Two is Better than One: Achieving High-Quality 3D Scene Modeling with a NeRF Ensemble

Francesco Di Sario, Riccardo Renzulli, Enzo Tartaglione, Marco Grangetto

Two is Better than One: Achieving High-Quality 3D Scene Modeling with a NeRF Ensemble

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. Neural Radiance Field (NeRF) is a popular method for synthesizing novel views of a scene from a set of input images. While NeRF has demonstrated state-of-the-art performance in several applications, it suffers from high computational requirements. Recent works have attempted to address these issues by including explicit volumetric information, which makes the optimization process difficult when fine-graining the voxel grids. In this paper, we propose an ensemble approach that combines the strengths of two NeRF models to achieve superior results compared to state-of-the-art architectures, with a similar number of parameters. Experimental results show that our ensemble approach is a promising strategy for performance enhancement, and beats vanilla approaches under the same parameter's cardinality constraint.

Keywords: NeRF · Ensemble · 3D scene modeling · Compression

1 Introduction

Neural Radiance Fields (NeRFs) [16] have recently shown impressive results in synthesizing photo-realistic 3D scenes from a set of 2D images. However, NeRF suffers from limited scene diversity, long training time, and sensitivity to training data [14]. To address these issues, recent works have proposed several improvements to the original NeRF framework, such as NeRF++, which extends NeRF to unbounded scenes [26].

Another promising approach to improve NeRF, yet to be explored, is ensembling. Ensemble methods combine multiple models to achieve better performance than a single model. In the context of NeRF, ensembling can be achieved by training multiple NeRF models on different subsets of the training data, or by training different models with different architectures or hyperparameters. Ensembling has been shown to be effective in improving the performance of various computer vision tasks, such as image classification [8] and object detection [12]. One of the key advantages of ensemble methods is their ability to combine the predictions of multiple models to produce a more accurate and robust prediction.

This is particularly useful when the individual models have different strengths and weaknesses, as the ensemble can leverage the strengths of each model while mitigating their weaknesses.

However, there are several challenges associated with ensemble methods that can limit their effectiveness. For example, how can we select the appropriate combination of models in the ensemble? This is particularly difficult when there are a large number of potential models to choose from, or when the individual models are highly correlated with each other. Another challenge is how to effectively combine the predictions of the individual models, particularly when they have different levels of accuracy or confidence.

In this work, we explore the potentiality of NeRF ensembling to improve performances. More specifically, we adopt a baseline, state-of-the-art architecture, DVGO [22], trained on a very well-known dataset, Synthetic-NeRF [16]. We observe that, by employing a vanilla ensembling strategy of two models, we may obtain suboptimal results. We propose a simple yet effective solution to counter it, observing consistent performance improvement, with respect to the baseline models, on a broad variety of tested resolutions, *under the same memory footprint constraints*. This paper aims at moving the first steps towards the definition of an ultimate, highly-performing, efficient NeRF ensembling strategy. At a glance, the contributions of this work are the following:

- To the best of our knowledge, this is the first work proposing a joint ensembling and compression scheme for NeRF models: a formulation to prevent performance degradation in case of conjoint pruning is proposed.
- We observe, on known benchmarks, that ensembling multiple models at different scales requires fewer parameters than training and compressing one large model directly (under the same generated image quality constraint).

2 Related works

Neural Radiance Field (NeRF) [16] stand out in recent years as the most prevalent method for novel view rendering that infers photo-realistic views given a moderate number of input images. Unlike traditional explicit volumetric representation techniques, NeRF encodes the entire content of the scene including view-dependent color emission and density into a single multi-layer perceptron (MLP) [16]. Besides, NeRF-based approaches are proving on the field to have good generalization when undergoing several transformations, like changing environmental light [1,21], image deformation [6,18,24] and are even usable in more challenging setups including meta learning [23], learn dynamically-changing scenes [7,11,15,25] and even in generative contexts [2,9,20]. Compared to explicit representations, NeRF requires very little storage space, but on the contrary suffers from lengthy training time and very slow rendering speed, as the MLP is queried an extremely high number of times for rendering a single image. In more detail, a NeRF takes as input a 3D point in space x and a viewing direction d and returns a color c and a density σ. It utilizes volume rendering techniques to achieve advanced 3D reconstruction: given a camera and a sparse set of cali-

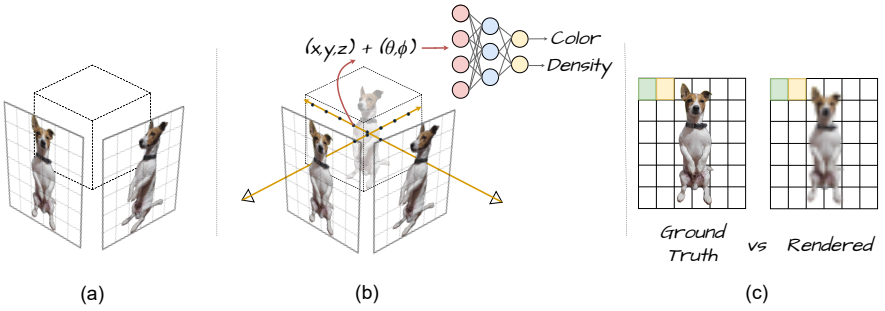


Fig. 1: An overview of NeRF training is presented. (a) illustrates the initial training setup, made of a sparse set of calibrated images of the same object under different viewpoints conditions. (b) shows the training process made of ray casting, ray sampling, and volume rendering to compute the pixel color. Then the generated image is compared to the ground truth (c).

brated images capturing the scene from various viewpoints, the rendering process involves casting a ray from the camera's eye to the center of a pixel, sampling $x_1, ..., x_k$ points along the ray and evaluating those points, obtaining a color, c, and a density value, σ. The final pixel color, $\hat{c}$, is determined by alpha-blending all the computed color values $(c_1, \cdots, c_k)$ along the ray

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

where α_i is the value used for blending the colors values (its calculation depends on the distance between adjacent sampling points $\delta = ||x_{j+1} - x_j||$ and T_i is the transmittance. This process is repeated for each pixel in the image. Once the final image is generated, it is compared to the ground truth using the photometric loss (2), and the parameters of the multilayer perceptron are optimized through backpropagation. Fig. 1 summarizes this process.

To reduce inference and training time, explicit prior on the 3D object representation can be imposed. The most intuitive yet effective approach relies on splitting the 3D volume into small blocks, each of which is learned by a tiny NeRF model. With KiloNeRF [19], the advantage of doing this is twofold. Firstly, the size of a single NeRF model is much smaller than the original one, reducing the latency time. Secondly, the rendering process itself becomes parallelizable, as multiple pixels can be rendered simultaneously. The downside of this approach is that the granularity of the KiloNeRFs needs to be properly tuned, and the proper tuning process can be time-consuming and require significant computational resources. Additionally, KiloNeRFs may struggle to capture fine details and high-frequency variations in the input data, which can lead to inaccurate reconstructions. To address these limitations, researchers have proposed several extensions and variations of the original NeRF and KiloNeRF models. For

Algorithm 1 NeRF ensemble training algorithm.

Require: Training set $\mathcal{D}_{\text{train}}$, validation set $\mathcal{D}_{\text{val}}$, ensemble of 2 NeRFs
1: **Stage 1:** Train 2 NeRFs independently on $\mathcal{D}_{\text{train}}$
2: **while** Performance does not drop **do**
3: **Stage 2 (Ens-FT):** Fine-tune the ensemble of NeRFs
4: **Stage 3 (CPE):** Compress the ensemble, using $\mathcal{D}_{\text{val}}$
5: **end while**
6: **return** Ensemble of NeRFs

instance, some works have explored the use of hierarchical or multi-scale representations to better capture details at different levels of the scene [28,27]. Others have investigated the incorporation of additional priors or constraints, such as symmetry or smoothness assumptions, to improve the robustness and generalization of the models [13,16]. In parallel, the development of NeRFs with direct voxel grid optimization is gaining more and more success. Direct Voxel Grid Optimization (DVGO) [22] is a popular baseline for NeRFs due to its simplicity and effectiveness. In contrast to traditional NeRFs, which use a continuous representation of the scene, DVGO operates directly on a voxel grid. This makes DVGO much more computationally efficient than NeRFs, as it allows for parallelization of the ray-marching process and significantly reduces the number of samples required to render an image. Additionally, DVGO is less prone to overfitting and can handle more complex scenes with higher levels of detail. Despite its limitations in terms of scalability, DVGO provides a strong and reliable baseline for evaluating the performance of more advanced methods such as NeRFs. Moreover, it has been shown that by training a NeRF with initialization from a DVGO model, the NeRF can achieve comparable performance while requiring significantly less training time and computational resources. Therefore, DVGO remains a useful and widely used baseline for testing and comparing novel techniques for 3D scene representation and rendering.

For this reason, in the present study, we have opted to employ the DVGO architecture as a reference NeRF framework. While hybrid models, also known as Explicit Voxel Grid models, exhibit superior efficiency and increased accuracy, they necessitate substantial storage capacity, typically on the order of gigabytes. As a result, some voxel pruning techniques have recently emerged with the goal of minimizing storage requirements for these models. Re:NeRF [4] represents a state-of-the-art approach for compressing EVG NeRFs and stands among the forefront methods in reducing storage demands.

3 Method

In this section, we present our framework, which allows for the combination of two NeRF models of different grid resolutions to improve performance on novel view synthesis tasks. A general overview of the employed strategy is presented in Algorithm 1. Our framework consists of three stages:

– independent training of multiple NeRF models (Stage 1);
– ensemble construction and fine-tuning (Stage 2);
– conjoint pruning phase of the ensembled NeRFs (Stage 3).

Independent Training of NeRF Models (Stage 1). In the first stage, we train multiple NeRF models independently, each with a different grid resolution. For each of them, we train the model for a fixed number of iterations, according to the standard learning policy defined, minimizing the NeRF loss function:

$$\mathcal{L}(\boldsymbol{w}) = \frac{1}{M} \sum_{m=1}^{M} \|C_m - C(\boldsymbol{w}, x_m, \omega_m)\|_2^2, \tag{2}$$

where $\boldsymbol{w}$ represents the model parameters, C_m is the ground truth color of the m-th pixel, $C(\boldsymbol{w}, x_m, \omega_m)$ is the predicted color by the NeRF for the same pixel, x_m is the 3D point where the pixel lies, ω_m is the corresponding viewing direction and M is the total number of training samples. After the whole training process ends, as observed in some recent work in the literature [4], the size of the generated models can be drastically reduced, with marginal or even no impact on the performance. This optional stage employs an iterative pruning strategy, followed by quantization and entropy coding, on the models at isolation: we will name it independent pruning of models (IPM).

Ensemble construction and ensemble fine-tuning (Stage 2). Once multiple NeRF models have been trained or compressed, we construct an ensemble by combining them. Specifically, we simply perform an interpolation for the outputs

$$C_m^{\text{avg}} = \frac{1}{2} \sum_{n=1}^{2} C^n(\boldsymbol{w}^n, x_m, \omega_m), \tag{3}$$

where C^2 indicates the output of the ensemble of 2 NeRFs, C^n is the output of the n-th NeRF, and $\boldsymbol{w}^n$ are the parameters of the n-th NeRF.

After constructing the ensemble, a fine-tuning stage follows. Specifically, we observe that by optimizing the output provided by (3), we have

$$\mathcal{L}^{\text{avg}}(\boldsymbol{w}) = \frac{1}{M} \sum_{m=1}^{M} \left\| C_m - \frac{1}{2} \sum_{n=1}^{2} C^n(\boldsymbol{w}^n, x_i, \omega_i) \right\|_2^2. \tag{4}$$

Conjoint pruning of the ensemble (Stage 3). The final phase consists in jointly pruning the ensemble. We refer to this phase as conjoint pruning of the ensemble (CPE). This phase targets superior performance compared to ensembling pre-trained models, with comparable (sometimes even lower) memory footprint. The optimization of the loss function needs to be carefully reconsidered as it may lead to suboptimal results as the pruning process progresses. Indeed, according to (4), the ensemble's output is computed by averaging the output of the ensemble models. In the scenario where the models are pre-trained, it is generally not a concern, and optimizing them based on this loss function

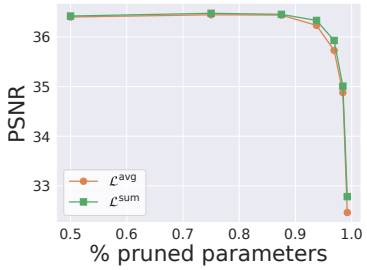


Fig. 2: Comparison between $\mathcal{L}^{\text{sum}}$ and $\mathcal{L}^{\text{avg}}$ losses on Lego Dataset. Each point is an average of the six ensemble configurations tested. It is important to note that as the pruning progresses, the sum approach tends to yield better performance.

can yield great results. Nevertheless, in the case of conjoint pruning, optimizing $\mathcal{L}^{\text{avg}}$ could be quite problematic. In particular, due to the pruning process, C^1 and/or C^2 (the output of the models of the ensemble) may contain multiple null values (or values close to zero), depending on the compression rate. Therefore, it is evident that an averaging of the ensemble models in such a scenario would result in a significant reduction of the signal output of both models, regardless of the pruning phase. We can avoid this problem by simply adopting, as the output of our ensemble,

$$C_i^2 = \sum_{n=1}^{2} C^n(\boldsymbol{w}^n, x_i, \omega_i), \tag{5}$$

which leads to the minimization of the following loss function

$$\mathcal{L}^{\text{sum}}(\boldsymbol{w}) = \frac{1}{M} \sum_{m=1}^{M} \left\| C_m - \sum_{n=1}^{2} C^n(\boldsymbol{w}^n, x_i, \omega_i) \right\|_2^2. \tag{6}$$

In Fig. 2, it can be observed that optimizing $\mathcal{L}^{\text{sum}}$ achieves better performances than optimizing $\mathcal{L}^{\text{avg}}$.

4 Experiments

In this section, we present the empirical results obtained on the Synthetic-NeRF [16] dataset. It contains eight different realistic objects created with Blender(*chair, drums, ficus, hotdog, lego, materials, mic* and *ship*), which are synthesized from NeRF.

4.1 Setup

The target image resolution has been set up to 800×800 pixels, having 100 views for training, 100 for validation, and 200 for testing. We choose DVGO [22] as a

Table 1: Results on Synthetic-NeRF for DVGO. All the presented results are averaged on the eight different datasets in Synthetic-NeRF. For the ensemble, the models have resolution 160^3 and the one indicated. In bold we report the best values, while in italic the second best ones.

Metric	Method			Compress	Resolution					
	IPM	ENS-FT	CPE		160^3	170^3	180^3	190^3	200^3	256^3
PSNR(↑)					31.813	31.939	32.063	32.158	32.270	32.751
		✓			**32.821**	**33.061**	**33.113**	**33.183**	**33.265**	**33.509**
	✓			LOW	31.801	31.909	32.240	32.313	32.421	32.644
				HIGH	31.397	31.545	31.875	31.963	32.071	32.299
	✓	✓		LOW	32.431	32.846	32.913	32.948	33.042	33.200
				HIGH	31.858	32.229	32.329	32.399	32.491	32.768
		✓	✓	LOW	*32.724*	*32.938*	*32.942*	*33.002*	*33.084*	*33.302*
				HIGH	32.110	32.430	32.424	32.523	32.617	32.899
SIZE(MB)(↓)					634.44	766.78	907.23	1074.01	1248.80	2619.70
		✓			1274.62	1401.79	1545.86	1706.12	1882.07	3334.06
	✓			LOW	4.67	5.35	6.13	6.99	8.80	13.55
				HIGH	**2.49**	**2.85**	**3.25**	**3.65**	**4.54**	**6.97**
	✓	✓		LOW	7.22	8.15	8.78	9.44	10.11	14.98
				HIGH	4.28	4.20	4.52	4.85	5.19	7.74
		✓	✓	LOW	7.37	7.84	8.46	9.08	9.77	14.37
				HIGH	*3.93*	*4.19*	*4.50*	*4.83*	*5.17*	*7.57*

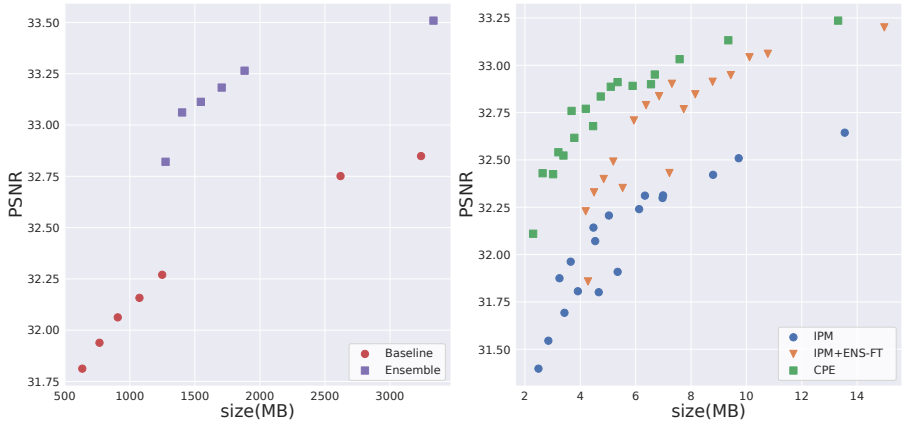
reference architecture, and we adopt the original paper's learning strategy and hyperparameters configuration. We propose standard image generation quality metrics like PSNR, SSIM, and LPIPS (computed on AlexNet).

Besides, we compare the various results in terms of the size (in MB) of the model compressed by Re:NeRF. We conduct experiments at different voxel grid resolutions: 160^3, 170^3, 180^3, 190^3, 200^3, and 256^3. According to the proposed approach, first, we train (and compress) several models at different resolutions; then, we construct all the possible ensembles of two models combining the lowest resolution (160^3) with all the available resolutions (from 160^3 up to 256^3). As a standard pruning, quantization, and compression approach, we adopt Re:NeRF [4]. Our code is developed using PyTorch 1.12, and the experiments are performed on an NVIDIA A40 GPU.[1]

4.2 Results

Table 1 reports the results achieved on Synthetic-NeRF. The table consists of four macro-sections, each corresponding to a reference measure (namely PSNR, SSIM, LPIPS, and SIZE). Analyzing each macro section, we observe the following: in the first row, the performance of the baseline models; in the second row (ENS-FT), the performance of the fine-tuned ensemble; in the third row (IPM),

[1] https://github.com/EIDOSLAB/nerf-ensemble-two-is-better-than-one.



(a) (b)

Fig. 3: (a) Comparison between baseline and ensembling in terms of PSNR and memory footprint. (b) Comparison among individually pruned models (IPM), IPM followed by one fine-tuning stage in ensembling (IPM+ENS-FT), and conjoint pruned ensemble (CPE).

the performance of individually compressed models at two rates, low and high (corresponding to 87.50% and 96.87% of the total parameters, respectively); in the fourth row (IPM + ENS-FT), the performance of the pre-compressed models in the ensemble, and finally, the last row of each macro section shows the performance of the conjoint pruning of the ensemble (CPE).

Please consider that every entry of the table is an average of eight models, trained on the eight datasets collected within Synthetic-NeRF. Consistently, we observe that, under the same resolution constraint, the proposed ensemble approach performs the best. More specifically, we observe a minor degradation of the performance as the compression regime increases (as also indicated in [4]). However, when investigating the model size, we observe that compressing the ensemble can sensibly reduce its size, making it drop from order GB to a few MB. In order to have a more visual impact on the benefits provided by our proposed ensembling approach, we propose, in Fig. 3a, a comparison between baseline models and the proposed ensemble, in terms of the model's size. We observe that, under the same model memory footprint, even without compression, the ensemble consistently outperforms the baseline. We also propose a comparison among single pruned models, ensemble with pre-compressed models and conjoint pruned ensemble in Fig 3b. Also in this case the ensembling shows a consistent performance improvement, despite consuming a comparable amount of memory. In Fig. 4 a qualitative comparison between the analyzed configurations is proposed.

Fig. 5 presents a comparison between our ensembling strategy and several state-of-the-art hybrid methods, such as Plenoxels [5], NSVF [10], Instant-

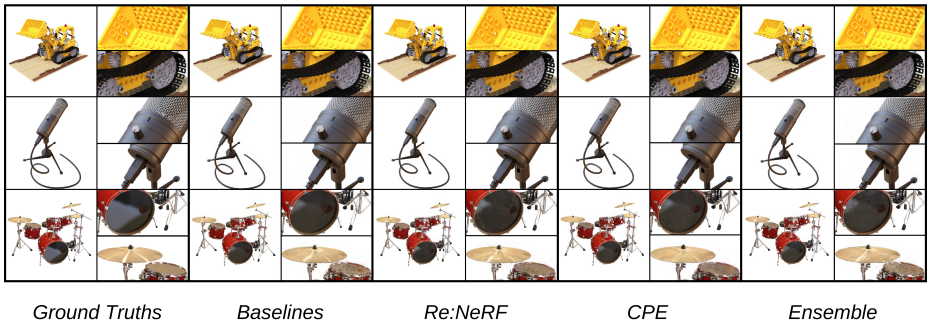


Fig. 4: Qualitative results. As a baseline, we adopted a grid of 256^3 voxels, while for the ensemble, two grids with dimensions of 160^3 and 256^3 voxels were used.

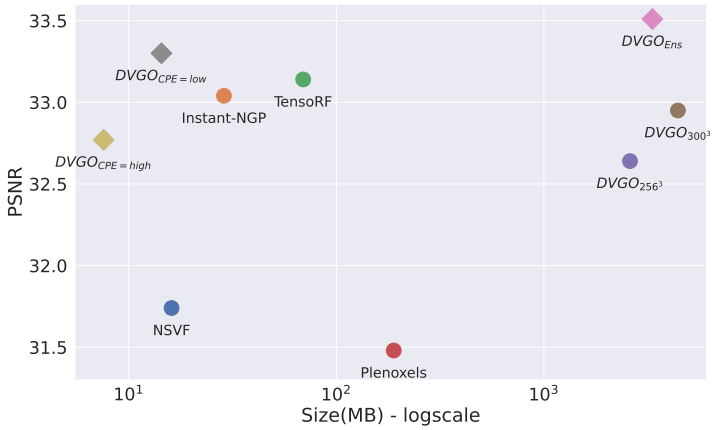


Fig. 5: Comparison of our ensembling strategy with state-of-the-art methods.

NGP [17] and TensoRF [3]. Similarly, in this case, our method has proven to be reliable, surpassing the current state-of-the-art in terms of both quality and memory footprint. In Fig. 6 we propose a study on the Lego dataset, in which we investigate various ensemble resolutions of up to 6 models. Our findings reveal that even a combination of just two models can result in a significant performance improvement of over 1 dB. While incorporating more than two models can lead to even greater performances, this approach also results in highly complex models with significantly more parameters and memory footprint.

5 Conclusion

In this work, we explored the potential of NeRF ensembling to improve performance. Specifically, we have sided a low-resolution architecture to a higher

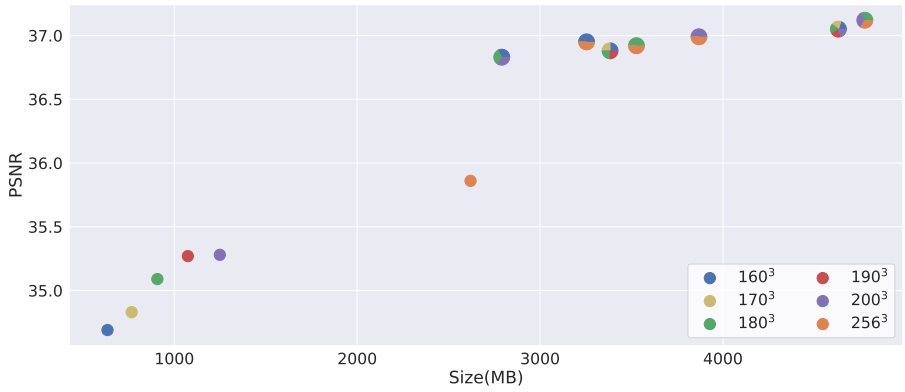


Fig. 6: Results on different ensembling resolutions on Lego dataset. Each pie chart represents an ensemble composed of a number of models equal to the number of slices.

one. Besides, a compression strategy, siding the ensemble, creates the perfect synergy for extracting the best performance out of a restricted number of parameters. We have observed consistent performance improvements on a broad variety of tested resolutions, under the same number of parameters. Our results demonstrate that ensembling can be a promising approach to improving NeRF performance, and further exploration of this method will be conducted in the next future. However, there are still several challenges associated with ensembling that need to be addressed. For example, how to select the appropriate combination of models in the ensemble and how to effectively combine their predictions. Overall, our work represents a step toward the development of an ultimate, highly-performing, and efficient NeRF ensembling strategy. Future research in this area could focus on addressing the challenges associated with ensembling and exploring more advanced techniques to improve performance.

References

1. Boss, M., Braun, R., Jampani, V., Barron, J.T., Liu, C., Lensch, H.: Nerd: Neural reflectance decomposition from image collections. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12684–12694 (2021)
2. 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)
3. Chen, A., Xu, Z., Geiger, A., Yu, J., Su, H.: Tensorf: Tensorial radiance fields. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII. pp. 333–350. Springer (2022)

4. Deng, C.L., Tartaglione, E.: Compressing explicit voxel grid representations: fast nerfs become also small. In: 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 1236–1245 (2023)
5. 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)
6. Gafni, G., Thies, J., Zollhofer, M., Nießner, M.: Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8649–8658 (2021)
7. Gao, C., Saraf, A., Kopf, J., Huang, J.B.: Dynamic view synthesis from dynamic monocular video. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5712–5721 (2021)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778. IEEE (2016)
9. Kosiorek, A.R., Strathmann, H., Zoran, D., Moreno, P., Schneider, R., Mokrá, S., Rezende, D.J.: Nerf-vae: A geometry aware 3d scene generative model. In: International Conference on Machine Learning. pp. 5742–5752. PMLR (2021)
10. Li, G., Xu, C., Chen, M., Han, Z., Zhang, J., Zeng, B., Lai, Y.K., Guo, B.: Neural volumetric rendering for metal microstructure design. ACM Transactions on Graphics (TOG) **39**(4), 1–14 (2020)
11. Li, Z., Niklaus, S., Snavely, N., Wang, O.: Neural scene flow fields for space-time view synthesis of dynamic scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6498–6508 (2021)
12. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C., Berg, A.C.: SSD: Single shot multibox detector. In: European Conference on Computer Vision. pp. 21–37. Springer (2016)
13. Liu, Y., Liu, S., Xu, K.: Point2surf: Learning implicit surfaces from point clouds with a multiscale feature network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
14. 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)
15. Martin-Brualla, R., Radwan, N., Sajjadi, M.S., Barron, J.T., Dosovitskiy, A., Duckworth, D.: Nerf in the wild: Neural radiance fields for unconstrained photo collections. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7210–7219 (2021)
16. 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)
17. 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)
18. Noguchi, A., Sun, X., Lin, S., Harada, T.: Neural articulated radiance field. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5762–5772 (2021)
19. 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)

20. Schwarz, K., Liao, Y., Niemeyer, M., Geiger, A.: Graf: Generative radiance fields for 3d-aware image synthesis. Advances in Neural Information Processing Systems **33**, 20154–20166 (2020)
21. Srinivasan, P.P., Deng, B., Zhang, X., Tancik, M., Mildenhall, B., Barron, J.T.: Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7495–7504 (2021)
22. 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)
23. Tancik, M., Mildenhall, B., Wang, T., Schmidt, D., Srinivasan, P.P., Barron, J.T., Ng, R.: Learned initializations for optimizing coordinate-based neural representations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2846–2855 (2021)
24. Tretschk, E., Tewari, A., Golyanik, V., Zollhöfer, M., Lassner, C., Theobalt, C.: Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12959–12970 (2021)
25. 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)
26. Yen, E.C.T., Liu, Y., Zhang, Z., Martin-Brualla, R., Ernst, J., Huang, M., Tong, X.: Nerf++: Analyzing and improving neural radiance fields. arXiv preprint arXiv:2010.07492 (2021)
27. Yen, Y., Liu, Z., Mitra, N.J.: Multiscale neural voxelization for high-resolution 3d object representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11632–11641 (2021)
28. Zhang, R., Lin, Z., Zhang, Y., Wang, Y., Zhou, P., He, R., Sun, J.: Neural radiance flow for 4d view synthesis and video processing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6914–6924 (2021)