Compressible-composable NeRF via Rank-residual Decomposition
- Supplementary Material -

Jiaxiang Tang[1], **Xiaokang Chen**[1], **Jingbo Wang**[2], **Gang Zeng**[1,3]
[1]School of Intelligence Science and Technology, Peking University
[2]Chinese University of Hong Kong [3]Intelligent Terminal Key Laboratory of SiChuan Province
{tjx, pkucxk}@pku.edu.cn, wj020@ie.cuhk.edu.hk, zeng@pku.edu.cn

A Additional Implementation Details

A.1 Model Settings

We detail the different settings for our model in this section. Our model setting can be denoted by $R_{\text{vec}}^{\text{density}}/R_{\text{mat}}^{\text{density}}$-$R_{\text{vec}}^{\text{color}}/R_{\text{mat}}^{\text{color}}$, corresponding to the number of vector- and matrix-based rank components for the density and color features. In particular, we focus on compressing the color components, since the density components take much less storage compared to the color components, and a correct density is the basis for correct color. In the main paper, we choose three different model settings to test the capability of our model at different scales. (1) The CP model only uses vector-based rank components. It takes less storage, but needs a relatively higher number of ranks to reach good rendering quality. Since the runtime GPU usage is determined by the total number of ranks, the CP model also leads to higher GPU memory usage and slower model querying. The full model uses 96 ranks for density, and 384 ranks for color. We use $M = 4$ groups for the rank-residual learning, so each group contains 96 ranks for color. (2) The HY model uses both vector- and matrix-based components. The use of matrix components takes more storage, but efficiently reduces the needed number of ranks, which consequently reduces GPU memory usage. We use 64 vector components and 16 matrix components for density, as well as 256 vector components and 64 matrix components for color. $M = 4$ groups are used for the rank-residual learning of color, with each group containing 64 vector components and 16 matrix components. (3) The HY-S model also uses both vector- and matrix-based components, but allows the model size to be adjusted in a larger range. Both density and color initially use 96 vector components. We then additionally append 4 groups for color, increasing the total number of matrix components to $\{4, 16, 32, 64\}$ gradually. Therefore, there are in total $M = 5$ groups including the first group with only 96 vector components.

A.2 Training Details

We train the model for $30,000$ iterations on both datasets, using the Adam optimizer [2] with an initial learning rate of 0.02 for the factorized matrices, and 0.001 for the singular values. To accelerate the training and inference, we adopt the occupancy pruning technique used in [1, 5]. An occupancy grid is pre-calculated at the $2,000$ and $4,000$ steps of training, and used to prune the non-occupied points along each ray. We also shrink the bounding box based on the occupancy grid at step $2,000$ for a more accurate modeling following [1]. This occupancy grid is also keeped in scene composition. As the transformation matrices, we record the occupancy grid for each single object, and use the warped rays to query the occupancy. We also apply the standard L1 regularization on the factored matrices for density following [1]. This regularization encourages sparsity in the parameters for tensor rank decomposition, which is shown to be beneficial in extrapolating novel views and removing floaters.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

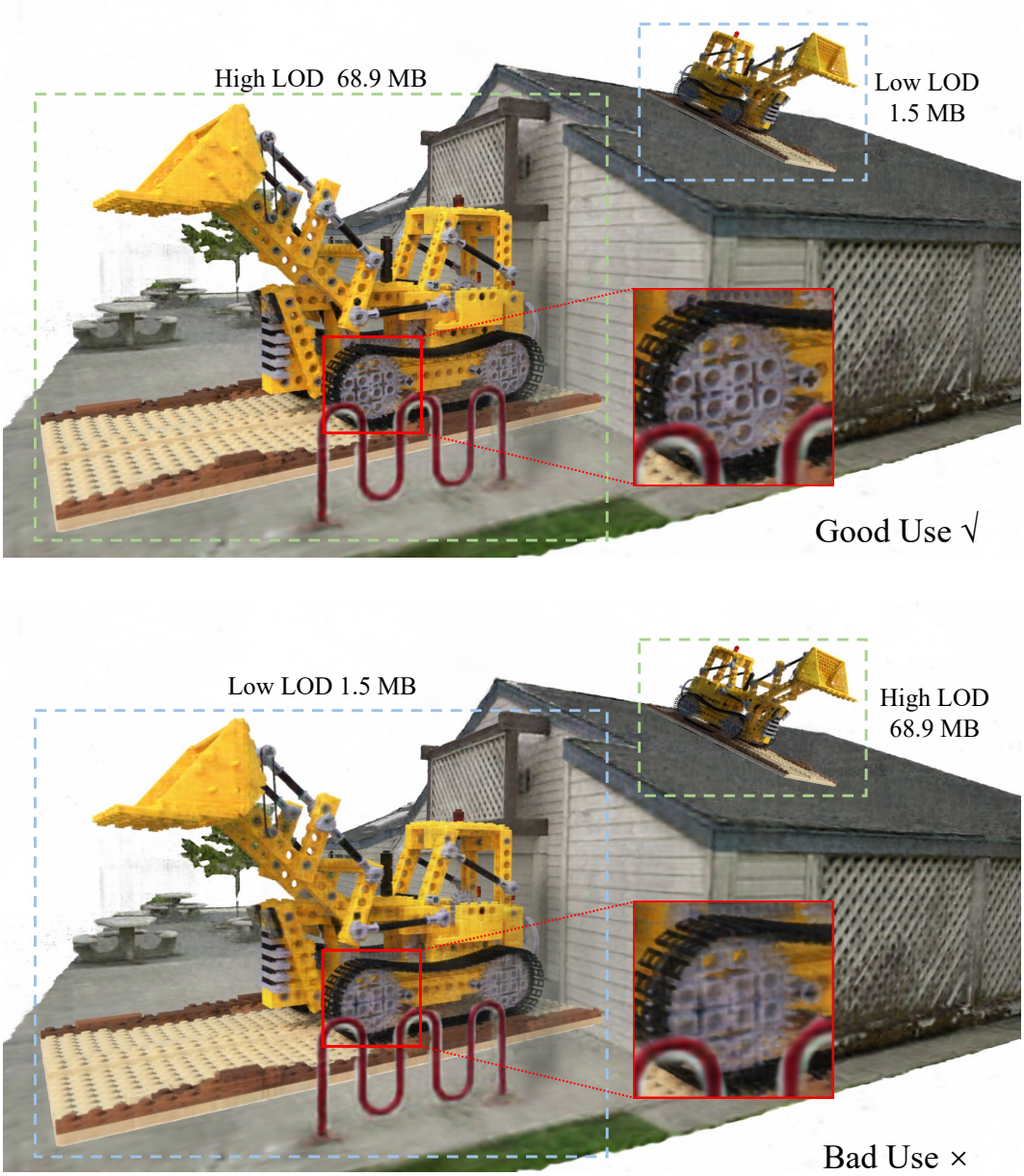

Figure 1: **Dynamic adjustment of level of detail (LOD).** We show an example of using different LODs in practice. The top image is a good use case, while the bottom image is a bad use case for comparison. Different rendering quality of different LODs are marked by the red box. By dynamically adjusting each model's LOD, we can balance between model size and performance, allowing more flexibility in scene composition.

A.3 Difference from TensoRF-SH

TensoRF [1] first brings the idea of tensor rank decomposition into modeling a neural radiance field. We now discuss the relationships between our method and TensoRF. In TensoRF, a Sphere Harmonics (SH) renderer variant can be used in place of the MLP renderer. However, due to the slightly worse performance of the SH variant, the authors decide to use the MLP renderer in the final design, and don't consider the compressibility or composability. For the CP decomposition, our model's structure is identical to the TensoRF-SH variant. The basis matrix $\mathbf{B}$ in TensoRF plays the same role as our rank weight matrix $\mathbf{S}$, except that we further explore its role in low-rank approximation. TensoRF also

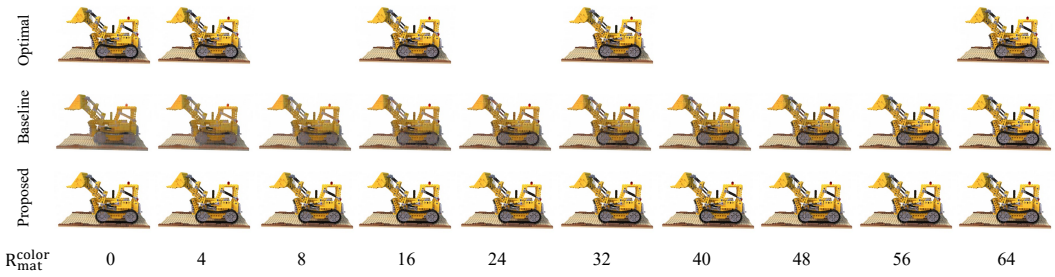

Figure 2: **Visualization of any rank compression.** Qualitative results of different compression strategies. The proposed method reaches near-optimal quality and doesn't need the retraining for the optimal models, while the baseline sort-and-truncate strategy fails.

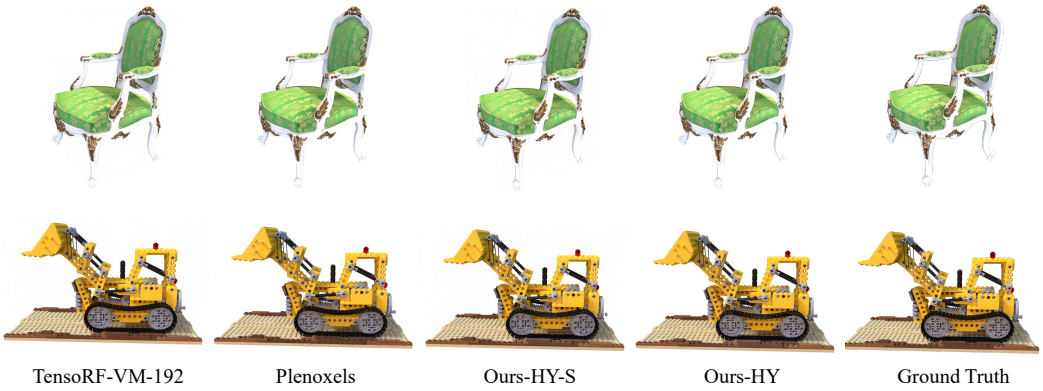

Figure 3: **Comparisons with recent works.** We provide qualitative comparisons with recent methods. Although quantitatively our model is only comparable to the state-of-the-arts, the rendering quality is good enough in most cases.

proposed a vector-matrix (VM) decomposition, but it is distinct from our hybrid (HY) decomposition. The most important property of our HY decomposition is the ability to arbitrarily adjust the ratio of vector- and matrix-based rank components (as demonstrated by our HY-S model setting). However, the VM decomposition requires the same rank for vector- and matrix-based rank components (*i.e.*, the ratio is always $1:1$), which limits the flexibility of model size controlling.

B Additional Experimental Results

B.1 Visualization of Different Levels of Detail

We show in Figure 1 an example of adopting different LODs for the same object in different cases. We transform and compose two bulldozers from the nerf-synthetic dataset [4] into the barn scene from the Tanks And Temples dataset [3]. The left bulldozer is placed as the primary object and requires more details for better visual effect, while the top right bulldozer is scaled down as a secondary object with fewer details. Our models are suitable for such cases due to the compressibility. We can use the full model for the primary object, while truncating it to a smaller model for the secondary object. This won't harm the quality of the overall rendering, but significantly saves storage space.

B.2 Visualization of the Near-optimal Compression

In Figure 2, we visualize the results for different compression strategies. We use the HY-S model on the LEGO dataset as an example, and only apply rank-residual supervision in 5 groups (*i.e.*, $M = 5$) where we also retrain an optimal model. The baseline models are gradually truncated from the optimal model of the full rank. This further demonstrates the near-optimal low-rank approximation property of the proposed rank-residual learning.

Table 1: **Training time.** Comparison of the training time and PSNR between different methods.

Method	Time	PSNR↑
NeRF [4]	$\sim$ 35h	31.01
TensoRF-CP-384 [1]	25m	31.56
TensoRF-VM-192 [1]	17m	33.14
Plenoxels [6]	11.4m	31.71
Instant-ngp [5]	5m	33.18
Ours-CP	29m	30.55
Ours-HY-S	30m	31.22
Ours-HY	41m	32.37

Table 2: **SH degrees.** Comparison of training time and PSNR between different SH degrees.

Degree	1	2	3	4
Time (minute)	54	44	49	73
PSNR	28.29	28.97	28.99	28.65

B.3 Qualitative Comparisons with SOTA

We provide qualitative comparisons with recent works in Figure 3. Although we only reach comparable quantitative results to the state-of-the-art method like TensoRF [1], the difference in rendering quality is hard to discern in most cases. Such rendering quality is enough to use for our compression and composition property.

B.4 Time Analysis

In Table 1, we also report the average training time over the nerf-synthetic dataset. Although our method cannot reach same level of speed compared to the recent works focusing on this problem [6, 5], it has been much more efficient compared to the vanilla NeRF [4]. Due to the rank-residual learning, the training time of our method is increased compared to TensoRF [1].

C Additional Ablation Study

C.1 Influence of the SH degrees

In Table 2, we performed an ablation study on the SH degrees with the materials dataset, which contains rich view-dependent effects. We show that SH degree of 3 achieves the best PSNR to model view-dependent effects. With too few or too many degrees, *e.g.*, 1 or 4, the model is hard to converge and takes more time to train.

C.2 Parallel rank-residual training

To verify the proposed parallel rank-residual training strategy, we experimented with three settings on the chair dataset in Table 3: (1) sequentially training per stage until its convergence, freezing the previous stages before training a new stage, (2) parallel training all stages, but for each stage we detach the output from the previous stages so each loss only applies on its corresponding rank group, which can be viewed as training each stage independently in parallel, (3) parallel training all stages without detaching, so each loss applies to all its previous rank groups.

The first setting is significantly slower to assure convergence of each stage. The second setting's final performance is worse due to optimizing later stages with not fully converged earlier stages. Compared to the first two settings, we think the third setting eases the training of the earlier stages, by letting the later stages model the complex details. Therefore, the earlier stages can focus on the fundamentals.

Table 3: **Parallel rank-residual training.** Comparison of the training time and PSNR on parallel and sequential training strategy.

Settings	Sequential	Parallel w/ detaching	Parallel w/o detaching
Time (minute)	83	28	26
PSNR	34.16	33.69	34.37

References

[1] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. *arXiv preprint arXiv:2203.09517*, 2022.

[2] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[3] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):1–13, 2017.

[4] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[5] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *arXiv:2201.05989*, January 2022.

[6] Sara Fridovich-Keil and Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *CVPR*, 2022.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [No]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]