Improving Robustness for Joint Optimization of Camera Poses and Decomposed LowRank Tensorial Radiance Fields

Bo-Yu Cheng Wei-Chen Chiu Yu-Lun Liu

Code available at : https://github.com/Nemo1999/Joint-TensoRF

AAAI 2024

Enriched Vision Applications Lab

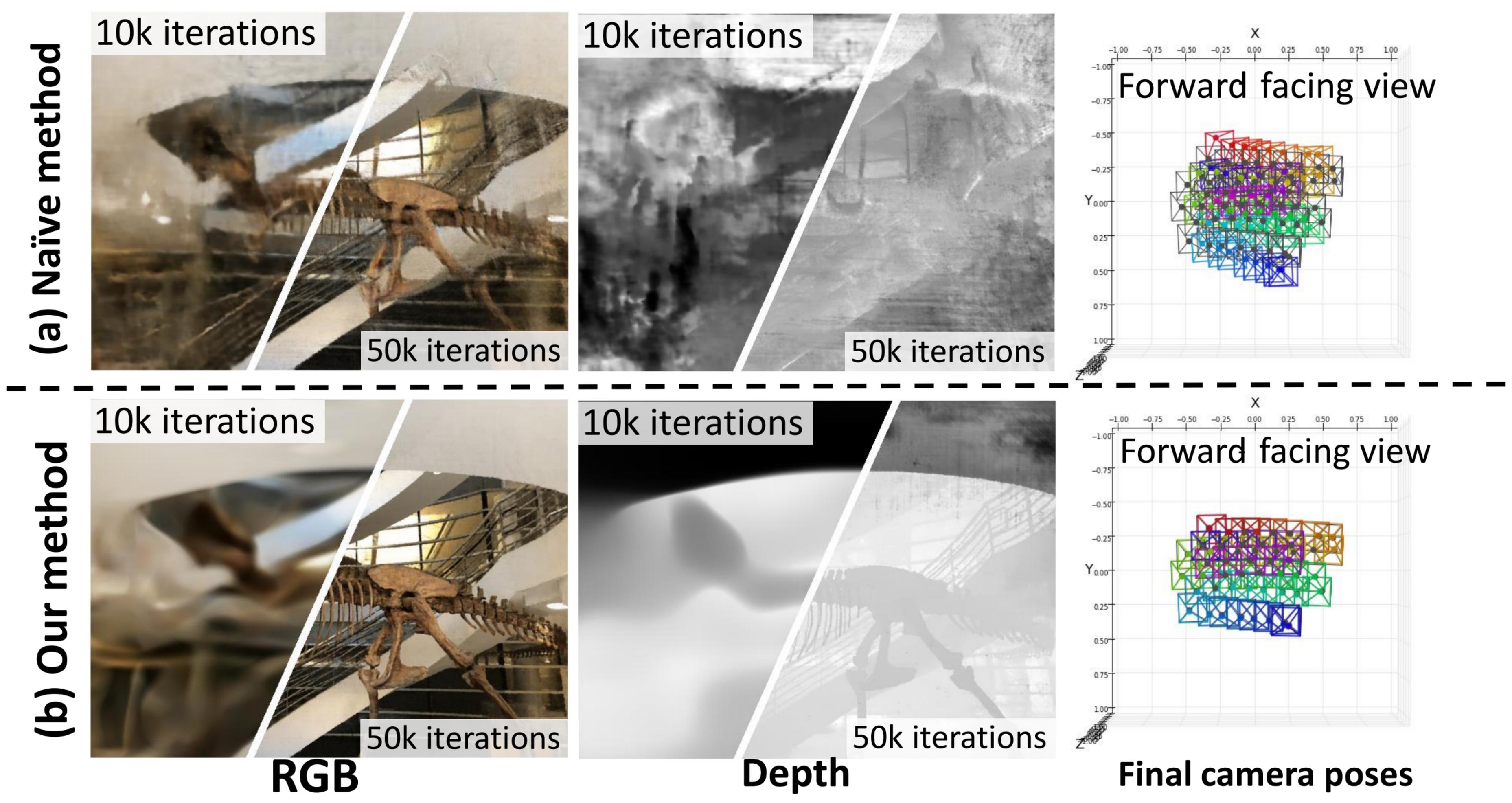
NATIONAL YANG MING CHIAO TUNG UNIVERSITY

Introduction

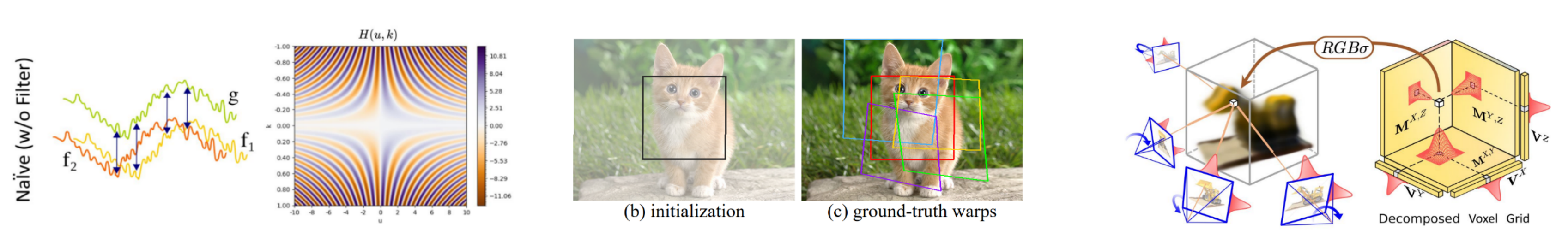
Goal: Following BARF, we enable joint optimization of camera pose on Tensorial Radiance Field, accelerating the joint optimization and getting better quality.

Challenge: Unlike MLP architecture used in BARF, voxel-based architectures lack spectral bias and is unstable in joint optimization.

Contribution: We solve the overfitting problem of naive method, and enable joint optimization of camera pose on TensoRF.



Proposed Methods: We start with 1D pilot study that discusses the effect of filtering strengh on the joint optimization, from which we propose various methods that are proven effective in 2D and 3D experiment.



1D Analysis — 2D Experiment — 3D Joint Optimization

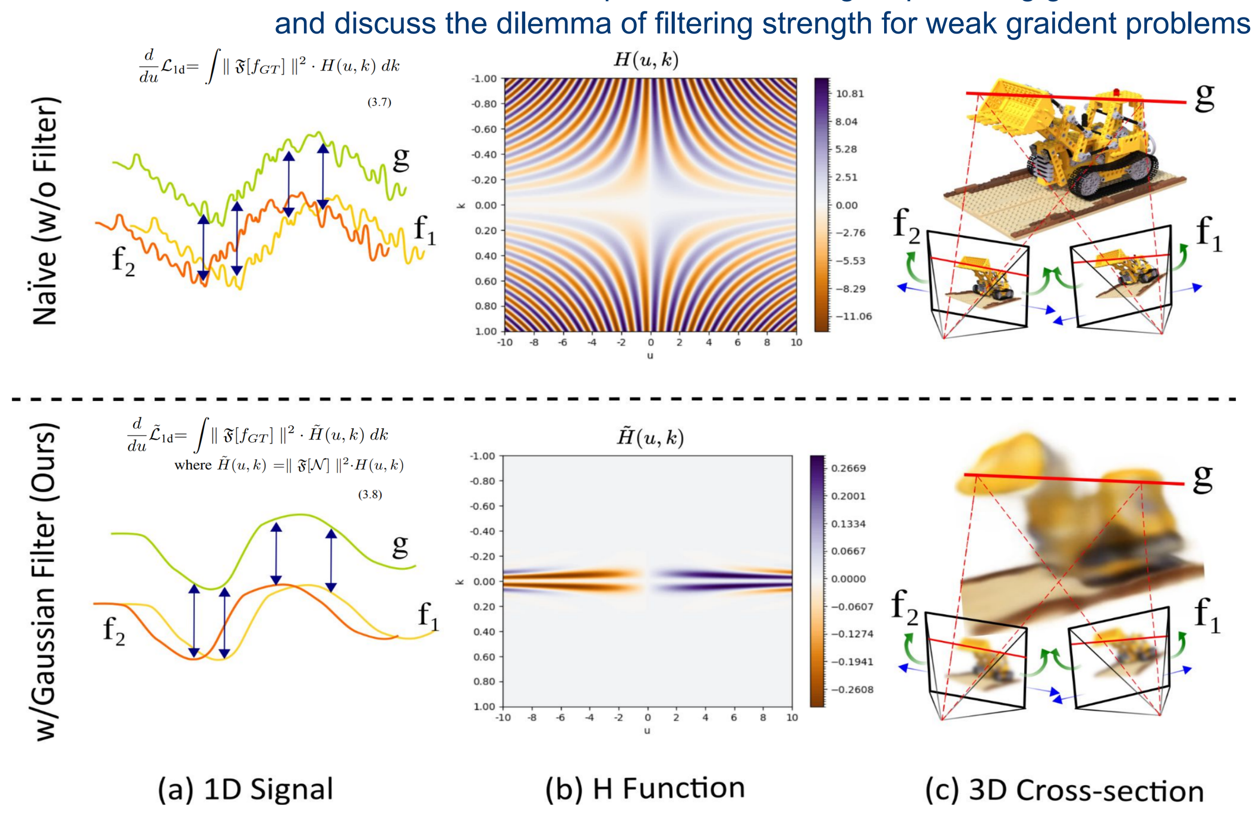
Demo: Brute-Force 2D Convolution

Spectrum property of 1D Gradient → Separated 3D Convolution; 2D Smoothed Supervision

Weak and Blurry Gradient Problem → Randomly Scaled Kernel; Edge Guided Loss Mask

1D Analysis

We analyze how the spectral property of 1D signal can affect joint alignment gradient. We also explains how filtering helps finding global minima, and discuss the dilemma of filtering strength for weak graident problems



2D Experiment

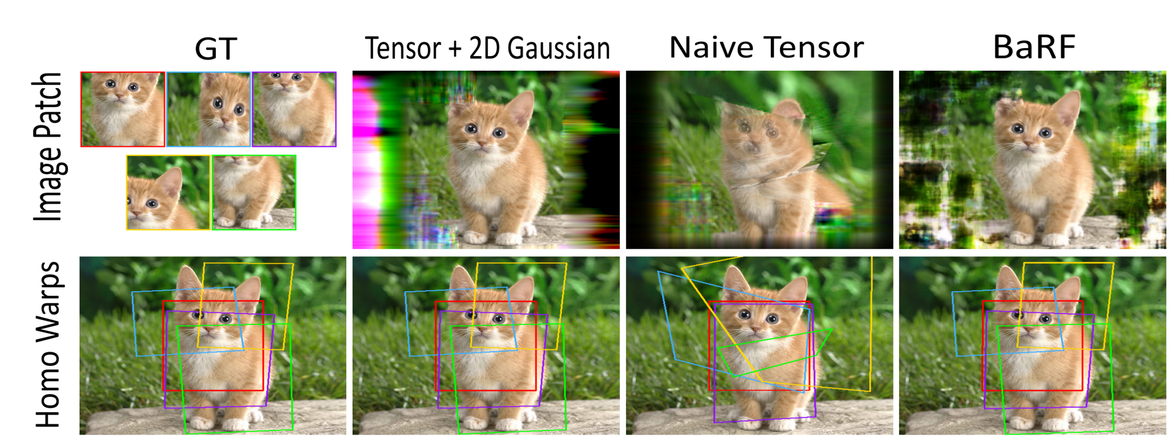
$$\mathcal{L}_{2d}(F_{2d}, \mathbf{P}_{2d}) = \sum_{i=1}^{L} \sum_{u \in \mathbf{U}_{2d}} \|F_{2d}(\mathcal{W}_{2d}(P_i, u)) - I_{iu}\|^2, \quad (9)$$

$$F_{2d}(\mathbf{x}) = (\mathcal{N}_{2d} *_{2d} \mathcal{T}_{2d})(\mathbf{x}) = (\mathcal{N}_{2d} *_{2d} (\sum_{r=1}^{R} \mathbf{v}_r^X \otimes \mathbf{v}_r^Y))(\mathbf{x}), \quad (10)$$

In 2D example, brute force 2D convolution out-performs BARF in both quality and efficiency

Methods	$\mathfrak{sl}(3)$ error ↓	patch PSNR ↑
BARF	0.0105	35.19
Naïve 2D TensoRF	0.5912	20.80
2D TensoRF + 2D Gaussian	**0.0023**	**40.70**

Table 1: **Quantitative results of planar image alignment.**



Fast Convergence: ***Separable Component-Wise Convolution*** allows efficient 3D spectrum control on tensorial field, which in terms prevent the need for MLP PE control in BARF and sequenctial multi-resolution grids learning in HASH (Heo et al. 2023).

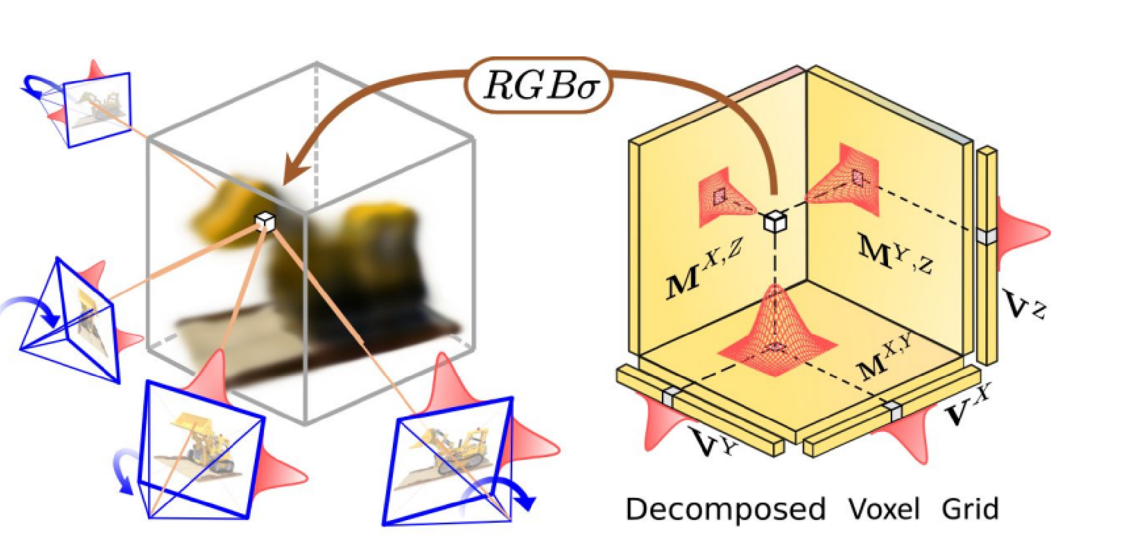
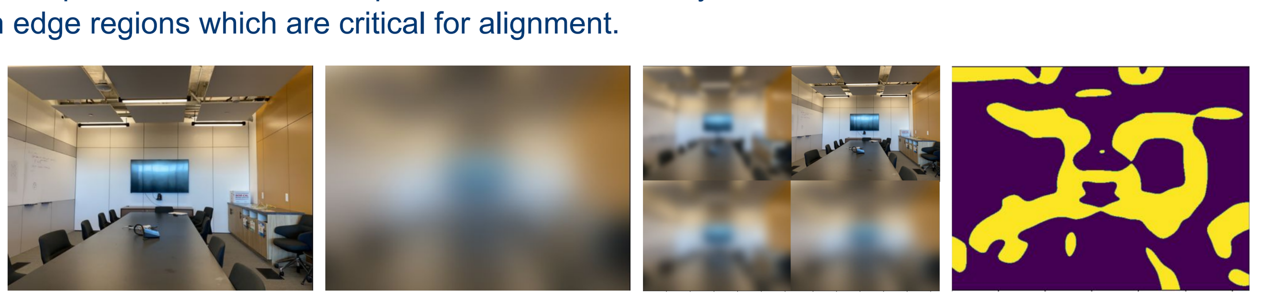


Figure 7: **PSNR and training iterations comparison.**

Improving Robustness: We propose ***Randomly Scaled Kernel*** and ***Edge Guided Loss Mask*** to improve the robustness of joint optimization, the former prevents local minima by randomized combination of 2D and 3D filtering, and the latter amplify gradient signal in edge regions which are critical for alignment.



(a) No Kernel (b) Overly Aggresive Kernel (c) Randomly Scaled Kernel (d) Edge Region

Algorithm : Our proposed 3D method combines

1. Separated 3D Convolution on tensorial field.
2. Smoothed 2D supervision.
3. Randomly Scaled Kernels for both 2D and 3D
4. Edge Guided Loss for amplifying more useful gradients

Ablation : We show the importance of each proposed components and also demonstrate the necessity by showing the methods proposed by BARF or GARF are not applicable to tensorial radiance field.

	3D Gauss.	2D Gauss.	Random Kernel	Edge Guided	Rot. ↓	Trans. ↓	PSNR ↑
(a)	✓	✓	✓	✓	**0.72**	**0.33**	**25.36**
(b)	✓	✓		✓	1.00	0.37	25.25
(c)	✓	✓			1.91	0.93	25.12
(d)	✓				33.00	12.7	20.10
(e)		✓			26.25	8.9	19.73
(d)					23.29	9.4	23.97

Table 4: **Ablation study of the components of the proposed method on the real-world LLFF dataset.**

	Rot. ↓	Trans. ↓	PSNR ↑	SSIM ↑	LPIPS ↓
TensoRF + BARF	45.47	0.17	20.71	0.630	0.314
TensoRF + GARF	73.92	0.29	10.47	0.287	0.679
Ours	0.43	0.003	26.92	0.872	0.104

Table 5: **Ablation on Directly Applying BARF and GARF on TensoRF (Potential Baseline)**

Algorithm 1: Conceptual Training Process for Our Proposed 3D Joint Optimization Training

$\mathcal{T}_\sigma, \mathcal{T}_c \leftarrow$ Initialize Voxel Grid
$\mathbf{P} \leftarrow$ Initialize Camera Poses
for $s = 1$ **to** train_iters **do**
  $i \leftarrow$ 2D_kernel_sched(s)
  $\sigma, c \leftarrow$ 3D_kernel_sched(s)
  $u_\sigma \leftarrow$ randomly sample density kernel scale
  $u_I \leftarrow$ randomly sample 2D kernel scale
  $\mathcal{N}_\sigma \leftarrow$ generate gaussian kernel with variance $(\sigma \cdot u_\sigma)^2$
  $\mathcal{N}_c \leftarrow$ generate gaussian kernel with variance c^2
  $\mathcal{N}_I \leftarrow$ generate gaussian kernel with variance $(i \cdot u_I)^2$
  $\tilde{\mathcal{L}}_{\text{joint}} \leftarrow \tilde{\mathcal{L}}_{\text{joint}}(\mathcal{T}_\sigma, \mathcal{T}_c, \mathbf{P}, \mathcal{N}_\sigma, \mathcal{N}_c, \mathcal{N}_I)$ (with randomly selected pixels in all training views)
  $\mathcal{L}_{\text{L1}} = \mathcal{L}_{\text{L1}}(\mathbf{v}_{\sigma,r}, \mathbf{M}_{c,r}, \mathbf{v}_{c,r}, \mathbf{M}_{\sigma,r})$
  $\mathcal{L}_{\text{TV}} = \mathcal{L}_{\text{TV}}(\mathbf{v}_{\sigma,r}, \mathbf{M}_{c,r}, \mathbf{v}_{c,r}, \mathbf{M}_{\sigma,r})$
  $\mathcal{L}_{3d} = w_1 \cdot \mathcal{L}_{\text{joint}} + w_2 \cdot \mathcal{L}_{\text{L1}} + w_3 \cdot \mathcal{L}_{\text{TV}}$
  back propagation
  update $\mathcal{T}_\sigma, \mathcal{T}_c, \mathbf{P}$
end for

3D Experiments

Quanlitative Results : Quantitative results shows superior synthesis qualtity compared to previous methods.

Scene	Camera Pose Registration								View Synthesis Quality							
	Rotation (°) ↓				Translation ↓				PSNR ↑				SSIM ↑			
	GARF	BARF	HASH	Ours	GARF	BARF	HASH	Ours	GARF	BARF	HASH	Ours	GARF	BARF	HASH	Ours
Chair	0.113	0.096	**0.085**	0.874	0.549	0.428	**0.365**	3.501	31.32	31.16	31.95	**35.22**	0.959	0.954	0.962	**0.984**
Drum	0.052	0.043	0.041	**0.037**	0.232	0.225	0.214	**0.118**	24.15	23.91	24.16	**25.78**	0.909	0.900	0.912	**0.934**
Ficus	0.081	0.085	0.079	**0.050**	0.461	0.474	0.479	**0.173**	26.29	26.26	28.31	**31.37**	0.935	0.934	0.943	**0.978**
Hotdog	0.235	0.248	0.229	**0.105**	1.123	1.308	1.123	**0.499**	34.69	34.54	35.41	**37.18**	0.972	0.970	0.981	**0.982**
Lego	0.101	0.082	0.071	**0.049**	0.299	0.291	0.272	**0.100**	29.29	28.33	31.65	**34.23**	0.925	0.927	0.973	**0.981**
Materials	**0.842**	0.844	0.852	0.854	**2.688**	2.692	2.743	2.690	27.91	27.84	27.14	**29.04**	0.941	0.936	0.911	**0.951**
Mic	0.070	0.071	**0.068**	1.177	0.293	0.301	**0.287**	5.000	31.39	31.18	32.33	**32.50**	0.971	0.969	0.975	**0.976**
Ship	0.073	0.075	0.079	**0.058**	0.310	0.326	0.287	**0.167**	27.64	27.50	27.92	**31.98**	0.862	0.849	0.879	**0.903**
Mean	0.195	0.193	**0.189**	0.400	0.744	0.756	**0.722**	1.533	28.96	28.84	29.86	**32.07**	0.935	0.930	0.943	**0.961**

Table 2: **Quantitative results on the NeRF-Synthetic dataset.** Our method achieves the best average novel-view synthesis quality and the best pose error in 5 out of 8 scenes. Notice that our method converges within 40k iterations, while all previous methods train for 200k iterations.

Scene	Camera Pose Registration								View Synthesis Quality							
	Rotation (°) ↓				Translation ↓				PSNR ↑				SSIM ↑			
	GARF	BARF	HASH	Ours	GARF	BARF	HASH	Ours	GARF	BARF	HASH	Ours	GARF	BARF	HASH	Ours
Fern	0.470	0.191	**0.110**	0.472	0.250	**0.102**	**0.102**	0.199	24.51	23.79	24.62	**26.17**	0.740	0.710	0.743	**0.842**
Flower	0.460	**0.251**	0.301	1.375	0.220	0.224	**0.211**	0.389	**26.40**	23.37	25.19	25.62	0.790	0.698	0.744	**0.810**
Fortress	**0.030**	0.479	0.211	0.449	0.270	0.364	**0.241**	0.419	29.09	29.08	**30.14**	29.68	0.820	0.823	**0.901**	0.882
Horns	**0.030**	0.304	0.049	0.386	0.210	0.222	**0.209**	0.251	22.54	22.78	**22.97**	22.84	0.690	0.727	0.736	**0.819**
Leaves	**0.130**	1.272	0.840	1.990	0.230	0.249	**0.228**	0.397	19.72	18.78	19.45	**21.24**	0.610	0.537	0.607	**0.753**
Orchids	0.430	0.627	0.399	**0.279**	0.410	0.404	0.386	**0.340**	19.37	19.45	20.02	**20.57**	0.570	0.574	0.610	**0.698**
Room	0.270	0.320	0.271	**0.188**	0.200	0.270	0.213	0.191	31.90	31.95	**32.73**	31.87	0.940	0.949	**0.968**	0.936
T-Rex	**0.420**	1.138	0.894	0.523	**0.360**	0.720	0.474	0.416	22.86	22.55	23.19	**24.19**	0.800	0.767	0.866	**0.878**
Mean	**0.280**	0.573	0.384	0.709	0.269	0.331	**0.258**	0.325	24.55	23.97	24.79	**25.27**	0.745	0.723	0.772	**0.827**

Table 3: **Quantitative results on the LLFF dataset.** Our method achieves the best average novel-view synthesis quality and best LPIPS in 7 out of 8 scenes. Our method converges within 50k iterations, while all previous methods train for 200k iterations.

Conclusion :

1. ***Theoretically***, we provide insights into the impact of scene properties on the convergence of joint optimization beyond the coarse-to-fine heuristic discussed in pror research, proposing a filtering based strategy for improving the joint optimization.

2. ***Algorithmically***, we introduce (and prove the equivalence of) an effective method for applying the filtering based strategy on decomposed low-rank tensor, notice that the propsed ***Separable Component-Wise Convolution*** is unique and more efficient than traditional separable methods in the sense that we (aside from the separated kernel) additionally utilize the separability of input signal. We additionaly proposed techniques (i.e. ***Randomly Scaled Kernel*** and ***Edge Guided Loss Mask***) for improving the robustness of joint optimization.

3. ***Comprehensive Evaluations*** demonstrates our proposed framework's state-of-the-art performance and rapid convergence.