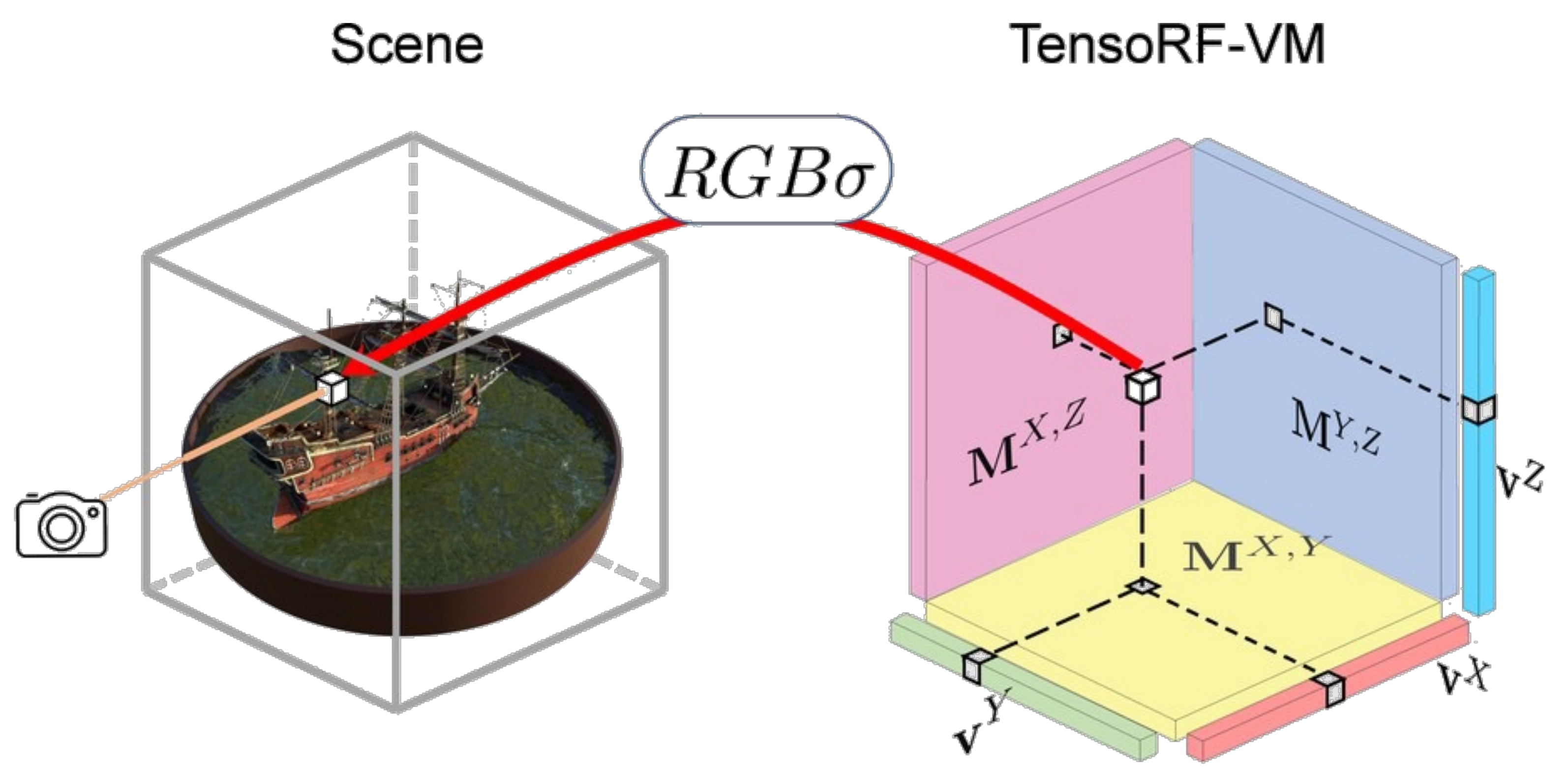
TensorRF: Tensorial Radiance Fields

Anpei Chen[1*], Zexiang Xu[2*], Andreas Geiger[3], Jingyi Yu[1], Hao Su[4]
[1]ShanghaiTech University [2]Adobe Research [3]University of Tübingen and MPI-IS, Tübingen [4]UC San Diego
* Indicates equal contribution

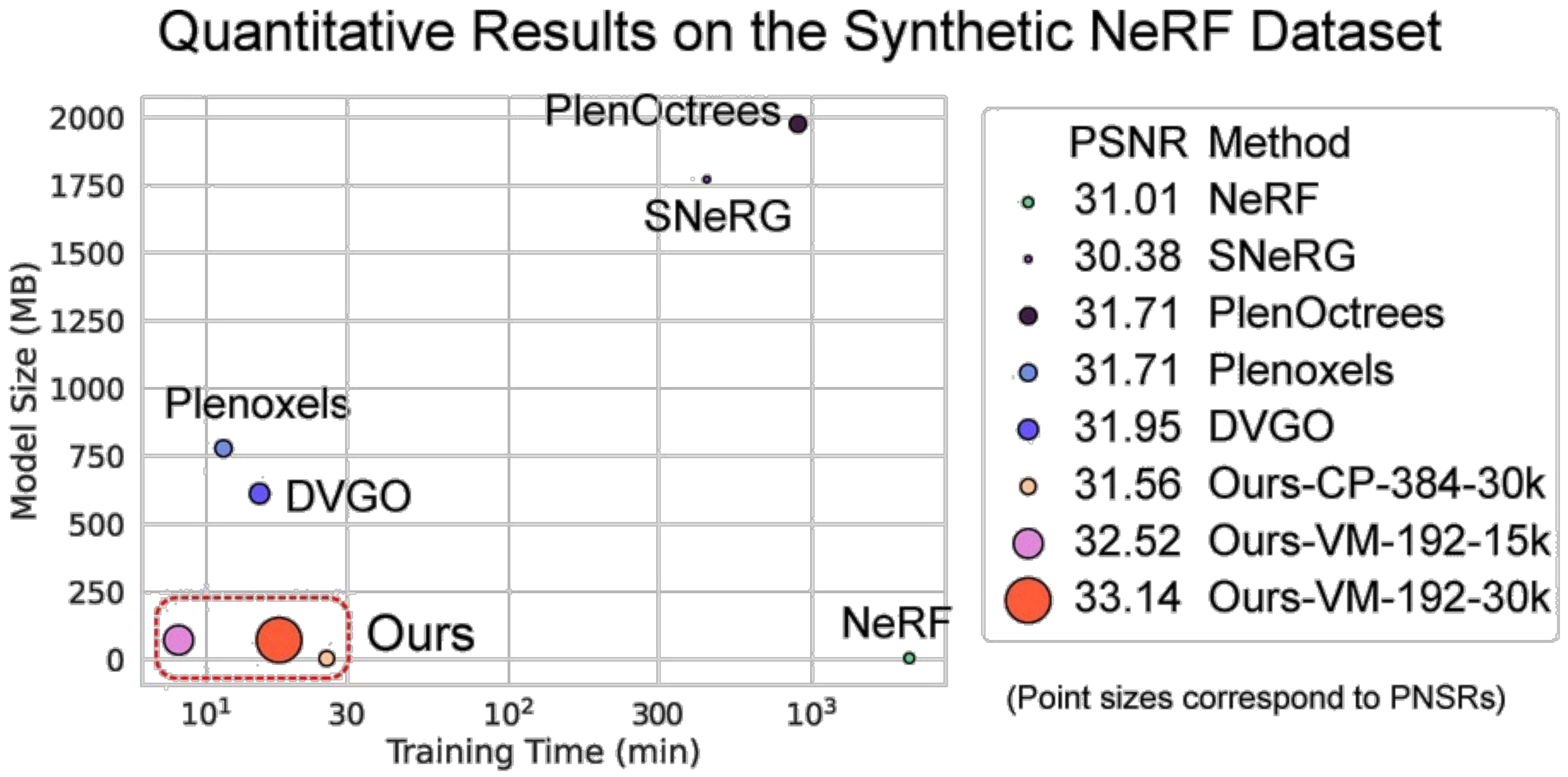
ECCV EUROPEAN CONFERENCE ON COMPUTER VISION TEL AVIV 2022
October 23-27, 2022, Tel Aviv

Scenes Modeling with Tensor Decomposition

We present a novel approach to model and reconstruct radiance fields. Unlike NeRF that uses pure MLPs,
we consider the full volume field as a 4D tensor and propose to factorize the tensor into a set of vectors and matrices that describe scene appearance and geometry along their corresponding axes.



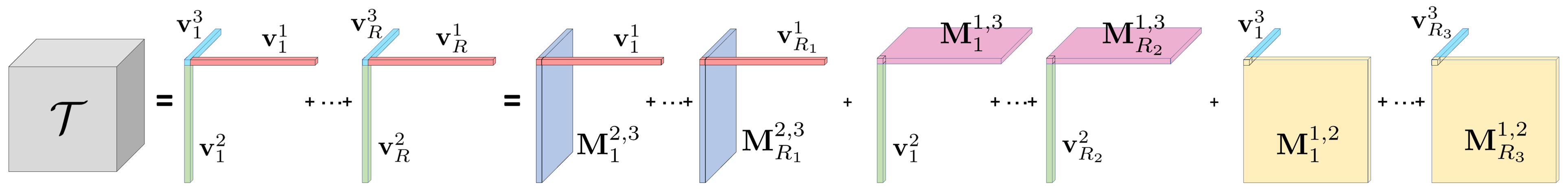
Performance Overview



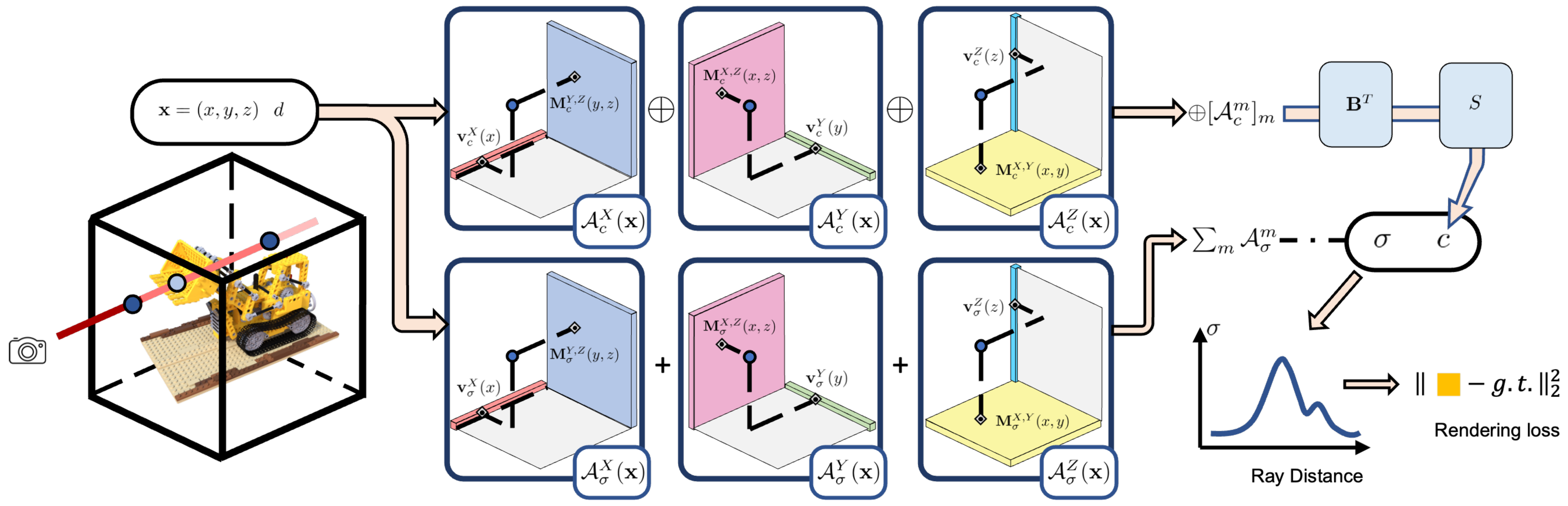
- We demonstrate that TensoRF with CP decomposition can achieve fast reconstruction with better rendering quality and even a smaller model size (**<4MB**) than NeRF.
- Moreover, TensoRF with VM decomposition can further boost our rendering quality to outperform previous state-of-the-art methods and reduce the reconstruction time (**<10min** only with standard PyTorch implementation).

Method

We factorize radiance fields into compact components for scene modeling. To doso, we apply both the classic CP decomposition and a new vector-matrix (VM) decomposition; both are illustrated in following figure:



Left: CP decomposition, which factorizes atensor as a sum of vector outer products. Right: our vector-matrix decomposition, which factorizes a tensor as a sum of vector-matrix outer products. Please refer to our paper for more decomposition derails.



We now present our TensoRF representation and reconstruction.

For each shading location **x** = (x,y,z), we use linearly/bilinearly sampled values from the vector (**V**)/matrix (**M**) factors to compute the corresponding trilinearly interpolated values of the tensor components.

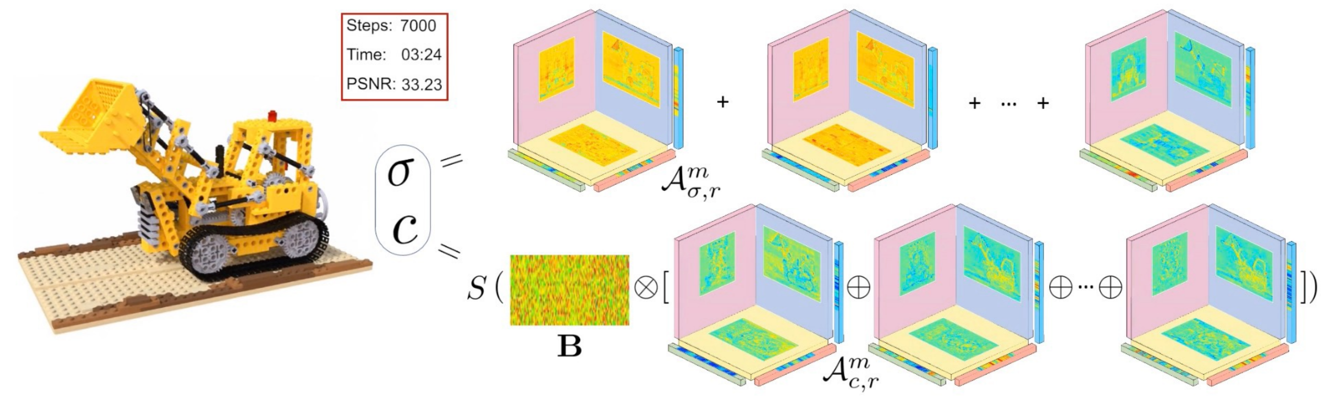
The density component values ($A_\sigma(x)$) are summed to get the volume density directly (σ). The appearance values ($A_c(x)$) are concatenated into a vector ($\oplus[A_c^m(x)]_m$) that is then multiplied by an appearance matrix (**B**) and sent to the decoding function S for RGB color (c) regression.

The decoding function S can be a Spherical Harmonic (SH) function or a fully-connected network (FCN).

Table

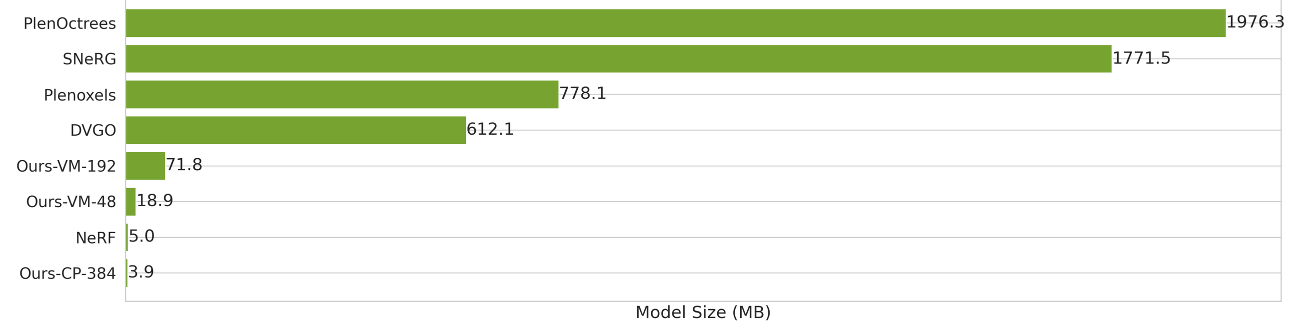
Method	BatchSize	Steps	Synthetic-NeRF Time↓	Size(MB)↓	PSNR↑	SSIM↑	NSVF PSNR↑	SSIM↑	TanksTemples PSNR↑	SSIM↑
SRN [46]	-	-	>10h	-	22.26	0.846	24.33	0.882	24.10	0.847
NSVF [26]	8192	150k	>48*h	-	31.75	0.953	35.18	0.979	28.48	0.901
NeRF [31]	4096	300k	∼35h	5.00	31.01	0.947	30.81	0.952	25.78	0.864
SNeRG [17]	8192	250k	∼15h	1771.5	30.38	0.950	-	-	-	-
PlenOctrees [59]	1024	200k	∼15h	1976.3	31.71	0.958	-	-	27.99	0.917
Plenoxels [43]	5000	128k	11.4m	778.1	31.71	0.958	-	-	27.43	0.906
DVGO [47]	5000	30k	15.0m	612.1	31.95	0.957	35.08	0.975.	28.41	0.911
Ours-CP-384	4096	30k	25.2m	3.9	31.56	0.949	34.48	0.971	27.59	0.897
Our-VM-192-SH	4096	30k	16.8m	71.9	32.00	0.955	35.30	0.977	27.81	0.907
Ours-VM-48	4096	30k	13.8m	18.9	32.39	0.957	35.34	0.976	28.06	0.909
Ours-VM-192	4096	15k	8.1m	71.8	32.52	0.959	35.59	0.978	28.07	0.913
Ours-VM-192	4096	30k	17.4m	71.8	33.14	0.963	36.52	0.982	28.56	0.920

Super Fast Convergence



Note that, unlike concurrent works Plenoxels and Instant-ngp that require customized CUDA kernels, our model's efficiency gains are obtained using a standard PyTorch implementation.

Super Compact Memory Footprint

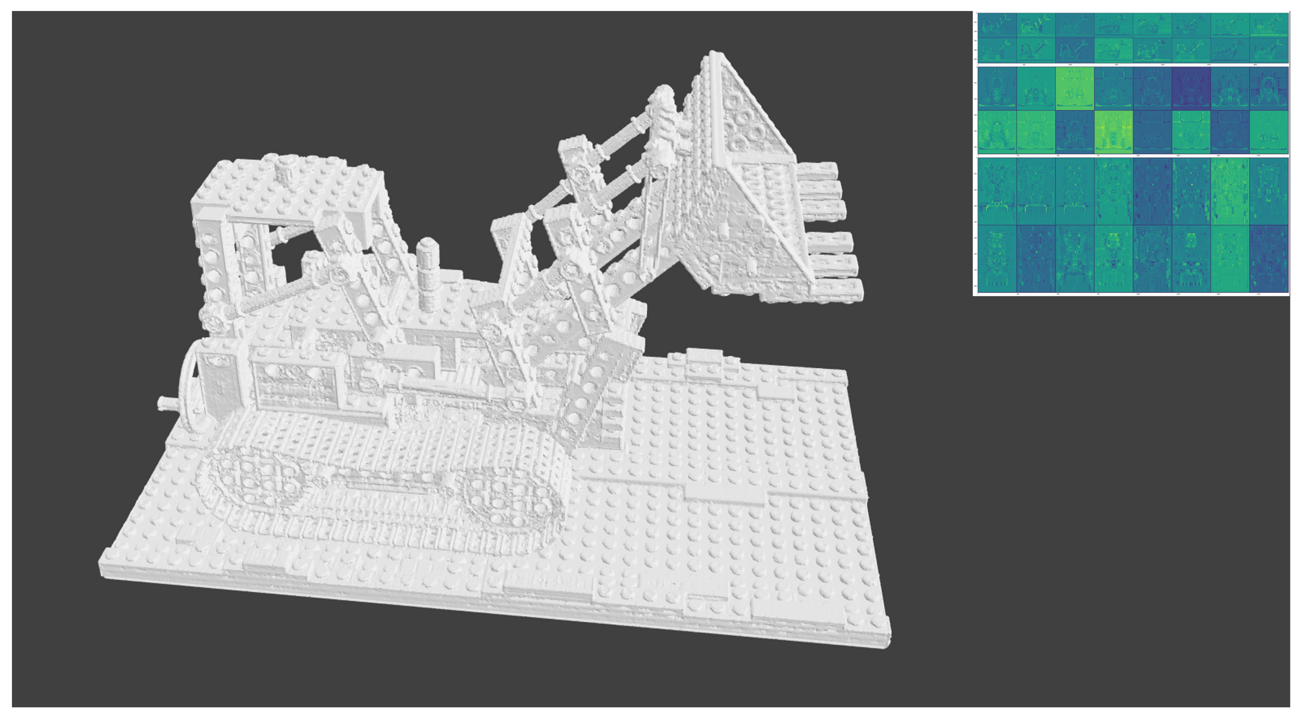


In contrast to previous works that directly reconstruct voxels, our tensor factorization reduces space complexity from O(n³) to O(n) (with CP) or O(n²) (with VM), significantly lowering memory footprint.

Super Vivid Details

Our approach can also achieve high-quality radiance field reconstruction for 360° objects and forward-facing scenes.

Geometric Visualization



Top right: here we visualize the trained density basis of the Lego scene, the number of the basis is 16 for each dimension. We normalize the basis with min/max value along each dimension thus the brightness corresponds to their energy.
We can also convert the above density basis to a mesh using marching cubes.

apchenstu.github.io/TensoRF/