NGP-RT: Fusing Multi-Level Hash Features with Lightweight Attention for Real-Time Novel View Synthesis

Yubin Hu[1][*][†], Xiaoyang Guo[2][*], Yang Xiao[3],
Jingwei Huang[4], and Yong-Jin Liu[1][✉]

[1] BNRist, Department of Computer Science and Technology, Tsinghua University
[2] Horizon Robotics   [3] Huawei Technologies   [4] Game AI Center, Tencent Games

Abstract. This paper presents NGP-RT, a novel approach for enhancing the rendering speed of Instant-NGP to achieve real-time novel view synthesis. As a classic NeRF-based method, Instant-NGP stores implicit features in multi-level grids or hash tables and applies a shallow MLP to convert the implicit features into explicit colors and densities. Although it achieves fast training speed, there is still a lot of room for improvement in its rendering speed due to the per-point MLP executions for implicit multi-level feature aggregation, especially for real-time applications. To address this challenge, our proposed NGP-RT explicitly stores colors and densities as hash features, and leverages a lightweight attention mechanism to disambiguate the hash collisions instead of using computationally intensive MLP. At the rendering stage, NGP-RT incorporates a pre-computed occupancy distance grid into the ray marching strategy to inform the distance to the nearest occupied voxel, thereby reducing the number of marching points and global memory access. Experimental results show that on the challenging Mip-NeRF 360 dataset, NGP-RT achieves better rendering quality than previous NeRF-based methods, achieving **108 fps** at **1080p** resolution on a single Nvidia RTX 3090 GPU. Our approach is promising for NeRF-based real-time applications that require efficient and high-quality rendering.

Keywords: Neural Radiance Field · Novel View Synthesis · Real-time Rendering

1 Introduction

Novel view synthesis is a long-standing problem in computer vision. The emergence of neural radiance fields [1,2,25,48] has led to state-of-the-art photorealistic view synthesis using the neural volumetric scene representation. However, in these pioneering works, the original MLP parameterization exhibits high computational complexity. To address this limitation, recent works have proposed

---

[*] These authors contributed equally to this work.
[†] The work was done while the author was an intern at Huawei.
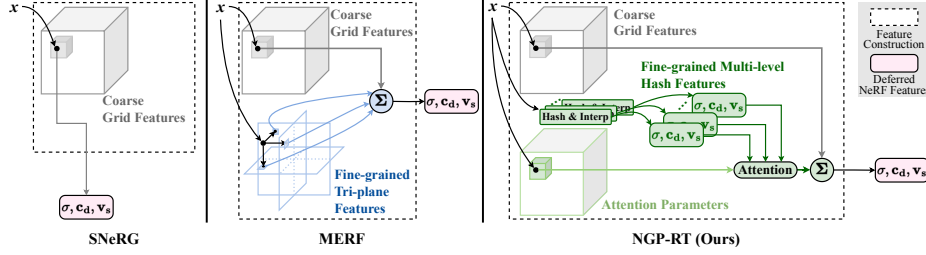[✉] Corresponding author.

Fig. 1: Comparisons of the feature construction methods in SNERG [12], MERF [31], and our NGP-RT. We utilize a lightweight attention mechanism to efficiently aggregate the multi-level explicit hash features, which sufficiently exploit the expressing power of NGP-style features under the deferred NeRF architecture for fast rendering.

replacing or augmenting the MLPs with feature grids [8, 34], tri-planes [15], multi-plane images [11, 20] or tensorial vectors [5, 9].

Among them, Instant-NGP [26] stands out as a promising technique that utilizes multi-level implicit feature grids to represent the radiance field and employs a shallow MLP to decode implicit features into explicit colors and densities. By storing high-resolution implicit features in limited-length hash tables, Instant-NGP achieves high-quality novel view synthesis after a short optimization period. Despite its advantages in training speed and rendering quality, Instant-NGP cannot meet the real-time rendering requirements for large scenes. The shallow MLP for feature aggregation of each querying point slows down the rendering speed, even though accessing the multi-level grid features is relatively fast. However, simply reducing the MLP parameters or removing the MLP would result in severe rendering quality degradation.

Recent works, such as SNeRG [12] and MERF [31], proposed a deferred NeRF architecture to remove the per-point MLP. They directly store explicit colors and densities instead of implicit features, and only execute MLP once for each casting ray, which highly accelerates the rendering process. However, their treatments of the explicit features show limited expressing power and are unsuitable for multi-level features of Instant-NGP. As shown in Figure 1, the low-resolution sparse grid of SNeRG fails to cover the detailed features at fine-grained resolution levels of Instant-NGP. At the same time, the simple sum aggregations in MERF are not flexible enough to compensate for hash collisions in high-resolution NGP features. Applying these feature construction methods directly to the multi-level features in Instant-NGP may lead to a compromise in their representation ability and result in a degradation of rendering quality.

To inherit the rendering efficiency of the deferred NeRF architecture and preserve the strong representation ability of Instant-NGP, we propose NGP-RT, a novel method that utilizes a lightweight attention mechanism to render high-fidelity novel views efficiently. The attention mechanism of NGP-RT employs simple yet effective weighted sum operations with learnable attention parameters that adaptively prioritize the explicit multi-level hash features. Considering

different modalities in the deferred NeRF features, we assign two attention parameters to each resolution level: one for the density value and the other for the color features. Moreover, to mitigate the hash collisions that map different positions to the same entry point of the hash table, we further design the attention parameters to be spatially distinct for hash features extracted from different positions. With these independent learnable parameters, positions mapped to the same entry point can flexibly adjust their relative importance to optimize the multi-view rendering results.

When rendering, we store the pre-computed attention parameters in a low-resolution grid. For each queried point on the casting ray, NGP-RT samples the attention parameters and applies channel-wise weighted sum operations to the multi-level explicit features. The lightweight aggregations reduce over 90% of the multiply-accumulate (MAC) operations compared to the shallow MLP of Instant-NGP, enabling the high-speed rendering process of NGP-RT.

With our design that alleviates the computational burden, global memory access has gradually become a bottleneck in the rendering process. Since the data access of point-wise features is necessary and challenging to shorten, we focus on reducing another source of global memory access: the occupancy check that accesses the multi-level occupancy grids. With access to the occupancy grids saved in global memory, existing methods [26, 31] perform occupancy checks for every marching point to skip empty spaces in the ray marching strategies. However, the typical voxel-by-voxel ray marching method results in a large number of marching points and occupancy checks, congesting the memory bandwidth. To address this issue, we incorporate a pre-computed occupancy distance grid into the ray marching strategy, which informs the distance to the nearest occupied voxel and allows for fewer marching steps. As a result, our method can reduce global memory access without sacrificing visual quality.

We summarize our contributions as follows: (1) We design a lightweight attention mechanism with learnable parameters to efficiently aggregate the multi-level hash features. (2) We incorporate the occupancy distance grid to reduce global memory access of occupancy checks and further accelerate the rendering process. (3) By leveraging the explicit multi-level hash features, we develop a real-time NeRF method that renders high-quality 1080p images at over 100 fps.

2   Related Work

In recent years, the neural radiance field [25] has attracted significant attention for its capability in 3D geometry modeling and high-fidelity novel view synthesis. Previous work has explored different variants of NeRFs [1, 17, 40, 43] to improve the rendering quality and training speed. However, achieving fast rendering speed for large-scale scenes with high quality at high resolution still remains challenging. In this section, we mainly review the existing techniques for rendering acceleration of NeRF-based methods, which can be divided into two main categories with different motivations.

One category of approaches employs rasterization on geometry proxies to alleviate the complexities associated with volumetric rendering, such as point clouds [29], meshes [6, 32, 33, 36, 38, 45] and planes [23]. MobileNeRF [6] represents radiance fields using a set of polygons with neural textures, and employs a traditional rasterization pipeline to achieve interactive frame rates. Wan *et al.* [38] proposes representing scenes using a two-layer duplex mesh with neural features. BakedSDF [45] optimizes a hybrid neural volume-surface scene representation baked into a high-quality triangle mesh for rendering. Although these methods exhibit excellent real-time performance, they produce suboptimal rendering results at object edges and incorrectly reconstructed surfaces, especially for large-scale 360-degree scenes. In contrast to NeRF-based methods based on volumetric neural representations, 3D Gaussians [18] have emerged as a promising new representation for real-time rendering, but require high storage consumption when rendering large-scale scenes with a massive number of 3D Gaussians. Moreover, it can be limited by its need for sparse point initialization and can exhibit spike-shaped artifacts when viewed from specific angles.

The other category of methods preserves the volumetric rendering scheme to maintain superior rendering quality, which can be further divided into several subcategories. **1)** Firstly, there are methods that focus on reducing the number of sampled points per ray with early termination [28] and adaptive sampling [21,27], or decreasing the number of pixels to be rendered with the render-then-upsample strategy [22,41,42]. Most of these techniques are orthogonal to the discussion of this paper and can be applied to our method for further speed improvements. **2)** Another subset of methods aims to propose more lightweight alternatives for the per-point deep MLPs, such as multiple small MLPs [30,42] and the MIMO MLP architecture [16]. Neural light fields [4, 39] encode the light field into a neural network and require only a single network forward pass for the rendering of each pixel. Although these methods improve the rendering speed of NeRF to some extent, the presence of the remaining MLP still prevents them from rendering fast at high resolutions. **3)** The third typical solution involves explicitly representing the radiance fields and completely removing the MLP for rendering. Many works store the explicit color and density values in the form of voxel grids [7, 8, 34], octrees [46], VDB structures [44], and NerfTrees [14]. In contrast to directly caching the colors and densities, FastNeRF [10] and its variants [35,37] compactly cache the factorized radiance fields and obtain the color values with efficient inner products. While these methods have demonstrated real-time performance on small objects, they tend to consume excessive storage and GPU memory for high-quality renderings of larger scenes.

Within this context, deferred neural rendering techniques [12,13,31,47] combine the storage of explicit features with a tiny view-dependent MLP. SNeRG [12] proposes the deferred rendering method, allowing to evaluate the MLP only once for each pixel. However, the voxel representation encounters challenges in efficiently representing scenes at high resolutions. MERF [31] addresses this limitation by integrating high-frequency features with high-resolution 2D feature planes. However, its expressive power is still limited compared to multi-level

hash features [26]. Our goal is to enhance the representation capability of existing deferred NeRFs for large-scale 360-degree scenes by incorporating powerful multi-level hash features.

3   Preliminaries

Our method is built upon the original NeRF method [25], Instant-NGP [26], and the deferred NeRF architecture. We briefly introduce them below.

The original NeRF represents a 3D scene with 3D fields of densities σ and colors $\mathbf{c}$, and models both fields as continuous functions of the 3D point coordinate $\mathbf{x} \in \mathbb{R}^3$ and view direction $\mathbf{d} \in \mathbb{S}^2$. Specifically, NeRF parameterizes the functions using an MLP, which can be formulated as follows:

$$\sigma, \mathbf{c} = \mathrm{MLP}_\Theta(\mathbf{x}, \mathbf{d}). \tag{1}$$

To render the color $\mathbf{C}$ of a ray emitted from point $\mathbf{o}$ in the direction $\mathbf{d}$, NeRF queries the MLP at 3D points sampled along the ray, $\mathbf{x}_i = \mathbf{o} + t_i\mathbf{d}$, and composites the resulting densities $\{\sigma_i\}$ and colors $\{\mathbf{c}_i\}$ according to the approximated volume rendering integral, as discussed by Max [24],

$$\mathbf{C} = \sum_{i=1}^{N} \alpha_i T_i \mathbf{c}_i, \quad T_i = \prod_{j=1}^{i-1}(1 - \alpha_j), \quad \alpha_i = 1 - e^{-\sigma_i \delta_i}, \tag{2}$$

where T_i and α_i denote the transmittance and alpha values of sample i, and $\delta_i = t_{i+1} - t_i$ is the distance between two adjacent samples.

Similarly, Instant-NGP represents the 3D scene with color and density fields but is parameterized by implicit multi-level grid features along with a tiny MLP. This parameterization can be formulated as:

$$\sigma, \mathbf{c} = \mathrm{MLP}_\theta([\mathbf{F}_{1,\mathbf{x}}, \mathbf{F}_{2,\mathbf{x}}, \dots \mathbf{F}_{K,\mathbf{x}}], \mathbf{d}), \tag{3}$$

where MLP_θ is shallower than MLP_Θ in Eq.(1), $\mathbf{F}_{k,\mathbf{x}}$ denotes the level-k feature of the 3D position $\mathbf{x}$ extracted from the corresponding grid or hash table, and $[\cdot]$ denotes the feature concatenation operation. The success of Instant-NGP demonstrates the strong expressing power of multi-level hash features, which motivates our utilization of the NGP-style features in real-time NeRF rendering.

To further reduce the number of MLP evaluations from N to 1 for each ray, SNeRG [12] utilizes a deferred shading model, in which the 3D scene is represented by the 3D fields of densities σ, diffuse colors $\mathbf{c_d}$, and specular feature vector $\mathbf{v_s}$. At the training time, these fields are parameterized by a deep MLP with the 3D coordinate $\mathbf{x}$ as input:

$$\sigma, \mathbf{c_d}, \mathbf{v_s} = \mathrm{MLP}_\Phi(\mathbf{x}). \tag{4}$$

After the training stage, these optimized fields are baked into a coarse grid that directly stores $[\sigma, \mathbf{c_d}, \mathbf{v_s}]$ as explicit deferred NeRF features, such that the

point-wise density σ_i, diffuse color $\mathbf{c_d}^i$, and feature vector $\mathbf{v_s}^i$ can be directly obtained from the feature grid without any MLP execution. After the volumetric accumulation, SNeRG computes another view-dependent color term for each casting ray via a tiny MLP. The entire deferred rendering process of SNeRG can be formulated as

$$\mathbf{C_d} = \sum_{i=1}^{N} \alpha_i T_i \mathbf{c_d}^i, \quad \mathbf{F} = \sum_{i=1}^{N} \alpha_i T_i \mathbf{v_s}^i,$$
$$\mathbf{C} = \mathbf{C_d} + \mathrm{MLP}_\psi(\mathbf{C_d}, \mathbf{F}, \mathbf{d}), \tag{5}$$

where T_i and α_i follow the notations in Eq. (2), and MLP_ψ denotes the tiny MLP for computing view-dependent colors. Such a combination of deferred NeRF and feature baking makes the fast rendering of SNeRG possible.

4   Method

4.1   NGP-RT Overview

Given a set of calibrated multi-view images of a static scene, our goal is to build a NeRF model that can be rendered in real time with high quality based on the superior expressive power of NGP-style multi-level hash features. However, directly integrating NGP-style features into the deferred NeRF architecture significantly reduces the rendering speed, since the indispensable MLP aggregations for each sample point consume a lot of time.

The tiny MLP used for decoding the implicit features into explicit colors and densities plays a crucial role in alleviating the gradient average problem caused by hash collision. Hidden neurons of the tiny MLP can learn an adaptive masking function that identifies regions dominated by low-frequency textures according to the low-level features, and assigns relatively lower importance to high-level features in these regions. Such an implicit masking effect greatly reduces the redundant usage of high-resolution features and alleviates the collisions of 3D positions mapped to the same hash table entry. Therefore, when using explicit hash features without the subsequent MLP, the importance assigned for conflicted 3D positions cannot be flexibly adjusted according to the contents of 3D scenes, which severely limits the rendering performance.

To address the above challenge, we propose NGP-RT, a real-time NeRF method that incorporates a lightweight attention mechanism to assign spatially varying importance to high-level hash features without sacrificing the rendering speed. As illustrated in Figure 2, NGP-RT divides the multi-level features into the coarse-grained and fine-grained parts. Given a 3D sample point $\mathbf{x}_i$, we construct the deferred NeRF feature $\mathbf{f}_i = [\sigma_i, \mathbf{c_d}^i, \mathbf{v_s}^i]$ at this position by summing the coarse-grained feature $\tilde{\mathbf{f}}_i = [\tilde{\sigma}_i, \tilde{\mathbf{c}}_{\mathbf{d}}^i, \tilde{\mathbf{v}}_{\mathbf{s}}^i]$ and the fine-grained feature $\hat{\mathbf{f}}_i = [\hat{\sigma}_i, \hat{\mathbf{c}}_{\mathbf{d}}^i, \hat{\mathbf{v}}_{\mathbf{s}}^i]$. After feature extraction, the resulting per-point features $\{\mathbf{f}_i | i = 1, \ldots, N\}$ are then fed into the deferred NeRF volume rendering process to compute the RGB color values $\mathbf{C}$ using Eq. (5). In the following parts, we omit the superscript i for simplicity.
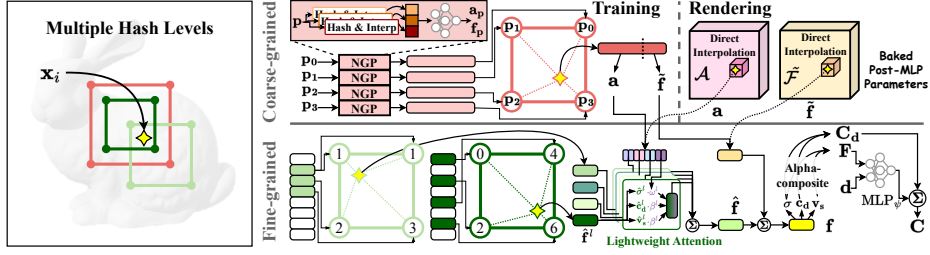
Fig. 2: 2D illustration of the feature construction pipeline of NGP-RT. Following the practice in MERF [31], NGP-RT constructs the deferred NeRF feature $\mathbf{f}$ with a coarse-grained part $\tilde{\mathbf{f}}$ and a fine-grained part $\hat{\mathbf{f}}$. At the training stage, we optimize $\tilde{\mathbf{f}}$ and the attention parameters $\mathbf{a}$ with an auxiliary NGP model. At the inference stage, we bake them into the low-resolution grids $\tilde{\mathcal{F}}$ and $\mathcal{A}$ for fast access and real-time rendering. The fine-grained features $\hat{\mathbf{f}}$ is fused from the high-resolution hash features fused by lightweight attention mechanism. NGP-RT employs the rendering process of deferred NeRF [12] for volume rendering. We omit most of the superscript i for simplicity.

For the coarse-grained level, we follow MERF [31] to optimize $\tilde{\mathbf{f}}$ with an auxiliary NGP model at the training stage. Subsequently, we bake the coarse-grained features into a sparse 3D voxel grid $\tilde{\mathcal{F}}$ with a resolution L_C for fast feature access. As for the fine-grained feature $\hat{\mathbf{f}}$, we model it as the aggregation of multi-level hash features from L high resolutions. We design a lightweight attention mechanism to aggregate the fine-level features with channel-wise weighted sum operations according to the learnable attention parameters. The attention parameters $\mathbf{a}$ vary spatially and are decoded from the low-level branch together with the coarse-grained feature $\tilde{\mathbf{f}}$, which effectively models the dependency of the masking function on coarse-grained contents. We bake the attention parameters into a sparse grid $\mathcal{A}$ of resolution L_C similar to $\tilde{\mathcal{F}}$ for real-time rendering.

Overall, the feature construction of NGP-RT at the rendering stage can be formulated as

$$\mathbf{f} = \tilde{\mathbf{f}} + \hat{\mathbf{f}},$$
$$\tilde{\mathbf{f}} = \mathrm{Interp}(\tilde{\mathcal{F}}, \mathbf{x}), \quad \mathbf{a} = \mathrm{Interp}(\mathcal{A}, \mathbf{x}), \qquad (6)$$
$$\hat{\mathbf{f}} = \mathrm{Att}(\hat{\mathbf{f}}^1, \ldots, \hat{\mathbf{f}}^L; \mathbf{a}),$$

where $\mathrm{Interp}(\mathcal{Y}, \mathbf{x})$ denotes the function that accesses the feature at position $\mathbf{x}$ from feature grid $\mathcal{Y}$ using tri-linear interpolation, and $\mathrm{Att}(\mathbf{f}^1, \ldots, \mathbf{f}^L; \mathbf{a})$ denotes our lightweight attention mechanism that aggregates the multi-level features $\{\mathbf{f}^1, \ldots, \mathbf{f}^L\}$ with the attention parameters $\mathbf{a}$.

At the training stage, we optimize $\mathbf{a}$ and $\tilde{\mathbf{f}}$ with an auxiliary NGP model, which aggregates auxiliary hash features into the attention parameters and coarse-grained features via a shallow MLP. To align with the direct feature interpolation at the rendering stage, we evaluate the auxiliary NGP model at grid corners of resolution L_C, and then conduct tri-linear interpolation to obtain $\mathbf{a}$ and $\tilde{\mathbf{f}}$. After training, we discard the auxiliary NGP model is discarded and directly obtain $\mathbf{a}$ and $\tilde{\mathbf{f}}$ from the sparse voxel grids $\tilde{\mathcal{F}}$ and $\mathcal{A}$, respectively.

4.2   Feature Fusion with Lightweight Attention

Our lightweight attention mechanism utilizes simple yet effective weighted sum operations with learnable parameters to aggregate features from different fine-grained resolution levels into $\hat{\mathbf{f}}$. We first design the attention parameters to be different for all fine-grained levels, such that they can help to balance the overload of different resolution levels and make features from different levels focus on different frequencies. Furthermore, by imitating the implicit masking function inside the tiny MLP of Instant-NGP, we design the attention parameters to be spatially different by formalizing them as part of the decodings from the low-resolution branch. Through the gradients back-propagated from the multi-view rendering loss, these parameters learn the appropriate importance of different spatial positions to alleviate the hash collisions and fully exploit the expressing power of hash features. Considering that the explicit deferred feature $\hat{\mathbf{f}}^l$ consists of density $\hat{\sigma}^l$ and color features $[\hat{\mathbf{c}}_{\mathbf{d}}^l, \hat{\mathbf{v}}_{\mathbf{s}}^l]$, we utilize two attention parameters to model the spatial importance of the two modalities separately. Consequently, the attention parameters at position $\mathbf{x}$ can be decomposed to

$$\mathbf{a} = [\omega^1, \beta^1, \ldots, \omega^L, \beta^L], \tag{7}$$

where ω^l and β^l denote attention parameters for $\hat{\sigma}^l$ and $[\hat{\mathbf{c}}_{\mathbf{d}}^l, \hat{\mathbf{v}}_{\mathbf{s}}^l]$ of the lth fine-grained resolution level, respectively.

Our attention operation multiplies these attention parameters to the corresponding features, formulated by

$$\begin{aligned}
\hat{\mathbf{f}} &= \mathrm{Att}(\hat{\mathbf{f}}^1, \ldots, \hat{\mathbf{f}}^L; \mathbf{a}) \\
&= \left[\sum_{l=1}^{L} \omega^l \cdot \hat{\sigma}^l, \ \ \sum_{l=1}^{L} \beta^l \cdot \hat{\mathbf{c}}_{\mathbf{d}}^l, \ \ \sum_{l=1}^{L} \beta^l \cdot \hat{\mathbf{v}}_{\mathbf{s}}^l \right],
\end{aligned} \tag{8}$$

where $\hat{\mathbf{f}}^l = [\hat{\sigma}^l, \hat{\mathbf{c}}_{\mathbf{d}}^l, \hat{\mathbf{v}}_{\mathbf{s}}^l]$ denotes the fine-grained features at the lth fine-grained resolution level. Our lightweight attention involves only a few MAC operations for each sample point $\mathbf{x}$, and is fast enough to be directly applied to the rendering process. Thus, we do not need to bake the aggregated high-resolution features at specific voxel grids, essentially preserving the details in the fine-level features.

The proposed lightweight attention mechanism effectively resolves explicit hash feature collisions with minimal computational cost, which is previously under-explored. This advance could also help other methods utilizing MLP for feature aggregation, such as the MLP-enhanced 3D Gaussians in 3D AIGC [49] and human reconstruction [19].

4.3   Ray Marching with Occupancy Distance

At the rendering stage, previous methods [26, 31] typically utilize the multi-level occupancy grids $\mathcal{O}$ and query them in the coarse-to-fine order to avoid redundant feature evaluations at positions in the empty space. As depicted in Figure 3, when encountering an empty marching point $\mathbf{x}$, the ray marching process skips
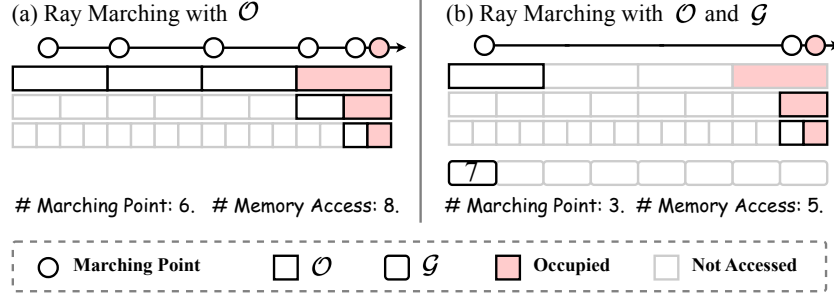
Fig. 3: Comparison between (a) the previous ray marching strategy with only multi-level occupancy grid $\mathcal{O}$ and (b) our strategy with $\mathcal{O}$ and occupancy distance grid $\mathcal{G}$.

that position and advances to the next voxel along the ray direction. The step size is determined based on the voxel size at the corresponding resolution level where the occupancy grids first indicate that $\mathbf{x}$ is unoccupied. Although the coarser-level occupancy grids help to advance with larger step size, the overall ray marching strategy still results in numerous marching points and a significant amount of global memory access to $\mathcal{O}$ at these points. The impact of frequent global memory access becomes more pronounced, particularly for the rendering process of NGP-RT, where dense computation is not the essential bottleneck.

To further accelerate the rendering process by reducing memory access, we incorporate the occupancy distance grid $\mathcal{G}$ into our ray marching strategy. Specifically, for each position $\mathbf{p} = (x, y, z)$, $\mathcal{G}_{\mathbf{p}}$ stores the pre-computed distance from $\mathbf{p}$ to the nearest occupied position of $\mathbf{p}$. To minimize the storage cost, we store the distance as a rounded-down integer in units of the voxel size of $\mathcal{G}$. With the occupancy distance grid, the step size $s_{\mathbf{p}}$ at position $\mathbf{p}$ in our ray marching strategy is determined as follows:

$$
s_{\mathbf{p}} = \begin{cases} v \cdot \mathcal{G}_{\mathbf{p}}, & \text{if } \mathcal{G}_{\mathbf{p}} > 0, \\ s_{\mathcal{O}}, & \text{otherwise,} \end{cases} \tag{9}
$$

where $s_{\mathcal{O}}$ denotes the step size determined by the multi-level occupancy grid, and v denotes the voxel size of the occupancy distance grid. To reduce overall memory access of $\mathcal{G}$, we only access the distance value when the marching point is unoccupied and the resolution of the exiting occupancy level is smaller than the resolution of $\mathcal{G}$. As illustrated in Figure 3, our ray marching strategy reduces the number of marching points and global memory access by skipping the redundant access to occupancy grids. In our experiments, we store $\mathcal{G}$ as a 256^3 grid and save the rounded distance values in the format of `uint8`.

5   Experiments

We evaluate our method on all 9 scenes of the challenging Mip-NeRF 360 dataset [2], including 5 outdoor scenes and 4 indoor scenes. We refer readers to the supplementary material for implementation details.

Table 1: Quantitative evaluation of our method compared to previous work on the Mip-NeRF 360 dataset. The upper part presents the results of *offline* rendering methods, while the lower part presents the performance of *real-time* rendering methods.

Method	Outdoor Scenes			Indoor Scenes			All Scenes			
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	FPS ↑
Instant-NGP [26]	22.80	0.569	0.365	29.15	0.870	0.221	25.62	0.703	0.301	10.4
Plenoxels [8]	21.68	0.513	0.491	24.83	0.766	0.426	23.08	0.625	0.301	6.79
Mip-NeRF 360 [2]	24.47	0.691	0.283	31.72	0.917	0.180	27.69	0.791	0.237	0.06
Zip-NeRF [3]	**25.57**	**0.750**	**0.207**	**32.25**	**0.926**	**0.168**	**28.54**	**0.828**	**0.189**	0.49
Gaussian-7K [18]	23.48	0.654	0.353	28.96	0.907	0.207	25.91	0.766	0.288	107
Mobile-NeRF [6]	21.95	0.470	0.470	-	-	-	-	-	-	-
BakedSDF [45]	22.47	0.585	0.349	27.06	0.836	0.258	24.51	0.697	0.309	>60
MERF [31]	**23.19**	**0.616**	**0.343**	27.80	0.855	0.271	25.24	0.722	0.311	119
NGP-RT (Ours)	22.76	0.614	0.383	**29.25**	**0.891**	**0.195**	**25.64**	**0.737**	**0.299**	108

Baselines. We primarily compare our NGP-RT model to two other methods: Instant-NGP, which utilizes implicit multi-level hash features, and MERF, known for its impressive performance with the deferred NeRF architecture. This allows us to assess the effectiveness of integrating NGP-style features into the deferred NeRF architecture. Additionally, we compare NGP-RT to NeRF-based real-time rendering methods like Mobile-NeRF and BakedSDF. We also include comparisons to offline rendering approaches based on NeRFs like Zip-NeRF, Mip-NeRF 360 and Plenoxels. In contrast to NeRF methods, 3D Gaussian Splatting [18] has gained popularity in novel view rendering. In this study, we further compare NGP-RT to 3D Gaussians at 7k iterations (Gaussian-7K), which exhibits comparable parameter number and model size to NGP-RT. This comparison highlights the improvements our method brings to NeRF-based approaches.

5.1   Comparisons

We evaluate all the above methods regarding the rendering quality and speed at the resolution 1080×1920. We present the quantitative results of PSNR, SSIM, LPIPS and FPS in Table 1 and some qualitative comparisons in Figure 4.

As shown in Table 1, NGP-RT brings improvements to the NeRF-based real-time rendering methods (i.e., those methods above the last divider), and achieves better rendering quality on the challenging Mip-NeRF 360 dataset with over 100 fps at 1080p resolution. Specifically, NGP-RT achieves comparable rendering quality to Instant-NGP with 10× rendering speed, demonstrating the effectiveness and efficiency of our lightweight attention mechanism. Compared to MERF, NGP-RT achieves similar rendering speed and better overall rendering quality,
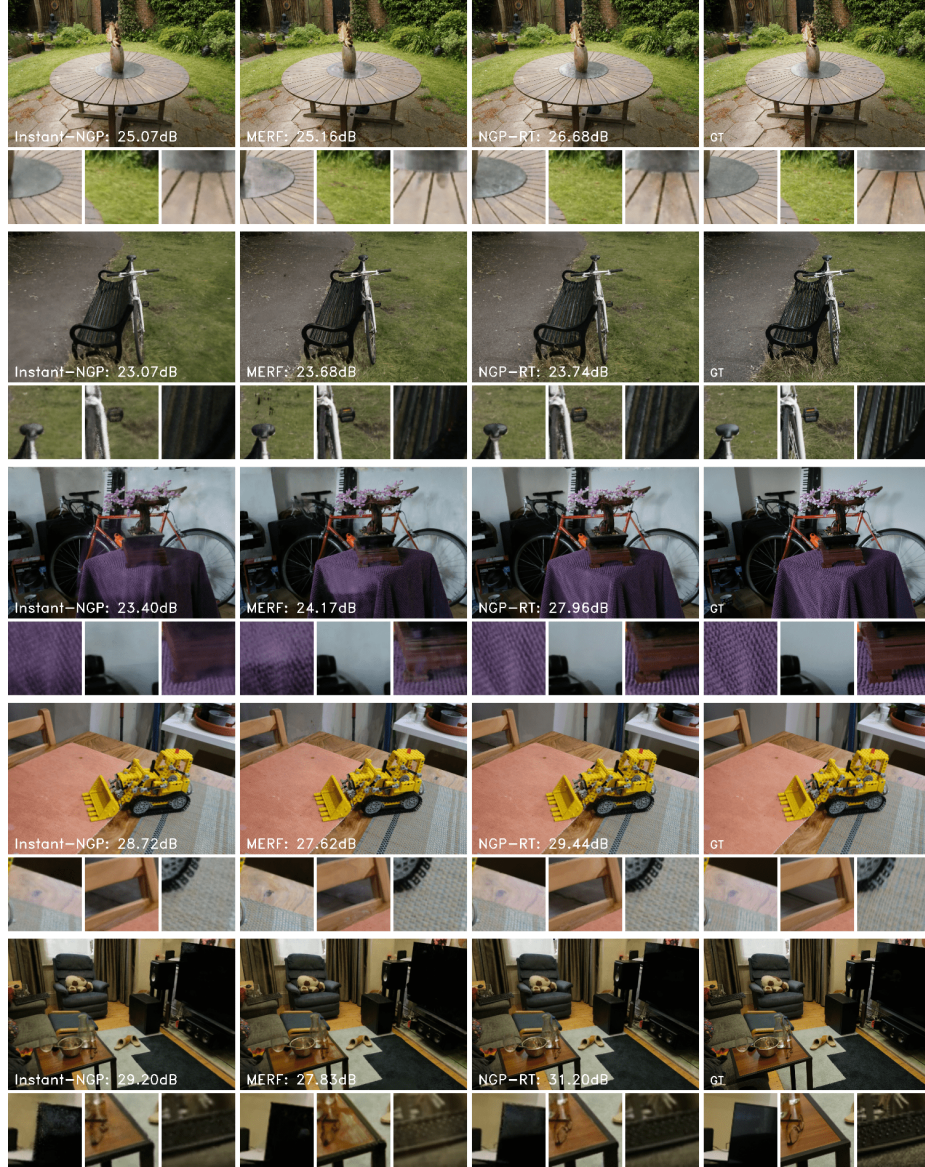
Fig. 4: We show comparisons of NGP-RT to previous methods and the ground truth images from several scenes in the Mip-NeRF 360 dataset. NGP-RT avoids inaccurate floaters and presents better light effects in its renderings.

Table 2: Performance of NGP-RT with varying fine-grained hash feature levels L.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	Time(ms) ↓	FPS ↑
NGP-RT (L=2)	25.64	0.737	0.299	**9.26**	**108**
NGP-RT (L=3)	25.93	0.749	0.283	12.55	79.7
NGP-RT (L=4)	**26.05**	**0.753**	**0.280**	16.17	61.8
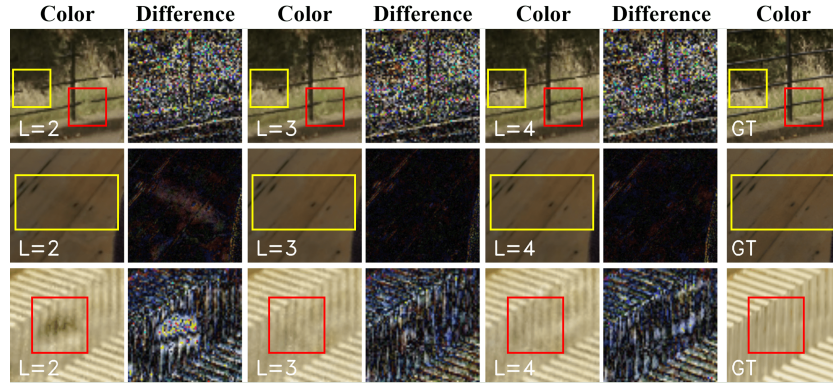
Fig. 5: Rendering results from NGP-RT with different fine-grained hash feature levels.

especially on the indoor scenes. Although both MERF and NGP-RT utilize a tiny MLP to model the view-dependent appearance, the multi-level hash features employed by NGP-RT have stronger expressing power than the tri-plane features used in MERF. As shown in Figure 4, such superior expressing power effectively compensate for the tiny MLP to produce more accurate view-dependent lighting effects, which contributes to the better rendering quality on the indoor scenes. We also notice that NGP-RT performs slightly worse than MERF on the outdoor scenes, potentially due to the high-frequency subtle structures contained in outdoor scenes that our model cannot model well.

Compared to Gaussian-7K [18], NGP-RT achieves comparable rendering speed and quality with similar model size, showcasing the effectiveness of the proposed techniques in enhancing NeRF-based methods. Our advancements allow NeRF-based methods to rival the performance of 3D Gaussian Splatting in real-time novel view rendering at 1080P resolution.

5.2   Ablation Study

Number of Fine-grained Levels. Since NGP-RT employs the multi-level hash features, we first study the influence of the number of fine-grained hash levels (L) that are aggregated by our lightweight attention.

As shown in Table 2 and Figure 5, our method performs better using more fine-grained feature levels due to their stronger expressing capability for detailed structures and special lighting effects. On the other hand, the increasing feature levels compromise the rendering speed of NGP-RT, since the global memory

Table 3: Performance of NGP-RT (L=4) with different feature aggregation methods. We denote the spatially variant/invarant attention parameters with V/Inv.

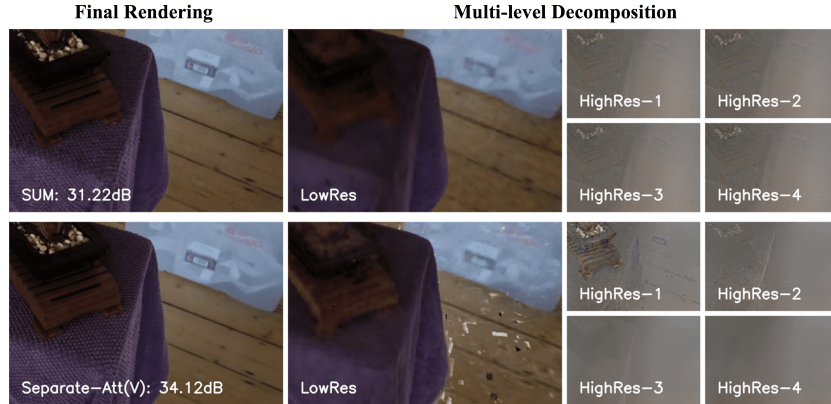| Method	PSNR ↑	SSIM ↑	LPIPS ↓	Time ↓	FPS ↑
MLP	26.22	0.764	0.268	68.02	14.7
SUM	25.51	0.719	0.315	**14.94**	**66.9**
Shared-Att (Inv)	25.69	0.738	0.295	15.18	65.9
Separate-Att (Inv)	25.73	0.740	0.291	15.13	66.1
Shared-Att (V)	25.89	0.745	0.289	16.04	62.3
Separate-Att (V)	**26.05**	**0.753**	**0.280**	16.17	61.8

Fig. 6: Visualization of contributions of different hash levels to the final renderings. We compare the decomposition of renderings from the SUM and Separate-Att(V) aggregation methods. The gray colors in decomposition results represent zero values.

access almost dominates the time consumption of NGP-RT. In particular, the memory access pattern of NGP-RT is relatively stochastic and leads to limited utilization efficiency of the memory cache, thus greatly influencing the overall rendering speed. We also observe that NGP-RT ($L = 4$) performs slightly better than NGP-RT ($L = 3$), indicating that additional hash levels bring limited quality improvements to NGP-RT.

Feature Aggregation Methods. Based on NGP-RT (L=4), we ablate different aggregation methods for the multi-level hash features, including simple sum operation (SUM), tiny MLP (MLP) utilized in Instant-NGP and two modes of lightweight attention (Shared-Att and Separate-Att) with spatially variant/invariant (V/Inv) learnable parameters. We denote the attention method with a single shared attention parameter for each resolution level with Shared-Att and the attention method with two separate parameters for density and color features of each level with Separate-Att. Results summarized in Table 3 demonstrate that Separate-Att (V) achieves comparable rendering quality to the MLP aggregation with more than $4\times$ FPS. Compared to the SUM aggregation utilized in MERF and other alternatives of lightweight attention mechanisms, our design achieves the best rendering quality with comparable rendering speed.

Table 4: NGP-RT rendering speed with or without the occupancy distance grid $\mathcal{G}$.

L	$\mathcal{G}$	PSNR ↑	# Marching ↓	# Occupied ↓	Time (ms) ↓
2	w/o	**25.67**	85.1	17.3	9.98
	w/	25.64	**46.7**	17.3	**9.26**
3	w/o	25.91	86.4	17.6	13.89
	w/	**25.93**	**45.3**	17.6	**12.55**
4	w/o	**26.06**	85.9	18.1	18.22
	w/	26.05	**46.5**	18.1	**16.17**

We also visualize the contributions of different fine-grained hash levels to the final renderings in Figure 6. When visualizing the contribution of a particular feature level, we set the color-related features of all other hash levels to zero values, including the coarse-level features, and render the results with the original multi-level fused densities. As shown in Figure 6, our lightweight attention mode efficiently splits the detailed textures into different fine-grained hash levels. The hash collision is greatly alleviated with attention parameters focusing on selected regions instead of the whole scene space. When multiple 3D positions are mapped to the same entry point of a hash table, a large portion is associated with small attention values, thus avoiding the interference of backward gradient values.

Ray Marching with Occupancy Distance. We conduct experiments on NGP-RT with different fine-grained hash levels to validate the acceleration brought by ray marching with the help of occupancy distance grid $\mathcal{G}$. As demonstrated by results presented in Table 4, the incorporation of $\mathcal{G}$ results in 7%-10% acceleration on the rendering speed, which is important for real-time rendering applications. Additionally, Table 4 provides the average number of marching points (# Marching) and the average number of occupied points (# Occupied) on each ray. Our approach successfully reduces over 40% of redundant marching points while maintaining the number of valid occupied sample points.

6   Conclusion

In this paper, we present NGP-RT, a novel method that enables real-time rendering of high-quality novel views based on multi-level hash features. By leveraging a lightweight attention mechanism in feature aggregation, NGP-RT preserves the strong representation ability of explicit multi-level hash features while significantly reducing the computational cost compared to the shallow MLP of Instant-NGP. Moreover, we introduce a ray marching strategy with the occupancy distance grid to minimize global memory access during occupancy checks, leading to further improvements in rendering speed. Experimental results demonstrate that NGP-RT achieves high-quality 1080p rendering at over 100 fps, making it suitable for interactive virtual reality experiences and immersive applications. Future research includes extending NGP-RT to handle dynamic scenes and achieve further performance improvements.

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: 2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021. pp. 5835–5844. IEEE (2021). https://doi.org/10.1109/ICCV48922.2021.00580, https://doi.org/10.1109/ICCV48922.2021.00580

2. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022. pp. 5460–5469. IEEE (2022). https://doi.org/10.1109/CVPR52688.2022.00539, https://doi.org/10.1109/CVPR52688.2022.00539

3. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Zip-nerf: Anti-aliased grid-based neural radiance fields. ICCV (2023)

4. Cao, J., Wang, H., Chemerys, P., Shakhrai, V., Hu, J., Fu, Y., Makoviichuk, D., Tulyakov, S., Ren, J.: Real-time neural light field on mobile devices. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8328–8337 (2023)

5. Chen, A., Xu, Z., Geiger, A., Yu, J., Su, H.: Tensorf: Tensorial radiance fields. In: Avidan, S., Brostow, G.J., Cissé, M., Farinella, G.M., Hassner, T. (eds.) Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXXII. Lecture Notes in Computer Science, vol. 13692, pp. 333–350. Springer (2022). https://doi.org/10.1007/978-3-031-19824-3_20, https://doi.org/10.1007/978-3-031-19824-3_20

6. Chen, Z., Funkhouser, T.A., Hedman, P., Tagliasacchi, A.: Mobilenerf: Exploiting the polygon rasterization pipeline for efficient neural field rendering on mobile architectures. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023. pp. 16569–16578. IEEE (2023). https://doi.org/10.1109/CVPR52729.2023.01590, https://doi.org/10.1109/CVPR52729.2023.01590

7. Deng, C.L., Tartaglione, E.: Compressing explicit voxel grid representations: fast nerfs become also small. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1236–1245 (2023)

8. Fridovich-Keil, S., Yu, A., Tancik, M., Chen, Q., Recht, B., Kanazawa, A.: Plenoxels: Radiance fields without neural networks. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022. pp. 5491–5500. IEEE (2022). https://doi.org/10.1109/CVPR52688.2022.00542, https://doi.org/10.1109/CVPR52688.2022.00542

9. Gao, Q., Xu, Q., Su, H., Neumann, U., Xu, Z.: Strivec: Sparse tri-vector radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 17569–17579 (October 2023)

10. 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)

11. He, Y., Wang, P., Hu, Y., Zhao, W., Yi, R., Liu, Y.J., Wang, W.: Mmpi: a flexible radiance field representation by multiple multi-plane images blending. arXiv preprint arXiv:2310.00249 (2023)

12. Hedman, P., Srinivasan, P.P., Mildenhall, B., Barron, J.T., Debevec, P.: Baking neural radiance fields for real-time view synthesis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5875–5884 (2021)

13. Hu, D., Zhang, Z., Hou, T., Liu, T., Fu, H., Gong, M.: Multiscale representation for real-time anti-aliasing neural rendering. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 17772–17783 (October 2023)
14. Hu, T., Liu, S., Chen, Y., Shen, T., Jia, J.: Efficientnerf efficient neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12902–12911 (2022)
15. 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 (ICCV). pp. 19774–19783 (October 2023)
16. Kaneko, T.: Mimo-nerf: Fast neural rendering with multi-input multi-output neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3273–3283 (2023)
17. Karnewar, A., Ritschel, T., Wang, O., Mitra, N.: Relu fields: The little non-linearity that could. In: ACM SIGGRAPH 2022 Conference Proceedings. pp. 1–9 (2022)
18. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Trans. Graph. **42**(4), 139:1–139:14 (2023). https://doi.org/10.1145/3592433, https://doi.org/10.1145/3592433
19. Kocabas, M., Chang, J.H.R., Gabriel, J., Tuzel, O., Ranjan, A.: Hugs: Human gaussian splats. arXiv preprint arXiv:2311.17910 (2023)
20. Kohler, J., Sanchez, N.G., Cavalli, L., Herold, C., Pumarola, A., Garcia, A.G., Thabet, A.: fmpi: Fast novel view synthesis in the wild with layered scene representations. arXiv preprint arXiv:2312.16109 (2023)
21. Kurz, A., Neff, T., Lv, Z., Zollhöfer, M., Steinberger, M.: Adanerf: Adaptive sampling for real-time rendering of neural radiance fields. In: European Conference on Computer Vision. pp. 254–270. Springer (2022)
22. Li, S., Li, H., Wang, Y., Liao, Y., Yu, L.: Steernerf: Accelerating nerf rendering via smooth viewpoint trajectory. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20701–20711 (2023)
23. Lin, Z.H., Ma, W.C., Hsu, H.Y., Wang, Y.C.F., Wang, S.: Neurmips: Neural mixture of planar experts for view synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15702–15712 (2022)
24. Max, N.: Optical models for direct volume rendering. IEEE Transactions on Visualization and Computer Graphics **1**(2), 99–108 (1995)
25. 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)
26. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM Transactions on Graphics (ToG) **41**(4), 1–15 (2022)
27. Neff, T., Stadlbauer, P., Parger, M., Kurz, A., Mueller, J.H., Chaitanya, C.R.A., Kaplanyan, A., Steinberger, M.: Donerf: Towards real-time rendering of compact neural radiance fields using depth oracle networks. In: Computer Graphics Forum. vol. 40, pp. 45–59. Wiley Online Library (2021)
28. Piala, M., Clark, R.: Terminerf: Ray termination prediction for efficient neural rendering. In: 2021 International Conference on 3D Vision (3DV). pp. 1106–1114. IEEE (2021)
29. Rakhimov, R., Ardelean, A.T., Lempitsky, V., Burnaev, E.: Npbg++: Accelerating neural point-based graphics. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15969–15979 (2022)

30. 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)
31. Reiser, C., Szeliski, R., Verbin, D., Srinivasan, P., Mildenhall, B., Geiger, A., Barron, J., Hedman, P.: Merf: Memory-efficient radiance fields for real-time view synthesis in unbounded scenes. ACM Transactions on Graphics (TOG) **42**(4), 1–12 (2023)
32. Riegler, G., Koltun, V.: Stable view synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12216–12225 (2021)
33. Rojas, S., Zarzar, J., Pérez, J.C., Sanakoyeu, A., Thabet, A., Pumarola, A., Ghanem, B.: Re-rend: Real-time rendering of nerfs across devices. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3632–3641 (2023)
34. Sun, C., Sun, M., Chen, H.: Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022. pp. 5449–5459. IEEE (2022). `https://doi.org/10.1109/CVPR52688.2022.00538`, `https://doi.org/10.1109/CVPR52688.2022.00538`
35. Tang, J., Chen, X., Wang, J., Zeng, G.: Compressible-composable nerf via rank-residual decomposition. Advances in Neural Information Processing Systems **35**, 14798–14809 (2022)
36. Tang, J., Zhou, H., Chen, X., Hu, T., Ding, E., Wang, J., Zeng, G.: Delicate textured mesh recovery from nerf via adaptive surface refinement. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 17739–17749 (October 2023)
37. Wadhwani, K., Kojima, T.: Squeezenerf: Further factorized fastnerf for memory-efficient inference. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2717–2725 (2022)
38. Wan, Z., Richardt, C., Božič, A., Li, C., Rengarajan, V., Nam, S., Xiang, X., Li, T., Zhu, B., Ranjan, R., et al.: Learning neural duplex radiance fields for real-time view synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8307–8316 (2023)
39. Wang, H., Ren, J., Huang, Z., Olszewski, K., Chai, M., Fu, Y., Tulyakov, S.: R2l: Distilling neural radiance field to neural light field for efficient novel view synthesis. In: European Conference on Computer Vision. pp. 612–629. Springer (2022)
40. Wang, P., Liu, Y., Chen, Z., Liu, L., Liu, Z., Komura, T., Theobalt, C., Wang, W.: F2-nerf: Fast neural radiance field training with free camera trajectories. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4150–4159 (2023)
41. Wang, Z., Li, L., Shen, Z., Shen, L., Bo, L.: 4k-nerf: High fidelity neural radiance fields at ultra high resolutions. arXiv preprint arXiv:2212.04701 (2022)
42. Wu, X., Xu, J., Zhu, Z., Bao, H., Huang, Q., Tompkin, J., Xu, W.: Scalable neural indoor scene rendering. ACM transactions on graphics **41**(4) (2022)
43. Xie, X., Gherardi, R., Pan, Z., Huang, S.: Hollownerf: Pruning hashgrid-based nerfs with trainable collision mitigation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3480–3490 (2023)
44. Yan, H., Liu, C., Ma, C., Mei, X.: Plen-vdb: Memory efficient vdb-based radiance fields for fast training and rendering. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023. pp. 88–96. IEEE (2023). `https://doi.org/10.1109/CVPR52729.2023.00017`, `https://doi.org/10.1109/CVPR52729.2023.00017`

45. Yariv, L., Hedman, P., Reiser, C., Verbin, D., Srinivasan, P.P., Szeliski, R., Barron, J.T., Mildenhall, B.: Bakedsdf: Meshing neural sdfs for real-time view synthesis. In: Brunvand, E., Sheffer, A., Wimmer, M. (eds.) ACM SIGGRAPH 2023 Conference Proceedings, SIGGRAPH 2023, Los Angeles, CA, USA, August 6-10, 2023. pp. 46:1–46:9. ACM (2023). `https://doi.org/10.1145/3588432.3591536`, `https://doi.org/10.1145/3588432.3591536`

46. 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)

47. Zhang, J., Huang, J., Cai, B., Fu, H., Gong, M., Wang, C., Wang, J., Luo, H., Jia, R., Zhao, B., et al.: Digging into radiance grid for real-time view synthesis with detail preservation. In: European Conference on Computer Vision. pp. 724–740. Springer (2022)

48. Zhang, K., Riegler, G., Snavely, N., Koltun, V.: Nerf++: Analyzing and improving neural radiance fields. CoRR **abs/2010.07492** (2020), `https://arxiv.org/abs/2010.07492`

49. Zou, Z.X., Yu, Z., Guo, Y.C., Li, Y., Liang, D., Cao, Y.P., Zhang, S.H.: Triplane meets gaussian splatting: Fast and generalizable single-view 3d reconstruction with transformers. arXiv preprint arXiv:2312.09147 (2023)