specific categories on a synthetic dataset [3]. This limitation stems from the inefficient training of the vanilla NeRF and is further exacerbated by the memory requirements of the meta-learning process.

In contrast, our work aims to apply gradient-based meta-learning to NeRFs that incorporate view-dependent color output, with the objective of achieving generalization in many categories of commonly used objects. Moreover, our goal extends beyond synthetic datasets and encompasses the real-world objects shot on mobile phones [27]. By exploring these avenues, we seek to enhance the flexibility and adaptability of NeRF models for a wider range of scenes and categories.

Significant modifications have been made to the architecture of NeRFs to enhance their efficiency [17, 20, 28]. One crucial aspect of NeRFs is the need for an encoding function [19, 37] that maps the three-dimensional coordinate vectors to a higher-dimensional space to mitigate spectral bias[23].

The original NeRF [19] architecture employed positional encodings inspired by Transformer [40] models to fulfill this requirement. However, recent studies have demonstrated that replacing these encoding functions with task-specific, learnable data structures can improve training efficiency and facilitate faster convergence. For example, [16] and [36] have presented approaches that utilize modified encoding functions, resulting in accelerated training and convergence. These modifications have been shown to be beneficial in terms of computational efficiency.

Additionally, [20] proposed a multi-resolution hash encoding function that drastically reduced the training time required for NeRF convergence by several orders of magnitude. This advancement, coupled with improvements in ray tracing algorithms and efficient implementation techniques, contributed to overall efficiency enhancements in NeRF models. Notably, the introduction of the hash encoding function ensured convergence with a significantly lower number of iterations, further optimizing the training process.

Motivated by the aforementioned findings in the existing literature, our paper aims to make three significant contributions:

1. First, we propose the utilization of hash encoding as a way to accelerate the meta-learning process. This increases the feasibility of meta-learning on a large number of scenes.
2. To evaluate the effectiveness of our proposed method, we conduct extensive experiments on categories of real-world objects. By employing a diverse set of object categories, we can assess the performance and generalization capabilities of our approach in a realistic and practical context.
3. We investigate the efficacy of meta-learning in acquiring a 3D prior and explore its potential for generating novel views independently, without reliance on external 2D priors.

2. Related Work

Neural Fields/Implicit Neural Representation

Implicit Neural Representations are computational models that establish a mapping between input coordinates and signal values, enabling the encoding of 2D or 3D scenes within coordinate networks. These networks have found extensive applications in various visual learning tasks, including image representation [5, 34], 3D scene reconstruction from 2D images [18, 22], imaging inverse problems [35], and multi-view synthesis [19]. These neural networks exhibit a bias towards low spatial frequency functions. In order to address this spectral bias inherent in neural networks, [23] proposes a solution that leverages Fourier analysis to capture higher frequency functions. Another technique called positional embedding, initially employed in Natural Language Processing[6], has been adopted to map input coordinate vectors into embedded coordinated vectors. Sinusoidal embedding [40] and Fourier features, in conjunction with positional embedding, have been widely utilized in neural fields to capture higher frequency signals [37]. [32] introduces a method that replaces monotonic non-linearities with

periodic nonlinearities to achieve this objective.

Novel View Synthesis

Novel View Synthesis(NVS) pertains to the generation of a new viewpoint of a scene based on a given set of input camera images captured from various poses. Earlier approaches in the field, as discussed in [1], were capable of producing photorealistic views; however, they heavily relied on densely captured images. Recent advancements, as highlighted in [19] and [16], have made significant progress in novel view synthesis by utilizing 3D representations within neural networks, requiring fewer input images. Nevertheless, these methods necessitate multiple camera views for a single scene to fit a particular model, resulting in lengthy training times. Furthermore, a distinct model optimization process is required for each scene [12, 44].

To address the computational cost, a recent work by Muller *et al*. [20] introduced an innovative approach capable of training a model within a few minutes. Additionally, concurrent research efforts [43] have also aimed to enhance both training time and accuracy. In this study, we investigate novel view synthesis using a limited number of training samples, utilizing the approach presented in [20] as our base model.

Meta Learning

Meta-learning [41] is a machine-learning paradigm that involves pre-training a model to acquire the ability to learn efficiently. Notably, Model Agnostic Meta-Learning (MAML) [9] and Reptile [21] are optimization-based algorithms commonly used in meta-learning. In addition to these, there exist other variants of meta-learning algorithms, such as those described in [2, 24–26]. Gradient-based meta-learning employs outer loops of Stochastic Gradient Descent (SGD) to learn an improved initialization, enabling fast convergence when faced with new instances of the same task during testing [9]. Specifically, this approach has been applied to tasks related to neural representation, such as effectively fitting tasks to represent signed distance fields [22], with [39] introducing the concept of learned initialization as the first work to address gradient-based meta-learning for Neural Radiance Fields (NeRFs). However, their experiments were constrained to simplified NeRF architectures and evaluation settings.

Another approach within the realm of meta-learning involves learning a hyper network as a prior for model initialization. Hypernetworks [10] refer to neural networks that produce weights for another neural network. Several studies have utilized hyper networks to estimate weights for implicit neural networks [5, 33]. However, these early works focused solely on developing models with 2-dimensional output or models with 3D supervision.

A recent proposal [4] suggests employing a transformer as a hyper network, drawing inspiration from the similarities between gradient-based meta-learning and the residual connections found in transformers. In our research, our objective is to apply meta-learning to learn the initialization of view-dependent NeRFs, and subsequently evaluate its performance in a challenging setting that has not been extensively explored before.

3. Method

Reconstructing a scene in 3D faithfully requires lots of multi-view images. Given enough multi-view images NeRF [19] and other multi-view reconstruction methods can reconstruct the scene with reliable 3D shapes and texture. Essentially, more views add more constraints to the optimization problem, thus creating a faithful reconstruction of the real scene. However, if we have very few images of a scene, for example, if we have only one side view of a car, then we need to rely on some additional information such as cars are usually symmetric to get some estimates of the 3D shape. Therefore, the lesser the number of views we have, we need to rely on the additional priors about the world to solve this under-constrained problem.

In this work, we operate on a limited number of views (eg 2-6 views). This requires learning additional priors about the world, such as symmetries, smooth surfaces and even sometimes man-made priors like objects are usually rectangular, etc. For example, apples have a solid shape prior almost all of them are sphere-shaped, and plants for example share some priors on the texture, most of the leaves are usually green. This extra knowledge about the world can be enforced by allowing the model to only render a scene that lies on the manifold of real images. This explicitly adds more constraint to the texture of the underlying 3D shape thus, reducing the plausible reconstructions. The same can be applied for 3D shape priors, with points clouds. On the other hand, in this work, we propose **FewShotNeRF** to learn additional priors implicitly from the weights of the NeRF model. Our method heavily relies on the understanding of NeRF and Metalearning, and we briefly discuss these two ideas next.

Neural Radiance Fields (NeRF): NeRFs synthesis realistic 3D scenes by directly learning volumetric representations from 2D images. Unlike traditional methods such as point clouds or meshes, NeRF excels at capturing fine details and complex occlusions. NeRF employs a neural network, specifically a multi-layer perceptron (MLP), to approximate the volumetric scene representation. This network maps 3D coordinates to radiance values, yielding highly realistic scene synthesis. To render images from NeRF, it employs ray marching casting rays from a virtual camera and integrating radiance values along the ray's path. NeRF's training requires a dataset with 2D images and camera poses. Through differentiable rendering and gradient

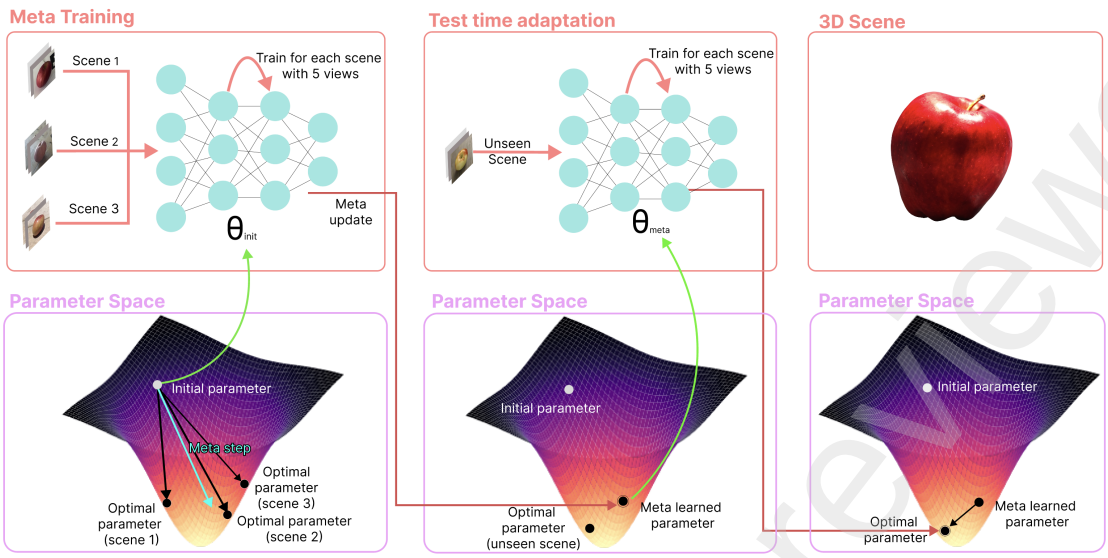

Figure 2. Method Overview: (Left) Our approach is rooted in the concept of meta-learning for initialization. We dynamically adjust the initialization by shifting it closer to the optimal parameters derived from NeRFs fitted to various scenes within the same category. This update leverages an extensive range of category-related scenes to imbue geometric resemblances into the initialization. (Center) During testing, we employ 2 to 6 images from distinct viewpoints, initiating NeRF fitting with the learned initialization. (Right) The resulting NeRF model facilitates the synthesis of novel views for the depicted scene.

descent, it optimizes MLP parameters by minimizing differences between rendered and ground truth images.

Meta Learning: Its objectives are effective adaptation and generalization. Here models are trained to adapt quickly to a task such that newer tasks can be solved with limited resources. Tasks within Meta Learning are typically structured within a task distribution or "meta-dataset." Individual tasks represent distinct learning problems, while the meta-dataset spans a range of tasks, facilitating the acquisition of transferable knowledge.

Meta-Learning Algorithms

In our study, we conducted a comprehensive evaluation of three prominent meta-learning algorithms, namely Reptile, First Order MAML, and Second Order MAML. Each algorithm was assessed based on multiple criteria, including computational efficiency, memory consumption, and meta-training adaptation steps. Through this systematic evaluation, we aimed to identify the most suitable algorithm for our task.

Reptile: Reptile emerged as a compelling choice due to its favorable computational cost and efficient training process. It demonstrated remarkable performance in scenarios with a large number of meta-training adaptation steps (200). Additionally, Reptile exhibited fast convergence and required relatively less memory consumption, making it well-suited for our experimental setup. However, it's important to note that Reptile's performance might plateau with a limited number of adaptation steps, leading to the need for a larger step count to achieve optimal results.

$$\theta' = \theta + \alpha(\theta'_{\text{meta}} - \theta) \tag{1}$$

Here, θ represents the initial model parameters, θ'_{meta} is the meta-learned model parameters, and α is the meta-learning rate.

First Order MAML: First Order MAML exhibited competitive performance, sharing similarities with Reptile in terms of the number of meta-training steps required for adaptation. However, one notable distinction was its higher memory consumption, which could impact its scalability for larger datasets or more complex tasks. While it offered comparable results, the increased memory requirements might limit its practical utility in certain scenarios.

$$\theta' = \theta - \alpha \nabla_\theta \mathcal{L}_{\text{meta-train}}(f_\theta) \tag{2}$$

Here, ∇_θ denotes the gradient with respect to the model parameters θ, and $\mathcal{L}_{\text{meta-train}}(f_\theta)$ represents the meta-training loss of the model f_θ.

Second Order MAML: Second Order MAML demonstrated a unique profile, showcasing the potential for achieving comparable outcomes with a relatively low number of meta-training adaptation steps (10). However, its utility was offset by very high memory consumption, which could hinder its applicability in resource-constrained environments. Despite its capacity to achieve competitive re-

sults, the trade-off between memory usage and performance improvement should be carefully considered.

$$\theta' = \theta - \alpha(\nabla_\theta \mathcal{L}_{\text{meta-train}}(f_\theta) + \beta \nabla_\theta^2 \mathcal{L}_{\text{meta-train}}(f_\theta)) \quad (3)$$

In this equation, β represents the second-order meta-learning rate, and ∇_θ^2 denotes the Hessian matrix (second-order gradient) with respect to the model parameters θ.

FewShotNeRF

We formally define our problem as follows: we consider a set of images denoted as $\mathbf{I} = \{I_1, I_2, ..., I_n\}$ along with their corresponding poses $\mathbf{P} = \{P_1, P_2, ..., P_n\}$. Additionally, we have a fixed budget of m optimization steps allocated for a specific scene $\mathcal{S}$. It is important to note that the size of $\mathbf{I}$ is limited to 2-6 images depending on the setup ($2 \leq n \leq 6$). We aim to learn a function f_θ that can be utilized to generate a new image set $\mathbf{I}'$ using their corresponding poses $\mathbf{P}'$. Importantly, there should be no overlap between the poses in $\mathbf{P}$ and $\mathbf{P}'$, denoted by $\mathbf{P} \cap \mathbf{P}' = \Phi$. The function f_θ follows the same architecture as described in NeRF [19], and the generated images are produced accordingly. The objective of training f_θ is to ensure that the generated image set $\mathbf{I}'$ matches the ground truth images $\mathbf{I}'_{gt}$ for the corresponding poses. Here, $\mathbf{I}'_{gt}$ represents the ground truth images for the given poses $\mathbf{P}'$. From the set of scenes, first, we sample a random k number of scenes $\{\mathcal{S}_1, \mathcal{S}_2, ...\mathcal{S}_k\}$. For each scene in the randomly sampled set, we learn a NeRF optimization in the inner loop.

$$\theta_j^i = NeRF(\mathcal{S}_j, \theta^i) \quad (4)$$

Here, θ_j^i is the optimized parameters of the MLP, for the scene $\mathcal{S}_j$, where the optimization is initialized with the parameters θ^i at ith meta-learning iterations. This process defines the inner loop optimization of the meta-learning algorithm, and here the inner loop is equivalent to the NeRF optimization. For the outer loop optimization, we use Reptile [21] as our main meta-learning algorithm, and we abilate performance of FewShotNeRF when optimizing with MAML [9] and FOMAML [9]. After optimizing the task-specific parameters in the inner loop, we update the meta-parameters in the out loop using a simple weighted sum of inner loop gradients as shown in Fig 2.

$$\theta^{i+1} = \theta^i - \alpha \sum_{j=1}^{k} (\theta_j^i - \theta^i) \quad (5)$$

Here,i is the number of meta-learning iterations. α is the learning rate of the meta-learning algorithm. We train the meta parameters for over a few hundred iterations and then at test time given a few views of a novel scene, we apply inner loop optimization (NeRF) to the meta initialized parameters θ and then render novel views using the inner-loop optimized parameters.

4. Experiments

We conducted a comprehensive evaluation of our approach using the publicly available CO3D dataset [27], which encompasses real-world multi-view objects. To assess the effectiveness of our method in addressing the challenge of generalizing Neural Radiance Fields (NeRFs), we performed a comparative analysis against existing techniques. Our evaluation encompassed diverse scenarios characterized by different quantities of input views employed for NeRF generation, specifically utilizing 2, 3, and 6 input frames in our experimental setups.

4.1. Setup

Dataset: We subjected our method to evaluation using the CO3D dataset [27]. It has different scenes belonging to 50 categories of commonly used objects. The frames are taken from mobile phone videos. We select only 10 core categories following [45] to perform our experiments. This dataset provides essential components, including relative camera poses for each frame and masks that delineate the object of interest from the background. Our selection of this dataset was motivated by the aim to investigate the efficacy of meta-learning in accelerating the learning process of Neural Radiance Fields (NeRF) within real-world scenes. This contrasts with the approach taken in [38], which concentrated on synthetic scenes characterized by a simplified NeRF architecture.

Baselines: We conducted thorough comparisons between our method and several existing approaches, all of which have been suitably adapted to accommodate the CO3D dataset as presented in [45].

Given the category-specific nature of our method, we conducted comparisons against a tailored category-specific version of Pixel-NerF [44]. This variant leverages pixel-wise image feature re-projection of CNN features to achieve its results. In addition, we evaluated our method against NerFormer [27] which is based on the feature-reprojection technique and ViewFormer [14] which is based on autoregressive generation.

Furthermore, we assessed the performance of our method against the state-of-the-art approach, Sparsefusion [45]. This method employs a diffusion-based prior to address data scarcity issues effectively. Through these comparisons, we demonstrated the distinct strengths and capabilities of our approach within the context of the CO3D dataset.

Implementation Details: We used a PyTorch implementation of Instant NGP [20] as our backbone model. We rendered the images at 128 x 128 following Zhou and Tulsiani [45] to ease the memory constraints. For meta-learning we adopted the Reptile [21] algorithm. Our method consisted of two main phases. The first phase is the Meta-learning phase and the second phase is the Test time adaptation

	Donut	Apple	Hydrant	Vase	Cake	Ball	Bench	Suitcase	Teddybear	Plant
	PSNR ↑	PSNR ↑	PSNR ↑	PSNR ↑	PSNR ↑	PSNR ↑	PSNR ↑	PSNR ↑	PSNR ↑	PSNR ↑
PixelNeRF[44]	20.9	20.0	19.0	21.3	18.3	18.5	17.7	21.7	18.5	19.3
NeRFormer[27]	20.3	19.5	18.2	17.7	16.9	16.8	15.9	20.0	15.8	17.8
ViewFormer[14]	19.3	20.1	17.5	20.4	17.3	18.3	16.4	21.0	15.5	17.8
EFT[45]	21.5	22.0	21.6	21.1	19.9	21.4	17.8	23.0	19.8	20.4
VLDM[45]	20.1	21.3	20.1	20.2	18.9	20.3	16.6	21.3	17.9	18.9
SparseFusion[45]	22.8	22.8	22.3	22.8	20.8	22.4	16.7	22.2	20.6	20.0
FewShotNeRF(25%)	**23.9**	**23.2**	**22.6**	**24.2**	**22.4**	**22.5**	**19.9**	**24.7**	**20.9**	**21.3**
FewShotNeRF(50%)	22.6	22.2	21.7	22.3	20.8	21.0	18.5	23.0	19.4	20.5
FewShotNeRF(75%)	21.7	21.3	21.0	20.8	19.6	19.9	17.4	21.7	18.3	19.3
FewShotNeRF(100%)	20.5	20.1	20.2	19.2	18.2	18.8	16.3	20.3	17.0	18.1

Table 1. Results on the CO3D dataset comparing our method with baselines on categories Donut, Apple, Hydrant, Vase, Cake, Ball, Bench, Suitcase, Teddybear, and Plant. Our method outperforms most of the methods and seems to show competitive performance to the SparseFusion, while not relying on any external models. To provide a nuanced understanding of our method's performance, we went beyond conventional averaging techniques. Unlike SparseFusion, which averages results from randomly selecting 10 scenes, we conducted a thorough evaluation using 150 scenes. We then calculated averages for each quartile, breaking down our method's performance at 25%, 50%, 75%, and 100% of the scenes. For the 25% quartile, we sorted the 150 scenes based on PSNR and selected the top 25%, demonstrating the robustness of our method even when considering only the scenes with the highest quality. Moving to the 50% quartile, we continued this process, ensuring a balanced representation of the dataset. At the 75% quartile, our evaluation included scenes that ranked within the top three-quarters based on PSNR, providing a broader perspective on our method's effectiveness. Finally, the 100% quartile encompassed the entire dataset, offering a comprehensive overview of our method's performance across the entirety of the tested scenes.

	2 Views		**3 Views**		**6 Views**	
	PSNR ↑	SSIM ↓	PSNR ↑	SSIM ↓	PSNR ↑	SSIM ↓
PixelNeRF[44]	19.52	0.667	20.67	0.712	22.47	0.776
NerFormer[27]	17.88	0.598	18.54	0.618	19.99	0.661
ViewFormer[14]	18.37	0.697	18.91	0.704	19.72	0.717
EFT[45]	20.85	0.680	22.71	0.747	24.57	0.804
VLDM [45]	19.55	0.711	20.85	0.737	22.35	0.768
SparseFusion[45]	21.34	0.752	20.85	0.766	23.74	0.791
FewShotNeRF(25%)	**22.50**	**0.781**	**23.01**	**0.781**	**25.76**	0.792

Table 2. Results on the CO3D dataset with 2,3,6 views on average across all of the selected 10 categories. We compare our method **FewShotNeRF** with PixelNeRF, NerFormer, SparseFusion, etc. Our experiments show that FewShotNeRF outperforms most of the comparisons and performs on par with SpareFusion. Note that the evaluation protocols are slightly different and our evaluation is more robust and stronger than a random sampling of 10 scenes.

phase. During the Meta-learning phase, We used a fixed budget of 200 inner optimization steps to adapt the model to a specific scene. We randomly sampled 5 scenes from the scenes selected for meta-learning after leaving 150 scenes for the testing phase and adapted them. The outer loop ran through 8 steps. During the Test time adaptation phase, we tested the method on the 150 scenes left out for testing, and using 2,3, or 6 frames, we adapted the model for 400 inner optimization steps. We evaluated the models using the PSNR and SSIM metrics on the remaining test frames.

Meta-Learning Algorithm: We opted for the Reptile meta-learning algorithm due to its suitability for our context. Learning a NeRF involves substantial computational demands, and incorporating memory-intensive algorithms like the one proposed in [9] would prove impractical, especially when dealing with a large dataset. To further understand that, We systematically evaluated three distinct meta-learning algorithms, ultimately selecting Reptile based on its favorable computational cost.

Meta Learning Algorithms	Mean (PSNR)	Standard Deviation	Variance	No of Meta Training iterations
Reptile [21]	22.04	2.76	7.62	200
MAML First Order[9]	18.42	3.23	10.46	200
MAML Second Order[9]	18.45	3.22	10.35	10

Table 3. This table compares the performance of three meta-learning algorithms in the Apple category. Reptile outperformed MAML First Order with the same meta-training iterations. However, MAML Second Order achieved competitive results with significantly fewer iterations, showcasing its efficiency. The findings highlight MAML Second Order's ability to achieve comparable performance with a reduced number of meta-training steps, challenging the conventional approaches of Reptile and MAML First Order.

Metrics: We conduct a comparative analysis between our method and related approaches, employing quantitative metrics including Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM). PSNR quantifies the absolute likeness between the reconstructed view and ground truth, while SSIM evaluates the structural similarity between these views.

4.2. Results on CO3D

This section showcases the quantitative outcomes of our approach under diverse input view scenarios. We meticulously choose these views to align evenly along the circular trajectory followed during object capture. This selection is crucial as our method relies only on the input view signal and the initialization prior, so selecting the input views evenly spaced is pivotal in generating a coherent and realistic object representation.

2-view Setup: Table 1 shows the comparison of our method with the baselines in the challenging scenario of just 2 input views. PSNR values on each of the 10 selected categories are shown. We report the values taken from [45]. Zhou and Tulsiani [45] selected only 10 scenes from the selected categories to test and report the values. Since the scene ids are not provided and it is not clear how these scenes are selected, we report the average PSNR value of all the scenes (150 scenes for each category in the test set) from the test split, for our method. Additionally, we also provide different quartile values of our method to emphasize the variation in the results. Our method performs better than NeRFormer[27], [44] on various catagories. We can see competitive results to [45] even without using an external prior such as a Diffusion model[11] and our numbers are computed over 150 scenes not random 10 scenes.

3-view Setup: Our 3-view setup is very similar to the 2-view setting, except the only change is the model sees 3 views during the training and adoption. Compared to the 2-view setting, the 3-view setting performs better. While this is an obvious observation in the NeRF land, this performance gain in the meta-learning setting suggests that the meta objective is able to capture strong inner-loop priors to the outer-loop. Detailed results, representing the average outcomes across the ten designated categories, are presented in 2.

6-view Setup: Extending our experimentation to a 6-view setup yields further improvements in the results. This progression is documented in Table 2.

4.3. Evolution of Image Quality via Meta-Training Iterations

In our experiment, we checked how image quality improves over several iterations of meta-training. We looked at the Peak Signal-to-Noise Ratio (PSNR) for 10 different scenes in each category during each round. The results showed a clear and consistent trend. Figure 3 displayed an increase in PSNR values with more meta-training iterations. This indicates a continual improvement in image quality. Our approach, which involved evaluating multiple scenes and averaging their PSNR values, highlights the reliability of our findings and the effectiveness of iterative meta-training in making images better.

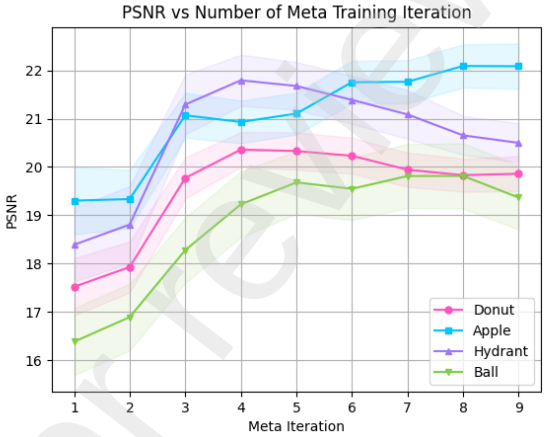

Figure 3. Evolution of PSNR Across Meta-Training Iterations. This graph illustrates the progressive increase in Peak Signal-to-Noise Ratio (PSNR) values with the number of meta-training iterations. The study includes an average of 10 scenes per category, highlighting the consistent improvement in image quality achieved through the iterative meta-training process.

5. Discussion

In this paper, we introduce **FewShotNeRF**, an approach to generalize Neural Radiance Fields (NeRFs) for view synthesis with few input views. We make three significant contributions through this work. First, we propose to utilize hash encoding to accelerate the training of NeRF models in the inner loop of meta-learning. Second, we conduct extensive experiments on real-world object categories to evaluate the effectiveness of this method and scale the meta-training to over 300 scenes to distill the 3D priors into a single model. These experiments provide valuable insights into the feasibility and potential benefits of using hash encoding for meta-learning NeRF models. Finally, our proposed method relies on learning 3D priors only from the meta objective without relying on external models.

In conclusion, the presented findings have the potential to shed light on the NeRF generalization. The utilization of hash encoding for meta-learning initialization, along with the extensive experimental evaluations, contributes to the advancement of generalizable NeRFs. Future work could further refine and extend the proposed methodology by exploring additional enhancements and could also focus on applying FewShotNeRF to more complex and challenging scenarios like dynamic scenes.

Figure 4. This sequence of images illustrates the qualitative progress achieved across four categories - hydrant, apple, ball, and donut. Beginning with the training input, followed by novel views generated using only 2, 3, and 6 training images, we witness the model's ability to enhance realism and accuracy in novel view generation.

References

[1] Light Field Rendering. https://graphics.stanford.edu/papers/light/. 3

[2] Antreas Antoniou, Harrison Edwards, and Amos Storkey. How to train your MAML. In *International Conference on Learning Representations*, 2022. 3

[3] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 2

[4] Yinbo Chen and Xiaolong Wang. Transformers as Meta-Learners for Implicit Neural Representations. In *2022 European Conference on Computer Vision*, Tel Aviv, Israel, 2022. arXiv. 2, 3

[5] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning Continuous Image Representation with Local Implicit Image Function. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8624–8634, Nashville, TN, USA, 2021. IEEE. 2, 3

[6] KR1442 Chowdhary and KR Chowdhary. Natural language processing. *Fundamentals of artificial intelligence*, pages 603–649, 2020. 2

[7] Emilien Dupont, Adam Goliński, Milad Alizadeh, Yee Whye Teh, and Arnaud Doucet. COIN: COmpression with Implicit Neural representations, 2021. 1

[8] Emilien Dupont, Hrushikesh Loya, Milad Alizadeh, Adam Goliński, Yee Whye Teh, and Arnaud Doucet. COIN++: Data Agnostic Neural Compression, 2022. 1

[9] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017. 2, 3, 5, 6

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

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 7

[12] Mohammad Mahdi Johari, Yann Lepoittevin, and François Fleuret. GeoNeRF: Generalizing NeRF with Geometry Priors. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orelands, 2022. IEEE. 3

[13] James Kajiya and Brian von herzen. Ray Tracing Volume Densities. *ACM SIGGRAPH Computer Graphics*, 18:165–174, 1984. 1

[14] Jonáš Kulhánek, Erik Derner, Torsten Sattler, and Robert Babuška. Viewformer: Nerf-free neural rendering from few images using transformers. In *European Conference on Computer Vision*, pages 198–216. Springer, 2022. 5, 6

[15] Hoang Le. Novel View Synthesis - A Neural Network Approach. Technical report, 2000. 1

[16] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural Sparse Voxel Fields. In *Advances in Neural Information Processing Systems*, pages 15651–15663. Curran Associates, Inc., 2020. 2, 3

[17] Julien N. P. Martel, David B. Lindell, Connor Z. Lin, Eric R. Chan, Marco Monteiro, and Gordon Wetzstein. ACORN: Adaptive Coordinate Networks for Neural Scene Representation. In *ACM Transactions on Graphics*. ACM, 2021. 2

[18] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy Networks: Learning 3D Reconstruction in Function Space. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4455–4465, Long Beach, CA, USA, 2019. IEEE. 1, 2

[19] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, 2020. 1, 2, 3, 5

[20] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics*, 41 (4):1–15, 2022. 1, 2, 3, 5

[21] Alex Nichol, Joshua Achiam, and John Schulman. On First-Order Meta-Learning Algorithms, 2018. 3, 5, 6

[22] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 165–174, Long Beach, CA, USA, 2019. IEEE. 1, 2, 3

[23] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the Spectral Bias of Neural Networks. In *Proceedings of the 36th International Conference on Machine Learning*, pages 5301–5310. PMLR, 2019. 2

[24] Jathushan Rajasegaran, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Mubarak Shah. Meta-learning the learning trends shared across tasks. *arXiv preprint arXiv:2010.09291*, 2020. 3

[25] Jathushan Rajasegaran, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Mubarak Shah. iTAML: An Incremental Task-Agnostic Meta-learning Approach. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13585–13594, Seattle, WA, USA, 2020. IEEE.

[26] Jathushan Rajasegaran, Chelsea Finn, and Sergey Levine. Fully online meta-learning without task boundaries. *arXiv preprint arXiv:2202.00263*, 2022. 3

[27] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common Objects in 3D: Large-Scale Learning and Evaluation of Real-life 3D Category Reconstruction. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10881–10891, Montreal, QC, Canada, 2021. IEEE. 1, 2, 5, 6, 7

[28] Vishwanath Saragadam, Jasper Tan, Guha Balakrishnan, Richard G. Baraniuk, and Ashok Veeraraghavan. MINER: Multiscale Implicit Neural Representations, 2022. 2

[29] Daniel Scharstein. *View synthesis using stereo vision*. Springer, 2003. 1

[30] Vincent Sitzmann, Michael Zollhoefer, and Gordon Wetzstein. Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019. 1

[31] Vincent Sitzmann, Eric Chan, Richard Tucker, Noah Snavely, and Gordon Wetzstein. MetaSDF: Meta-Learning Signed Distance Functions. In *Advances in Neural Information Processing Systems*, pages 10136–10147. Curran Associates, Inc., 2020. 1, 2

[32] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit Neural Representations with Periodic Activation Functions. In *Advances in Neural Information Processing Systems*, pages 7462–7473. Curran Associates, Inc., 2020. 1, 2

[33] Ivan Skorokhodov, Savva Ignatyev, and Mohamed Elhoseiny. Adversarial Generation of Continuous Images, 2021. 3

[34] Kenneth O. Stanley. Compositional pattern producing networks: A novel abstraction of development, 2007. 2

[35] Yu Sun, Jiaming Liu, Mingyang Xie, Brendt Wohlberg, and Ulugbek S. Kamilov. CoIL: Coordinate-based Internal Learning for Imaging Inverse Problems, 2021. 2

[36] Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Neural Geometric Level of Detail: Real-time Rendering with Implicit 3D Shapes. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11353–11362, Nashville, TN, USA, 2021. IEEE. 2

[37] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains, 2020. 2

[38] Matthew Tancik, Ben Mildenhall, Terrance Wang, Divi Schmidt, Pratul P. Srinivasan, Jonathan T. Barron, and Ren Ng. Learned Initializations for Optimizing Coordinate-Based Neural Representations. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2845–2854, Nashville, TN, USA, 2021. IEEE. 1, 2, 5

[39] Matthew Tancik, Ben Mildenhall, Terrance Wang, Divi Schmidt, Pratul P. Srinivasan, Jonathan T. Barron, and Ren Ng. Learned Initializations for Optimizing Coordinate-Based Neural Representations, 2021. 3

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. 2

[41] Ricardo Vilalta and Youssef Drissi. A perspective view and survey of meta-learning. *Artificial intelligence review*, 18: 77–95, 2002. 3

[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, 2022. 1, 2

[43] Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance Fields without Neural Networks, 2021. 3

[44] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural Radiance Fields from One or Few Images. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2021. 1, 2, 3, 5, 6, 7

[45] Zhizhuo Zhou and Shubham Tulsiani. Sparsefusion: Distilling view-conditioned diffusion for 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12588–12597, 2023. 2, 5, 6, 7

FewShotNeRF: Meta-Learning-based Novel view Synthesis for Rapid Scene-Specific Adaptation

Supplementary Material

5.1. Metrics

We compare our method with the baselines using the standard metrics used for Image-based comparisons. We selected PSNR as a pixel-wise comparison method and SSIM as a perceptual comparison metric. Those results are calculated as follows.

$$PSNR = -10 \log_{10}(MSE) \tag{6}$$

Where MSE is calculated as follows

$$MSE = \frac{1}{w \cdot h} \sum_{i=1}^{w} \sum_{j=1}^{h} (I_{\text{original}}(i,j) - I_{\text{reconstructed}}(i,j))^2 \tag{7}$$

Here w represents the width of the image and h represents the height of the image.

The equation we used to calculate the $SSIM$ is as follows.

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{x,y} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{8}$$

Let x and y denote the original and reconstructed images, respectively. Furthermore, let μ and σ represent the mean and standard deviation of pixel intensities in the images. The covariances of pixel intensities in the original and reconstructed images are denoted by $\sigma_{x,y}$. To prevent zero-division errors, constants c_1 and c_2 are introduced.

5.2. Impact of View Count on Image Enhancement Quality

This table 4 underscores the efficacy of meta-learning in the context of Neural Radiance Fields (NeRF), specifically in rapidly acquiring a scene's understanding with a few images while achieving high Peak Signal-to-Noise Ratio (PSNR). The comparison between meta-training with 3 views and 6 views reveals the model's capacity to excel in scene comprehension without reliance on external models. The reported PSNR values showcase the efficiency of meta-learning in enhancing image quality, particularly noteworthy for its ability to achieve substantial improvements even with a limited number of input images.

	3 Views								6 Views							
	Top 25%		Top 50%		Top 75%		Top 100%		Top 25%		Top 50%		Top 75%		Top 100%	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Donut	23.43	0.783	22.27	0.693	21.39	0.583	20.30	0.554	26.45	0.797	25.45	0.773	24.57	0.753	23.41	0.723
Apple	25.28	0.794	24.26	0.703	23.30	0.682	22.04	0.653	27.71	0.808	26.77	0.789	25.95	0.774	24.71	0.745
Hydrant	22.36	0.850	21.36	0.740	20.70	0.722	19.96	0.692	26.03	0.862	25.14	0.851	24.47	0.840	23.60	0.814
Vase	24.43	0.826	22.45	0.700	20.99	0.647	19.52	0.586	26.94	0.847	25.33	0.808	23.71	0.766	21.93	0.699
Cake	22.03	0.801	20.25	0.678	18.82	0.500	17.52	0.461	26.10	0.775	24.63	0.739	23.26	0.706	21.48	0.650
Ball	23.29	0.742	21.54	0.602	20.45	0.573	19.33	0.538	25.35	0.767	23.82	0.731	22.71	0.702	21.46	0.662
Bench	20.73	0.794	19.39	0.624	18.36	0.532	17.30	0.483	21.96	0.708	20.64	0.651	19.56	0.604	18.37	0.552
Suitcase	25.08	0.809	23.34	0.629	22.09	0.596	20.74	0.556	28.46	0.837	26.74	0.809	25.20	0.779	23.71	0.739
Teddybear	20.87	0.749	19.40	0.568	18.28	0.533	17.03	0.488	24.34	0.763	22.64	0.714	21.01	0.659	19.30	0.593
Plant	22.60	0.662	21.21	0.565	20.16	0.531	19.12	0.491	24.32	0.708	22.74	0.663	21.70	0.622	20.44	0.572

Table 4. Comparison of PSNR and SSIM Values for Meta-Training with 3 Views and 6 Views. The table presents the Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) values obtained through meta-training using 3 views and 6 views, without dependency on external models. Additionally, the table reports average PSNR and SSIM values by selecting the top quartiles from the evaluated scenes, demonstrating the impact of varying view counts on image quality enhancement.