Boost Your NeRF: A Model-Agnostic Mixture of Experts Framework for High Quality and Efficient Rendering

Francesco Di Sario[1][0009−0005−6969−1246],
Riccardo Renzulli[1][0000−0003−0532−5966],
Enzo Tartaglione[2][0000−0003−4274−8298], and
Marco Grangetto[1][0000−0002−2709−7864]

[1] University of Turin, Italy
[2] LTCI, Télécom Paris, Institut Polytechnique de Paris, France
francesco.disario@unito.it

Abstract. Since the introduction of NeRFs, considerable attention has been focused on improving their training and inference times, leading to the development of Fast-NeRFs models. Despite demonstrating impressive rendering speed and quality, the rapid convergence of such models poses challenges for further improving reconstruction quality. Common strategies to improve rendering quality involves augmenting model parameters or increasing the number of sampled points. However, these computationally intensive approaches encounter limitations in achieving significant quality enhancements. This study introduces a model-agnostic framework inspired by Sparsely-Gated Mixture of Experts to enhance rendering quality without escalating computational complexity. Our approach enables specialization in rendering different scene components by employing a mixture of experts with varying resolutions. We present a novel gate formulation designed to maximize expert capabilities and propose a resolution-based routing technique to effectively induce sparsity and decompose scenes. Our work significantly improves reconstruction quality while maintaining competitive performance.

1 Introduction

Neural Radiance Fields (NeRFs) [24] have recently shown impressive results in synthesizing photo-realistic 3D scenes from a set of 2D images. However, NeRFs suffer from limited scene diversity, long training time, and sensitivity to training data [21]. Since the introduction of NeRFs, significant attention has been directed toward improving their training and inference times, resulting in the development of Fast-NeRFs. By using auxiliary data structures such as voxel grids to store scene geometry and skip empty spaces, the training and inference process can be accelerated by several orders of magnitude. As a result, the neural component of these models, which is usually used to transform learned features into view-dependent color representations, becomes much smaller and can sometimes be replaced entirely by spherical harmonics [10]. Despite Fast

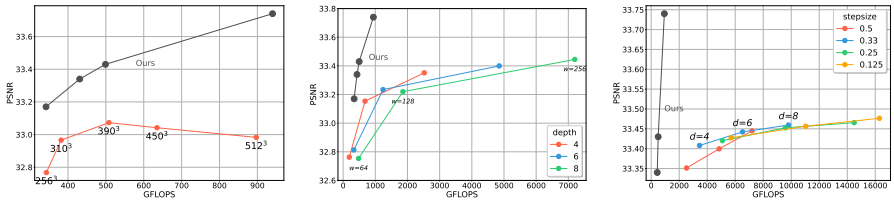


Fig. 1: Different strategies for improving reconstruction quality with Fast-NeRFs (DVGO [29]). Increasing the resolution of data structures like voxel grids can improve render quality, but only up to a certain point, after which quality declines (left). The MLP component's impact on rendering quality is analyzed by varying its depth and width while keeping other variables constant (center). The effect of the number of sampled points along each ray is examined by decreasing the step size (right). In gray, our method's performance, which significantly improves PSNR at a low computational cost.

NeRFs being an improvement from the original NeRF framework, achieving high-quality and efficient reconstructed geometry is still an open problem.

Typical "naive" approaches to enhance the reconstruction quality of these models include:

1. Increasing the parameters and resolution of the used data structures (e.g., voxel grid, hash grid, etc.).
2. Increasing the number of sampled points per ray.
3. Increasing the number of parameters in the neural network or the order of spherical harmonics.

However, as evident from Figure 1, the increase in reconstruction quality results in a significant increase in computational costs. In the first approach, increasing the resolution can lead to a significant improvement in the reconstruction quality, but a plateau is reached beyond which overfitting occurs, and reconstruction quality degrades. This comes at the expense of a considerable increase in spatial complexity (especially with dense voxel grids) and training times. Increasing the number of sampled points per ray can marginally improve reconstruction quality further but at a significant increase in computational complexity. In fact, the higher the number of sampled rays, the higher forward passes through a neural network are needed. Augmenting the number of parameters in the neural network is another solution. However, as the neural network grows in depth and width, it tends towards a fully implicit model, deviating from the principles of fast models (limiting the neural part as much as possible - or even removing it altogether).

We propose a technique capable of significantly enhancing the reconstruction quality of such models while maintaining competitive training and rendering times. Inspired by the Sparse Mixture of Experts (MoE) paradigm [28], we have developed a model-agnostic framework capable of improving the reconstruction quality of various state-of-the-art models. Intuitively, our mixture of experts consists of different-capacity (resolution) models. During training, each model

specializes in rendering the most suitable parts of the scene, *i.e.* low-resolution models render low-frequency parts of the scene, while high-resolution models render high-frequency parts. Our formulation of the gate allows our method to be model-agnostic, enabling, on the one hand, the insertion of the gate function at the early stages of the MoE mechanism and, on the other hand, the multiplication of the gate's output with the experts' output as late as possible. This maximizes the capabilities of the mixture of experts and allows for working directly with the output of each expert, which is considered a black box. This is why our method is inherently model-agnostic. In summary, our contributions are as follows:

1. We propose the first model-agnostic framework based on Sparse MoE of models at different resolutions, which significantly improves the rendering quality of such models while maintaining competitive training and inference times (Sec. 3).
2. We provide a novel gate formulation inspired by Fast-NeRF models, which maximizes the capability of the mixture of experts (Sec. 3.3).
3. We introduce a new routing technique based on resolution, favoring the assignment of tokens to low-resolution models and discouraging the assignment of tokens to high-resolution models, inducing increasing sparsity in high-resolution models and decomposing the scene based on frequency (Sec. 3.5).
4. We conduct extensive experiments to test our method, including different NeRFs architectures and datasets, showing higher rendering quality and efficiency (Sec. 4).

2 Related works

2.1 Neural Radiance Field

NeRF [24] (Neural Radiance Field) has emerged as a prominent method for synthesizing novel views, showing significant progress. This approach requires a moderate number of input images along with their known camera poses. Unlike traditional methods that rely on explicit and discretized volumetric representations such as voxel grids and multiplane images, NeRF employs a coordinate-based multilayer perceptron (MLP) to create an implicit and continuous volumetric representation. NeRFs can represent a 3D scene as a MLP F_θ, with θ being the set of its trainable parameters, such that $F_\theta : (\mathbf{x}, \mathbf{d}) \rightarrow (\mathbf{c}, \sigma)$ which maps $(\mathbf{x}, \mathbf{d})$, a 3D position $\mathbf{x}$ and a viewing direction $\mathbf{d}$, to a view-dependent color emission $\mathbf{c}$ and density value σ. To render the color of a pixel $\hat{C}(\mathbf{r})$, a ray $\mathbf{r}$ traverses the center of the camera through the pixel of the image plane from an origin point $\mathbf{o}$ to a ray having position $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$. Then, N points are sampled on $\mathbf{r}$, and the MLP is queried for each point, obtaining a density and a color value. Finally, these results are accumulated into a single color with the volume rendering equation [22]:

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{K} T_i \alpha_i c_i, \qquad T_i = \prod_{j=1}^{i-1} (1 - \alpha_j), \qquad \alpha_i = 1 - \exp(\sigma_i \delta_i), \tag{1}$$

where α_i is the probability of termination at point i and T_i is the accumulated transmittance from the near plane to point i. NeRFs are trained by minimizing a photometric loss, which is an L2 loss between the rendered and the ground truth pixels. In more detail, given a batch B of randomly sampled rays, the loss is defined as

$$L_{\text{nerf}} = \frac{1}{|B|} \sum_{r \in B} \left\| \hat{C}(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2, \tag{2}$$

where $\hat{C}(\mathbf{r})$ is the predicted color and $C(\mathbf{r})$ the ground truth color for the ray $\mathbf{r}$.

2.2 Fast Neural Radiance Fields

Since the introduction of NeRF, significant effort has been directed towards the development of faster models. Several studies focus on speeding up rendering times, for example by working on ray sampling efficiency [3, 6, 12, 37], by integrating explicit volumetric representations [5, 11, 11, 15, 31–33] or by utilizing thousands of tiny MLPs [26]. However, all these methods still require large training times. One noteworthy development is represented by the introduction of explicit volumetric representations in the training pipeline, directly optimizing such representations [4,9,10,13,16,25,29]. These models enable fast training and inference, with render quality only slightly inferior to full-implicit models [1, 2]. We refer to these as *Fast-NeRFs*. Plenoxels [10] is the first important work in this context, as it represents a scene as a sparse 3D grid where each voxel stores spherical harmonic coefficients and density. Spherical harmonics serve as an orthogonal basis for functions defined over the sphere and thus can be used for computing view-dependent color emission, without the need for a multi-layer perceptron. Additionally, Plenoxels demonstrated the advantages of linearly interpolating voxels, facilitating the learning of a continuous plenoptic function throughout the volume, akin to NeRF, albeit with discrete data. Another notable work is DVGO [29]. Each scene is there represented as two dense voxel grids (one for density and one for feature colors) alongside a MLP, for learning view-dependent color. Similar to Plenoxels, the value of each voxel is linearly interpolated with the 8 nearest voxels, but after applying the activation functions. DVGO also incorporates a preliminary coarse geometry stage to learn the scene's general structure, facilitating adjustments to the bounding box and enabling a more intelligent ray sampling strategy. Despite utilizing lower-resolution models, DVGO achieves high reconstruction quality. The major drawback of these models is their substantial memory storage requirements. TensoRF [4] addresses this challenge by replacing the dense voxel grid with a planes and vectors decomposition, significantly reducing storage demands while maintaining comparable performance and rendering quality. Alternatively, Instant-NGP [25] proposed a multi-resolution voxel grid encoded via a hash function. They define L hash grids of increased resolutions: each entry of each hash grid has 2^T parameters and F features. This enables even faster training times and real-time rendering

performances while maintaining a compact model; moreover, it demonstrates the efficacy of a multi-resolution approach in enhancing render quality. Inspired by this methodology, K-Planes [9] introduces a multi-resolution planar factorization of 3D space, offering reconstruction quality and model compactness similar to the previous methods, but with slightly larger training times. A similar multi-resolution planar factorization has also been proposed in Tri-MipRF [13] for mitigating the aliasing in distant or low-resolution views and blurriness in close-up shots. Given their properties, we decided to evaluate our paradigm using three different models: a 3D dense voxel grid-based model (DVGO), a decomposed grid-based model (TensoRF), and a multi-resolution hash grid-based model (Instant-NGP).

2.3 Mixture of Experts and Sparse MoE

The Mixture of Experts (MoE) paradigm [14] has gained prominence in various machine learning applications. MoE consists of multiple expert networks, each specializing in different regions of the input space, with a meta-network determining the contribution of each expert to the final prediction. Building upon the MoE framework, Sparse Mixture of Experts [19, 27, 28, 36] constitutes a scalable and efficient variant. At the core of all Sparse MoE algorithms lies an assignment problem between tokens and experts. One approach to tackle this is by approximating the solution with a gating function, which learns to assign input tokens to the most suitable experts. It typically comprises a linear layer, a softmax activation, and a Top-K (where $1 \leq K \leq 2$) operation, aimed at routing the input token to only a subset of the experts. To balance the assignment across all the experts, auxiliary loss functions penalizing unevenly distributed routing are often employed. This sparsity-inducing technique significantly reduces computational overhead while preserving the expressive power of the MoE architecture.

2.4 MoE and NeRF

Combining the sparse mixture of experts' paradigms with neural radiance fields presents a non-trivial challenge. In recent research, Switch-NeRF [35] has been introduced as a novel end-to-end large-scale NeRF with learning-based scene decomposition. Inspired by Fedus *et al.* [8], they propose a full-implicit model with an MLP-based gate comprising 4 layers with 128 neurons and a Top-1 function. However, their architecture is ad-hoc, and suffers from extensive training times. This poses a challenge for us, as we aim to consider each expert as a black box, taking a point in space and a direction as input and outputting radiance and density values. Moreover, as the mixture comprises experts at different resolutions, we also aim to prioritize routing input tokens toward lower-resolution models and minimize the usage of high-resolution models.

Our work intends to overcome these limitations. We design a novel gate formulation inspired by Fast-NeRFs, that guarantees fast convergence and better performances. We position the gate at the beginning of our MoE pipeline and postpone the multiplication of the gate output with expert predictions as much as

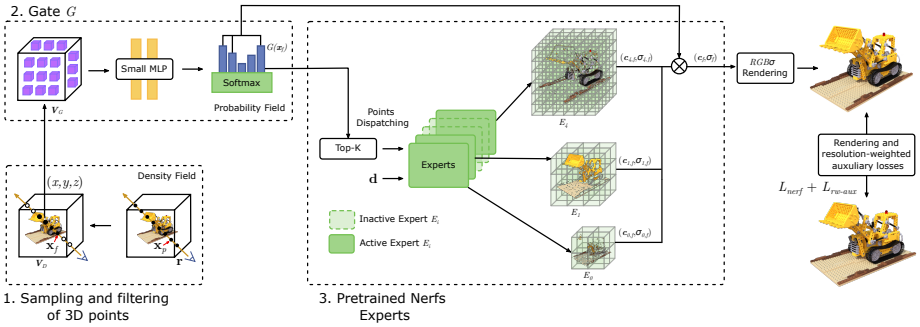


Fig. 2: A density field is used to compute density values for sampled points along a ray. A filtering step discards low-density points, routed through gating network G for expert assignment. Top-K experts compute radiance and density, aggregating and weighting these values by the corresponding gating probability to get the final values $\mathbf{c}_f$ and σ_f. Volume rendering equation yields pixel colors, and joint optimization refines our resolution-weighted auxiliary loss, allowing for high-quality and efficient rendering.

possible. This allows our framework to be entirely model-agnostic. Furthermore, our mixture of experts maintains competitive training times, enabling, in the worst case, training models on complex scenes in approximately 1 hour while achieving state-of-the-art accuracy.

3 Method

In this section, we introduce our model-agnostic ensembling approach. We present in Sec. 3.1 an overview of our method. We also describe the ray sampling and filtering mechanism in Sec. 3.2. Then, in Sec. 3.3, we discuss the effective construction of a mixture of experts with a gating mechanism in the context of NeRF. Finally, in Sec. 3.5, we present a novel formulation for decomposing the scene into resolution-based parts with an end-to-end training procedure.

3.1 Overview

Figure 2 shows a high-level scheme of our method. After independently training a set of M NeRF models $\{E_i\}_{i=1}^{M}$ at different resolutions, we sample N points along a ray $\mathbf{r}$. For each point $\mathbf{x}_p$, a density value σ is calculated, and based on this, a filtering step is performed based on the density volume $\mathbf{V}_D$, discarding points in regions with a density below a certain threshold T. Initializing the density volume $\mathbf{V}_D$ with the one learned from the lowest resolution model can be helpful (though it can also be learned). Subsequently, the filtered point $\mathbf{x}_f$ is fed to the gate G, so we denote with $G(\mathbf{x}_f)$ the probabilities with which the point is assigned to the M experts. Based on these values, each point is routed to the Top-K experts. Each selected expert then computes the radiance $\mathbf{c}_{i,f}$ and density values $\sigma_{i,f}$ for the point, both of which are multiplied by their respective

probabilities and summed together to get the final radiance $\mathbf{c}_f$ and density σ_f. Subsequently, the volume rendering Equation 1 is used to aggregate all colors and compute the pixel color for the given ray. Finally, we compute the total loss and jointly optimize the gate and the experts.

In the next sections, we will dive into the gating and sparse mixture of expert modules.

3.2 Ray Sampling and Filtering

The first phase of our method involves learning a coarse and explicit density field starting from the density volume $\mathbf{V}_D$, that we can leverage for skipping empty spaces. Given a ray $\mathbf{r}$ as explained in Sec. 2.1, we sample N points along it:

$$\mathbf{X}_{p,r} = \text{sampling}(\mathbf{r}) \in \mathbb{R}^{N\times 3}, \tag{3}$$

with $p \in [0, ..., N-1]$. We suppress the index r for abuse of notation. Next, we compute the density for each point $\mathbf{x}_p \in \mathbb{R}^3$ and discard those with negligible density. The density value $\sigma_{D,p}$ for each point is computed by linearly interpolating v neighboring voxels:

$$\sigma_{D,p} = \text{act}(\text{interpolate}(\mathbf{x}_p, V_D)) \in \mathbb{R} \tag{4}$$

where act represents a density activation function (such as softplus). We denote with $\mathbf{x_f} \in R^3$ a remaining point after the filtering operation.

3.3 Trainable Gating Model

Our gating mechanism incorporates a hybrid architecture: an explicit feature grid $\mathbf{V}_G$ and a shallow MLP. The gating mechanism can also be seen as a probability field. First, we compute per-point features:

$$\mathbf{feat} = \text{interpolate}(\mathbf{x}_f, \mathbf{V}_G) \in \mathbb{R}^C \tag{5}$$

where C represents the number of channels of $\mathbf{V}_G$. Subsequently, we transform each feature into a probability. We compute logits as

$$\mathbf{logits} = \text{MLP}(\mathbf{feat}) \in \mathbb{R}^M \tag{6}$$

where M is the number of experts in the mixture of experts. Finally, we apply a per-row softmax. Our gating function can be summarized as:

$$G(\mathbf{x_f}) = \text{softmax}(\text{MLP}(\text{interpolate}(\mathbf{x}_f, \mathbf{V}_G))) \in \mathbb{R}^M \tag{7}$$

3.4 NeRFs Experts

As mentioned in Sec. 3.1, we employ a set of M pretrained NeRFs models as experts. After feeding as input each point $\mathbf{x}_f$ to the gate, we route them to the k experts with the k highest probabilities $G(\mathbf{x}_f)$. We define this set of indexes of the selected experts as

$$\mathcal{K} = \underset{Top-k}{\text{argmax}}(G(\mathbf{x}_f)). \tag{8}$$

Each expert is a Fast-NeRF model that receives the dispatched points and the direction of the ray they lie on as input and outputs a density and a radiance value. We treat each expert as a black box, ensuring our method is inherently model-agnostic. The radiance $\mathbf{c}_f$ and the density σ_f for the point $\mathbf{x}_f$ laying on a ray with direction $\mathbf{d}$ can be expressed as the sum of each expert predictions weighted by their probability:

$$\mathbf{c}_f, \sigma_f = \sum_{i \in \mathcal{K}} E_i(\mathbf{x_f}, \mathbf{d}) \cdot G(\mathbf{x_f})_i \tag{9}$$

where E_i denotes the i-th expert and $G(\mathbf{x_f})_i$ the probability for the point to be dispatched to that expert i.

Once we obtain radiance and density values for all the points on the ray, we can compute the pixel color with Equation 1.

3.5 Resolution-based Routing

To balance the load and prevent the gate from focusing on assigning points to a single expert (typically the one with the highest resolution), we employ a resolution-weighted auxiliary loss. Given M experts, and a batch of points B, the auxiliary loss is defined as:

$$L_{\text{aux}} = \frac{M}{|B|^2} \sum_{i=1}^{M} c_i m_i, \qquad m_i = \sum_{\mathbf{x}_f \in B} G(\mathbf{x}_f)_i, \tag{10}$$

where c_i represents the number of inputs dispatched to the expert i, and m_i is the sum of all the probabilities for each point in the batch for the expert i. This loss helps balance the workload, ensuring that each expert processes a similar number of points. Ideally, in a perfectly balanced scenario, L_{aux} is expected to be 1, as both c_i and m_i would be $\frac{M}{N}$, resulting in $\sum_{i=1}^{M} c_i m_i$ being $\frac{M^2}{N}$. However, in the case of total imbalance, it tends to the number M. We aim to take a step further by assigning as many points as possible to low-resolution experts while discouraging point assignment to high-resolution models. Hence, we introduce some penalty terms w_i associated with each expert. The higher the resolution of the model, the higher the penalty. We define a novel resolution-based auxiliary loss:

$$L_{\text{rw-aux}} = \frac{M}{|B|^2} \sum_{i=1}^{M} c_i m_i w_i. \tag{11}$$

As for the weighting strategy, we opted for the following geometric progression. The weight w_i for the i expert is computed as:

$$w_i = \exp\left(\frac{\ln M}{M-1}\right)^i, \quad i \in [0, ..., M-1]. \tag{12}$$

The total loss is then defined as:

$$L_{\text{tot}} = L_{\text{nerf}} + \lambda L_{\text{rw-aux}}. \tag{13}$$

As we will see in Sec. 4, our proposed loss not only further improves the reconstruction quality but also encourages sparsity in higher-resolution models. A comparison with other weighting strategies is proposed in the Supplementary **??**.

4 Experiments

4.1 Setting

The code was written in Python 3.8, using PyTorch 1.12, and executed on a single NVIDIA A40 GPU. We tested our technique on the DVGO, TensoRF, and Instant-NGP architecture. The code from official repositories was used as the starting point. For Instant-NGP, the native implementation in PyTorch `ngp_pl` was employed, which exhibits comparable performance and reconstruction quality to the official NVIDIA implementation. We experimented with various configurations, ranging from a minimum of 3 models to 5 models of different resolutions. The experts are ordered by resolution, such that the $i+1$-th expert has approximately double the parameters of the i-th expert. For all configurations and models, λ was set to 10^{-3}. The Gate consists of a grid (dense voxel grid for DVGO, factorized grid for TensoRF, and hash grid for Instant-NGP) and a shallow MLP (2 layers with 64 neurons each) with ReLU activation function. The resolution of the gate is low: 128^3 for both DVGO and TensoRF, while for Instant-NGP we used $L = 6$. The number of iterations is set to $20k$ for all the architectures. All other hyperparameters are left as the original implementations. We draw experiments with our method using Top-k experts, with $k = 1$ and $k = 2$. We compare our results versus baselines with comparable resolutions, as well as a Fast-NeRF ensemble (*Ens*) obtained by jointly fine-tuning all models and averaging their predictions. It can be noted that this ensemble can be interpreted as a limit case for our method with $k = M$ and $G(\mathbf{x}_f)_i = 1/M, \forall i$ similar to the method proposed by [7].

4.2 Metrics and Datasets

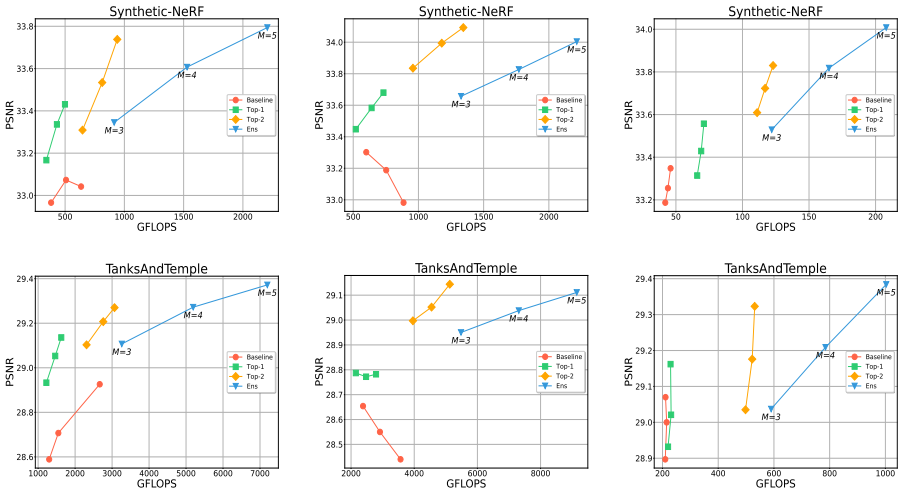
For each test, image-quality metrics such as PSNR, SSIM [30], and LPIPS [34] (computed on AlexNet [18]) are presented. Additionally, we report the number of non-zero parameters as $\|w_0\|$, the average GFLOPs required by each model to render the images of the test set, and total training times. We present results obtained across four major datasets, namely: Synthetic-NeRF [24], Neural Sparse Voxel Field Dataset (NSVF) [20], TanksAndTemple [17], and Local Light Field Fusion Dataset (LLFF) [23].

Table 1: Results on DVGO, TensoRF and Instant-NGP with $M = 5$. $\|w\|_0$ is expressed as multiple of 10^6, while for Instant-NGP as a multiple of 10^5.

Dataset	Metrics	***DVGO***				***TensoRF***				***Instant-NGP***			
		baseline	*Top-1*	*Top-2*	*Ens*	*baseline*	*Top-1*	*Top-2*	*Ens*	*baseline*	*Top-1*	*Top-2*	*Ens*
Blender	**PSNR** ↑	33.04	33.43	33.74	**33.79**	32.98	33.68	**34.09**	34.00	33.35	33.56	33.83	**34.01**
	SSIM ↑	0.961	0.964	**0.965**	**0.966**	0.958	0.965	**0.968**	0.968	0.963	0.963	0.965	**0.966**
	LPIPS ↓	0.026	0.024	**0.022**	**0.022**	0.029	0.023	**0.021**	0.022	**0.025**	0.045	0.043	0.042
	$\|w\|_0$ ↓	99	**26**	39	97	40	**24**	33	49	31	**17**	21	26
	GFLOPs ↓	635	**499**	940	2206	886	**732**	1344	2214	**46**	71	123	208
	Time ↓	26’	**21’**	25’	32’	**44’**	69’	76’	70’	**11’**	32’	34’	37’
NSVF	**PSNR** ↑	35.21	37.12	37.59	**37.68**	36.70	37.40	37.98	**38.08**	36.44	33.59	37.04	**37.31**
	SSIM ↑	0.977	0.984	**0.986**	**0.986**	0.981	0.984	0.986	**0.987**	0.983	0.983	0.984	**0.985**
	LPIPS ↓	0.015	0.009	0.008	**0.007**	0.013	0.009	**0.008**	**0.008**	**0.010**	0.023	0.022	0.020
	$\|w\|_0$ ↓	95	**27**	43	100	42	**28**	38	54	30	**17**	20	27
	GFLOPs ↓	564	**430**	811	1903	706	**575**	1053	1763	**28**	52	72	123
	Time ↓	**12’**	22’	25’	30’	**46’**	75’	75’	75’	**10’**	31’	33’	34’
TaT	**PSNR** ↑	28.93	29.14	29.27	**29.37**	28.44	28.78	**29.14**	29.11	29.07	29.16	29.32	**29.38**
	SSIM ↑	0.927	0.925	0.929	**0.932**	0.905	0.924	**0.929**	0.928	0.924	0.927	0.929	**0.930**
	LPIPS ↓	0.107	0.108	0.105	**0.103**	0	0.106	**0.099**	0.109	**0.101**	0.125	0.124	0.121
	$\|w\|_0$ ↓	74	**16**	26	65	**7**	9	15	80	88	**37**	47	70
	GFLOPs ↓	2666	**1626**	3066	7198	3567	**2791**	5126	9146	**211**	229	531	1003
	Time ↓	**22’**	25’	28’	38’	72’	**70’**	78’	101’	**14’**	37’	39’	45’
LLFF	**PSNR** ↑	26.24	26.43	26.62	**26.65**	26.71	26.73	27.09	**27.10**	24.97	24.90	**25.17**	25.19
	SSIM ↑	0.831	0.832	0.839	**0.843**	0.835	0.836	0.862	**0.864**	0.764	0.763	**0.777**	0.778
	LPIPS ↓	0.136	0.115	0.111	**0.107**	0.114	0.111	**0.101**	**0.101**	**0.128**	0.239	0.237	0.234
	$\|w\|_0$ ↓	62	**26**	40	113	19	**11**	16	23	152	**68**	75	148
	GFLOPs ↓	1678	**1514**	2508	4972	4542	**3226**	5921	13522	573	**887**	1391	2712
	Time ↓	**24’**	28’	32’	36’	58’	**49’**	57’	68’	**24’**	25’	46’	42’

4.3 Quantitative Results

The main experimental results in terms of the metrics defined above are shown in Table 1. Here, we present the results obtained using a mixture of $M = 5$ experts; further configurations can be found in the supplementary Sec. **??**. Our experiments reveal that our MoE provides a significant rendering quality improvement with respect to the baseline with no or limited impact in terms of computational cost. The increase in rendering quality is notable in Synthetic NeRF (up to 1 dB) and NSVF (up to 1.3 dB). Although more moderate, improvements are still evident, even in more challenging datasets such as TanksAndTemple and LLFF, with about 0.5 dB gain. While Top-1 can already achieve state-of-the-art render quality, the Top-2 strategy further enhances image quality, reaching levels comparable to the ensemble but with much greater efficiency (about half the average FLOPs per rendering) and using significantly fewer parameters. This observation is also illustrated in Figure 3 , where FLOPS/PSNR plots are shown (each marker in every curve refers to the cases with $M = 3, 4, 5$ experts respectively). Training times are on average are longer but still acceptable (in the worst-case scenario, around 1h is required to train our MoE). However, in the case of Top-1, they can be faster than baselines. This is because of our resolution methods, which tend to favor low-resolution models. Based on these analyses, the Top-2 strategy strikes an excellent quality/cost trade-off. The ensembling configura-



(a) DVGO **(b)** TensoRF **(c)** Instant-NGP

Fig. 3: PSNR/GFLOPs plots for Synthetic-NeRF and TanksAndTemple. The remaining datasets show similar results.

tion, using all experts, represents an upper bound in terms of image quality while being the worst case in computational terms.

4.4 Qualitative Results

Figure 4 shows visual comparisons among the baseline, ensemble, Top-1, and Top-2. From the selected image crops, one can appreciate that our method ensures superior reconstruction quality compared to the baselines, effectively reproducing sharper details while reducing noise on texture-less spots. This is particularly notable when examining elements such as the window decorations in the Palace scene or the text on the box above the desk in the Room scene. Additionally, surfaces such as the semi-transparent glass in the Wineholder or the Caterpillar scene appear less noisy and more faithful to the original. This improvement can be attributed to the synergy among different resolution models: low-resolution models excel at representing lower frequencies, thereby introducing less noise, while high-resolution models can focus solely on high-frequency components. Additionally, it is important to notice that there is no significant difference between the ensemble and Top-2.

4.5 Comparison with Naive Methods

As shown in Figure 1, our method leads to significantly higher quality reconstructions at a greatly reduced computational cost. Decreasing the step size yields the least noticeable improvement while incurring a substantial computational

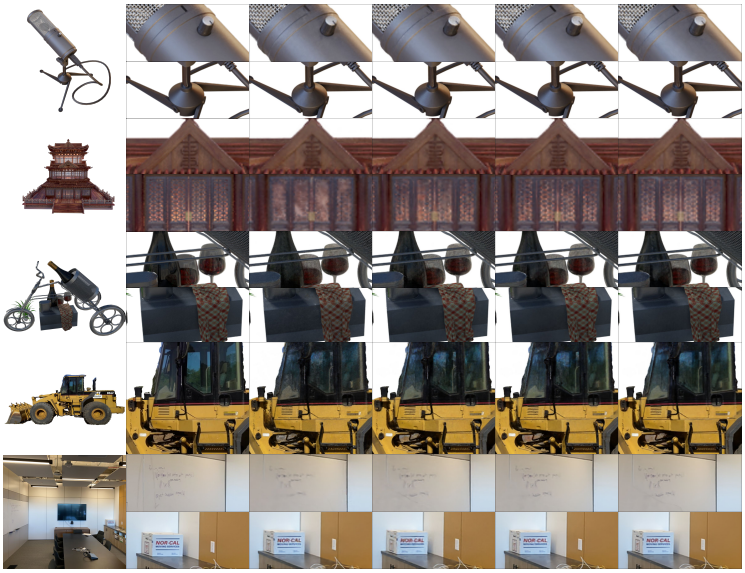


Fig. 4: Qualitative results on some of the scenes of each dataset. From left to right: ground truths, baselines, Top-1, Top-2 and Ensemble. Each model has the same parameters and has been trained for the same number of iterations (DVGO).

expense, as each sampled point requires an evaluation by the color decoder. Our method achieves high-quality reconstructions with the same step size as the baseline models. Increasing the MLP parameters can boost accuracy, but again, the computational costs become substantial compared to our method. Similarly, increasing resolution results in way inferior PSNR with respect to our method at a comparable computational cost. Our method also allows scaling to higher resolutions, while baseline models tend to introduce noise and artifacts, leading to a decrease in reconstruction quality as the resolution increases.

4.6 Gate Visualization and Scene Decomposition

In Figure 5, we visualize the gate (probability field in grayscale) and each expert output in the case $M = 3$. On the right side of the figure, one can appreciate per-model renders and experts' specialization. It is worth noting how higher-resolution experts render high-frequency details. This is particularly evident in the *Mic* scene.

4.7 Ablation

Here we investigate how different design choices of our model affect the performance and rendering quality. These choices include the gate resolution, different gate formulations, and the number of experts in the Top-k.

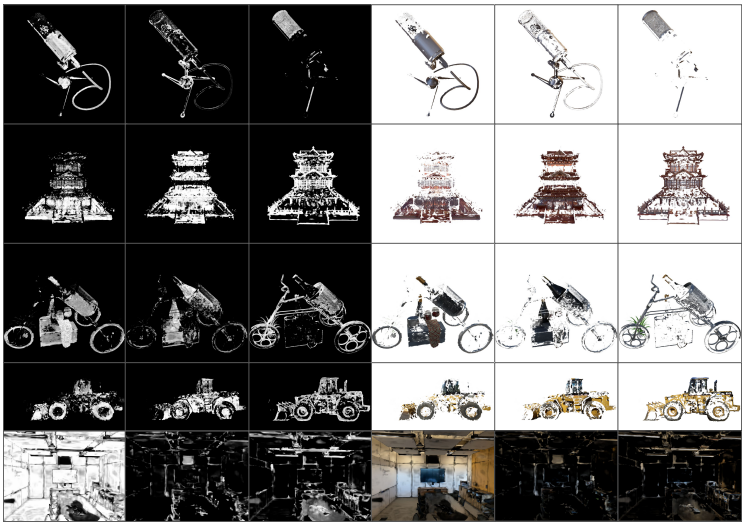


Fig. 5: Gate (gray-scale images) and per-model output visualizations with 3 experts and resolution-based routing (DVGO). Images are ordered by increasing resolutions.

Here we investigate how different design choices of our model affect the performance and rendering quality. These choices include the gate resolution, different gate formulations, and the number of experts in the Top-k.

Gate Resolution and Gate Formulation. We show in Table 2 that a low-resolution gate is sufficient to achieve high-quality rendering. Interestingly, as the gate resolution increases, there is a slight decrease in rendering quality during testing, coupled with an increase in required FLOPs per image rendering and number of parameters. We also compare various gating strategies, including linear gating, the configuration proposed by Switch-NeRF, and our gate formulation. Through experimentation, it becomes evident that our approach outperforms others in terms of both render quality and performance, achieving performances comparable to a linear gating function.

Why Top-2? Here we investigate the performance and rendering quality trends with varying values of k. In Figure 6, we can see that while $k = 1$ can lead to a significant increase in quality with comparable performance to baseline models, the Top-2 further enhances quality at the expense of increased (but still acceptable) computation. The Top-3, Top-4, and Top-5 provide marginal quality improvements at a significantly higher computational cost. Hence, we consider the Top-2 the optimal balance between performance and quality.

Res	Gate Type	PSNR	$\|w\|_0$	GFLOPs
128^3	Ours	36.8	26	476
256^3	Ours	36.77	33	601
300^3	Ours	36.79	42	677
-	Linear	36.25	24	444
-	Switch-NeRF	36.44	25	1095

Table 2: Comparison of different gate resolutions and formulations on Lego scene and DVGO.

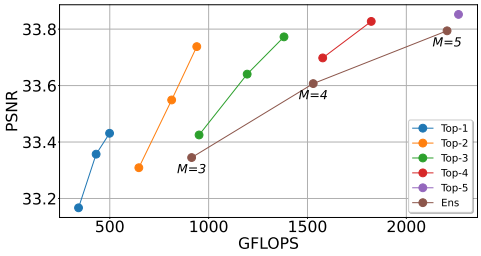


Fig. 6: Comparison of Gating Functions (Top-1 to Top-5) on Synthetic-NeRF with DVGO.

5 Limitations

Our study present some limitations. First of all, a pre-training phase, on which each model is trained *independently* is required for achieving good results. Training end-to-end, without a pre-training phase, can lead to reconstruction quality that is noticeably inferior to baselines. Pre-training models at different resolutions allows for diversified architectures, making it easier for the gate to learn the decomposition. The training is also strongly dependent on the auxiliary loss. Different values of λ can significantly influence performances and load balancing. We conducted a sweep to identify a value that works well across many scenarios, but it may not be suitable for different datasets. Another limitation is the overhead introduced by the MoE. Each input token is first interpolated with the gate's grid, decoded into probabilities and routed to the chosen expert. These operations can significantly slow-down the rendering process, leading to higher training and inference times. Although considerable effort was devoted to developing the most efficient gate possible, our MoE is still slower than the respective baselines.

6 Conclusions

In this paper, we introduced a model-agnostic framework for enhancing the rendering of Fast-NeRF models. Our formulation of the Gate reduces computational costs in both training and inference phases while ensuring better quality compared to a traditional gate. Additionally, the introduction of an auxiliary loss with res penalty allows for increased utilization of low-resolution models, reducing the number of active parameters and promoting sparsity in higher-resolution models. Our results demonstrate how this approach can significantly improve reconstruction quality while considering performance metrics. Specifically, we show that a Top-2 strategy strikes a good balance between performance and quality.

Acknowledgements

This work was partially funded by Hi!PARIS Center on Data Analytics and Artificial Intelligence.

References

1. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5855–5864 (2021)
2. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Zip-nerf: Anti-aliased grid-based neural radiance fields. arXiv preprint arXiv:2304.06706 (2023)
3. Chan, E.R., Monteiro, M., Kellnhofer, P., Wu, J., Wetzstein, G.: pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5799–5809 (2021)
4. Chen, A., Xu, Z., Geiger, A., Yu, J., Su, H.: Tensorf: Tensorial radiance fields. ECCV (2022)
5. Chen, Z., Funkhouser, T., Hedman, P., Tagliasacchi, A.: Mobilenerf: Exploiting the polygon rasterization pipeline for efficient neural field rendering on mobile architectures. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16569–16578 (2023)
6. Deng, Y., Yang, J., Xiang, J., Tong, X.: Gram: Generative radiance manifolds for 3d-aware image generation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10673–10683 (2022)
7. Di Sario, F., Renzulli, R., Tartaglione, E., Grangetto, M.: Two is better than one: Achieving high-quality 3d scene modeling with a nerf ensemble. In: International Conference on Image Analysis and Processing. pp. 320–331. Springer (2023)
8. Fedus, W., Zoph, B., Shazeer, N.: Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. The Journal of Machine Learning Research **23**(1), 5232–5270 (2022)
9. Fridovich-Keil, S., Meanti, G., Warburg, F.R., Recht, B., Kanazawa, A.: K-planes for radiance fields in space, time, and appearance (2023)
10. Fridovich-Keil, S., Yu, A., Tancik, M., Chen, Q., Recht, B., Kanazawa, A.: Plenoxels: Radiance fields without neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5501–5510 (2022)
11. Garbin, S.J., Kowalski, M., Johnson, M., Shotton, J., Valentin, J.: Fastnerf: High-fidelity neural rendering at 200fps. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14346–14355 (2021)
12. Hou, A., Liu, F., Ren, Z., Sarkis, M., Bi, N., Tong, Y., Liu, X.: Infamous-nerf: Improving face modeling using semantically-aligned hypernetworks with neural radiance fields. arXiv preprint arXiv:2312.16197 (2023)
13. Hu, W., Wang, Y., Ma, L., Yang, B., Gao, L., Liu, X., Ma, Y.: Tri-miprf: Tri-mip representation for efficient anti-aliasing neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 19774–19783 (2023)

14. Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive mixtures of local experts. Neural computation **3**(1), 79–87 (1991)
15. Jeong, Y., Ahn, S., Choy, C., Anandkumar, A., Cho, M., Park, J.: Self-calibrating neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5846–5854 (2021)
16. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4) (July 2023)
17. Knapitsch, A., Park, J., Zhou, Q.Y., Koltun, V.: Tanks and temples: Benchmarking large-scale scene reconstruction. ACM Transactions on Graphics (ToG) **36**(4), 1–13 (2017)
18. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems **25** (2012)
19. Lepikhin, D., Lee, H., Xu, Y., Chen, D., Firat, O., Huang, Y., Krikun, M., Shazeer, N., Gshard, Z.: Scaling giant models with conditional computation and automatic sharding. arXiv preprint arXiv:2006.16668 (2020)
20. Liu, L., Gu, J., Zaw Lin, K., Chua, T.S., Theobalt, C.: Neural sparse voxel fields. Advances in Neural Information Processing Systems **33**, 15651–15663 (2020)
21. Liu, Y., Li, X., Yu, F., Zhou, Q.: Probabilistic neural scene representations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2021)
22. Max, N.: Optical models for direct volume rendering. IEEE Transactions on Visualization and Computer Graphics **1**(2), 99–108 (1995)
23. Mildenhall, B., Srinivasan, P.P., Ortiz-Cayon, R., Kalantari, N.K., Ramamoorthi, R., Ng, R., Kar, A.: Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. ACM Transactions on Graphics (TOG) **38**(4), 1–14 (2019)
24. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: European conference on computer vision. pp. 405–421. Springer (2020)
25. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM Trans. Graph. **41**(4), 102:1–102:15 (Jul 2022)
26. Reiser, C., Peng, S., Liao, Y., Geiger, A.: Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14335–14345 (2021)
27. Riquelme, C., Puigcerver, J., Mustafa, B., Neumann, M., Jenatton, R., Susano Pinto, A., Keysers, D., Houlsby, N.: Scaling vision with sparse mixture of experts. Advances in Neural Information Processing Systems **34**, 8583–8595 (2021)
28. Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., Dean, J.: Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. arXiv preprint arXiv:1701.06538 (2017)
29. Sun, C., Sun, M., Chen, H.T.: Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5459–5469 (2022)
30. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004)

31. Wizadwongsa, S., Phongthawee, P., Yenphraphai, J., Suwajanakorn, S.: Nex: Real-time view synthesis with neural basis expansion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8534–8543 (2021)
32. Xian, W., Huang, J.B., Kopf, J., Kim, C.: Space-time neural irradiance fields for free-viewpoint video. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9421–9431 (2021)
33. Yu, A., Li, R., Tancik, M., Li, H., Ng, R., Kanazawa, A.: Plenoctrees for real-time rendering of neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5752–5761 (2021)
34. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)
35. Zhenxing, M., Xu, D.: Switch-nerf: Learning scene decomposition with mixture of experts for large-scale neural radiance fields. In: The Eleventh International Conference on Learning Representations (2022)
36. Zhou, Y., Lei, T., Liu, H., Du, N., Huang, Y., Zhao, V., Dai, A.M., Le, Q.V., Laudon, J., et al.: Mixture-of-experts with expert choice routing. Advances in Neural Information Processing Systems **35**, 7103–7114 (2022)
37. Zhuang, Y., Zhu, H., Sun, X., Cao, X.: Mofanerf: Morphable facial neural radiance field. In: European conference on computer vision. pp. 268–285. Springer (2022)