MU-GeNeRF: Multi-view Uncertainty-guided Generalizable Neural Radiance Fields for Distractor-aware Scene

Wenjie Mu[1], Zhan Li[1], Chuanzhou Su[1], Xuanyi Shen[1], Ziniu Liu[1],Fan Lu[1], Yujian Mo[1], Junqiao Zhao[1], Tiantian Feng[1], Chen Ye[1], Guang Chen[1,2]

[1]Tongji University, [2]Shanghai Innovation Institute

Background: Why GeNeRF Fails in the Wild?

Generalizable Neural Radiance Fields (GeNeRF)

- Learns a universal multi-view prior → directly synthesizes novel views from sparse images
- No per-scene optimization required → efficient cross-scene reconstruction

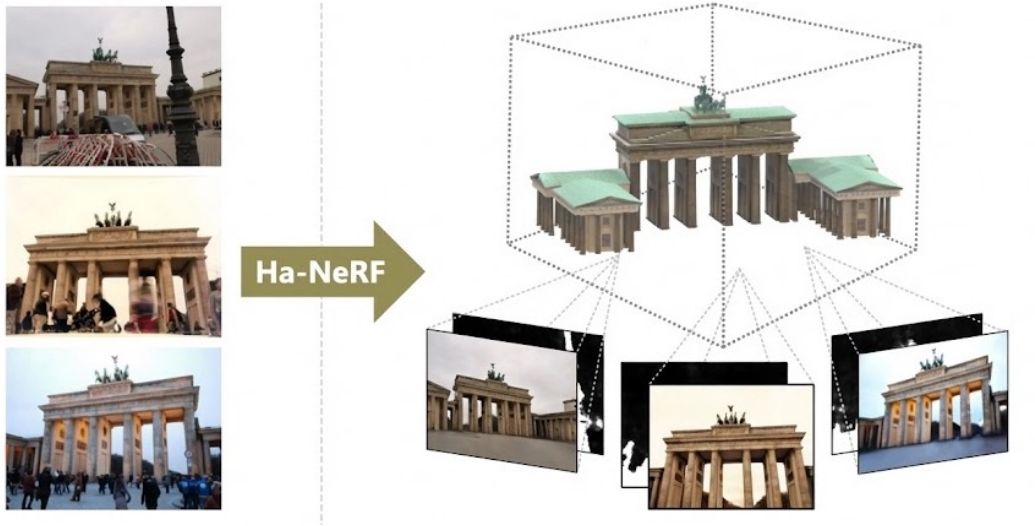
The Challenge: Transient Distractors in the Wild

- Dynamic objects, shadows, occlusions break cross-view consistency → Erroneous supervision corrupts the learned prior
- GeNeRF lacks a mechanism to distinguish distractors from static structure → Result: Blurred renderings, distorted geometry

Main Paradigms for Distractor-free NeRF

1) Explicit Transient Modeling (e.g., NeRF-W, Ha-NeRF)

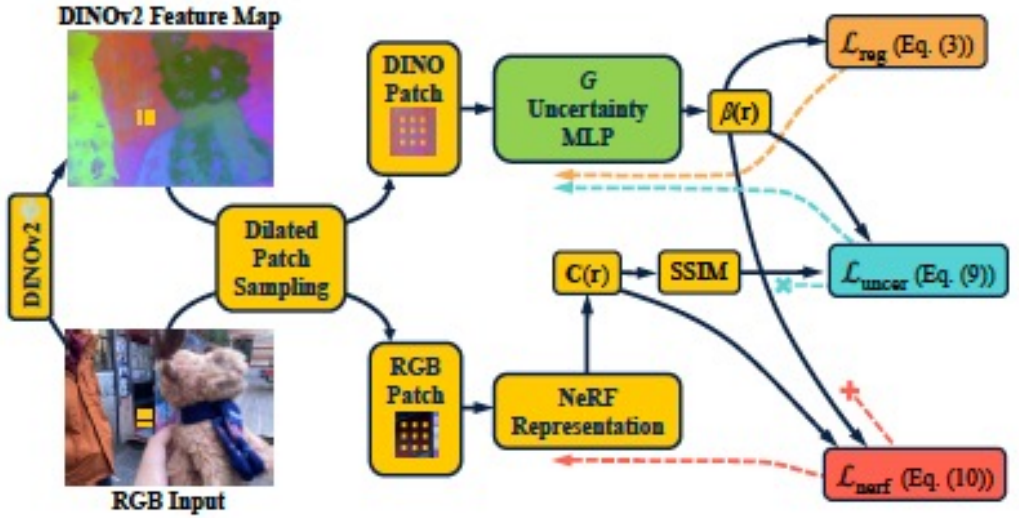
- Introduces a separate transient component into the rendering pipeline
- Estimates per-point uncertainty / transient density to explicitly decouple static scene from distractors
- Requires per-scene optimization to establish multi-view consistency



Ha-NeRF: Hallucinated Neural Radiance Fields in the Wild (CVPR 2022)

2) Robust Outlier Suppression (e.g., NeRF On-the-go)

- Treats transient distractors as high-uncertainty outliers during training
- Constructs a heteroscedastic loss to adaptively down-weight unreliable regions
- Implicitly filters distractors without explicit modeling



NeRF On-the-go: Exploiting Uncertainty for Distractor-free NeRFs in the Wild (CVPR 2024)

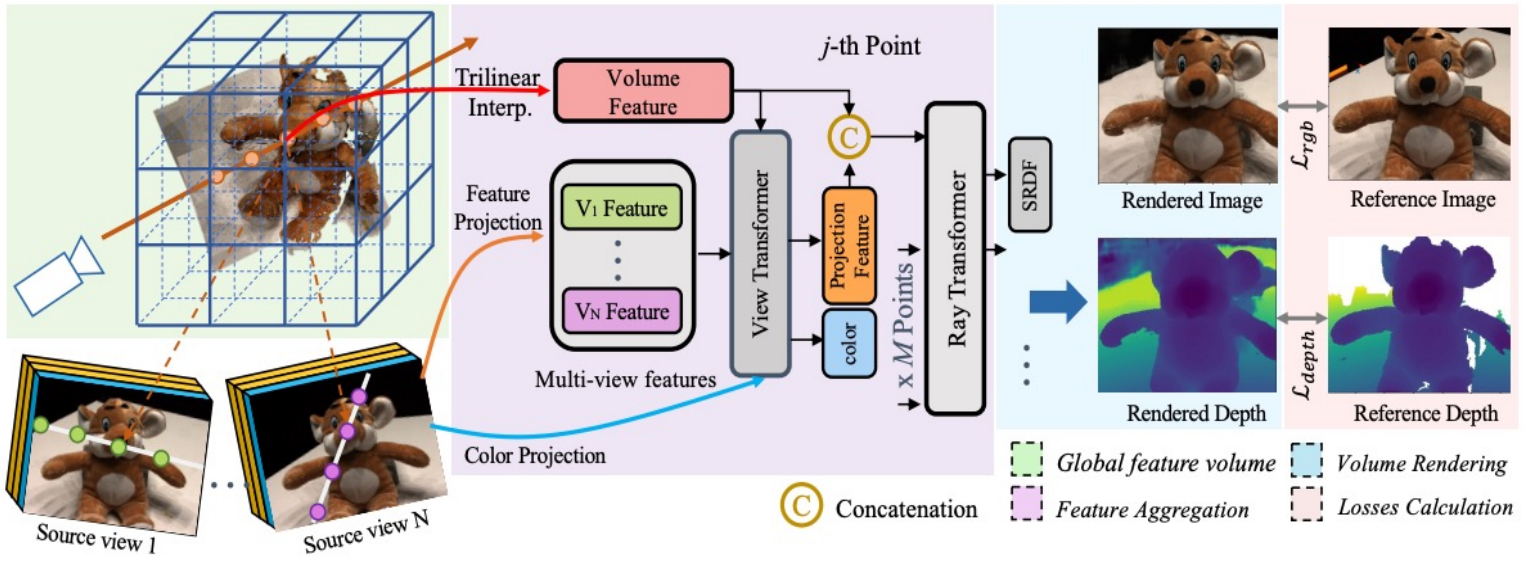
Why Existing Distractor-free Methods Cannot Transfer to GeNeRF?

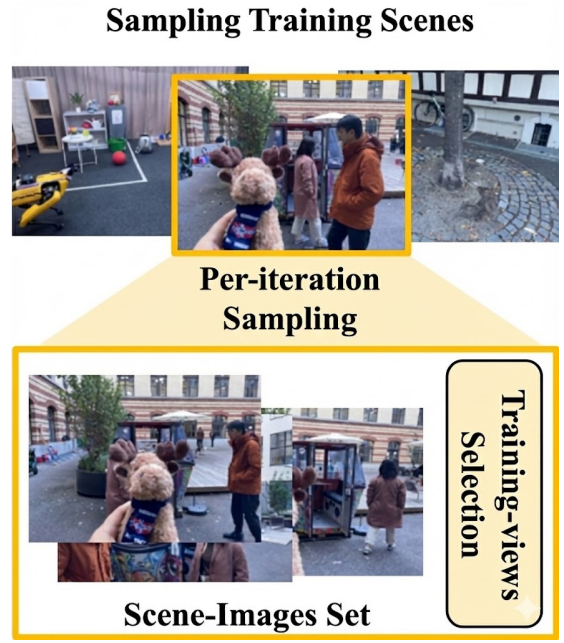
Per-scene NeRF paradigm:

- Estimates uncertainty from per-view reconstruction errors
- Relies on overfitting to a specific scene's noise pattern
- Essentially "memorizes" what is static vs. transient for that scene

GeNeRF is fundamentally different:

- No per-scene optimization — learns a shared multi-view aggregation prior
- Reconstruction errors have mixed origins: Transient distractors in the target view; Structural inconsistencies across source views (occlusion, dynamics)
- Indiscriminately treating all errors as distractors → misjudges inconsistent static structures → degrades geometry





[VolRecon: Volume Rendering of Signed Ray Distance Functions for Generalizable Multi-View Reconstruction (CVPR 2023)]

Multi-View Uncertainty-guided Distractor-aware Framework

Core idea:
Decompose reconstruction errors into two complementary uncertainties to robustly suppress distractors under the GeNeRF setting.

Effect:
Suppresses cross-view mismatches and distractors during training → cleaner supervision → more reliable geometric modeling without per-scene optimization.

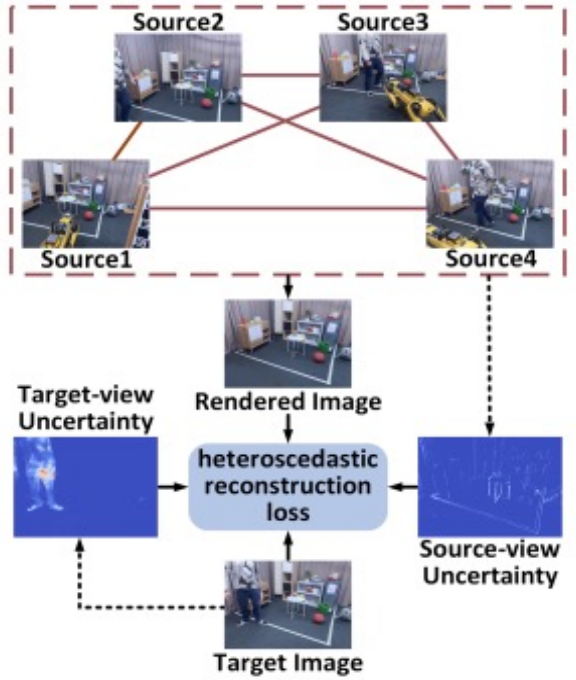
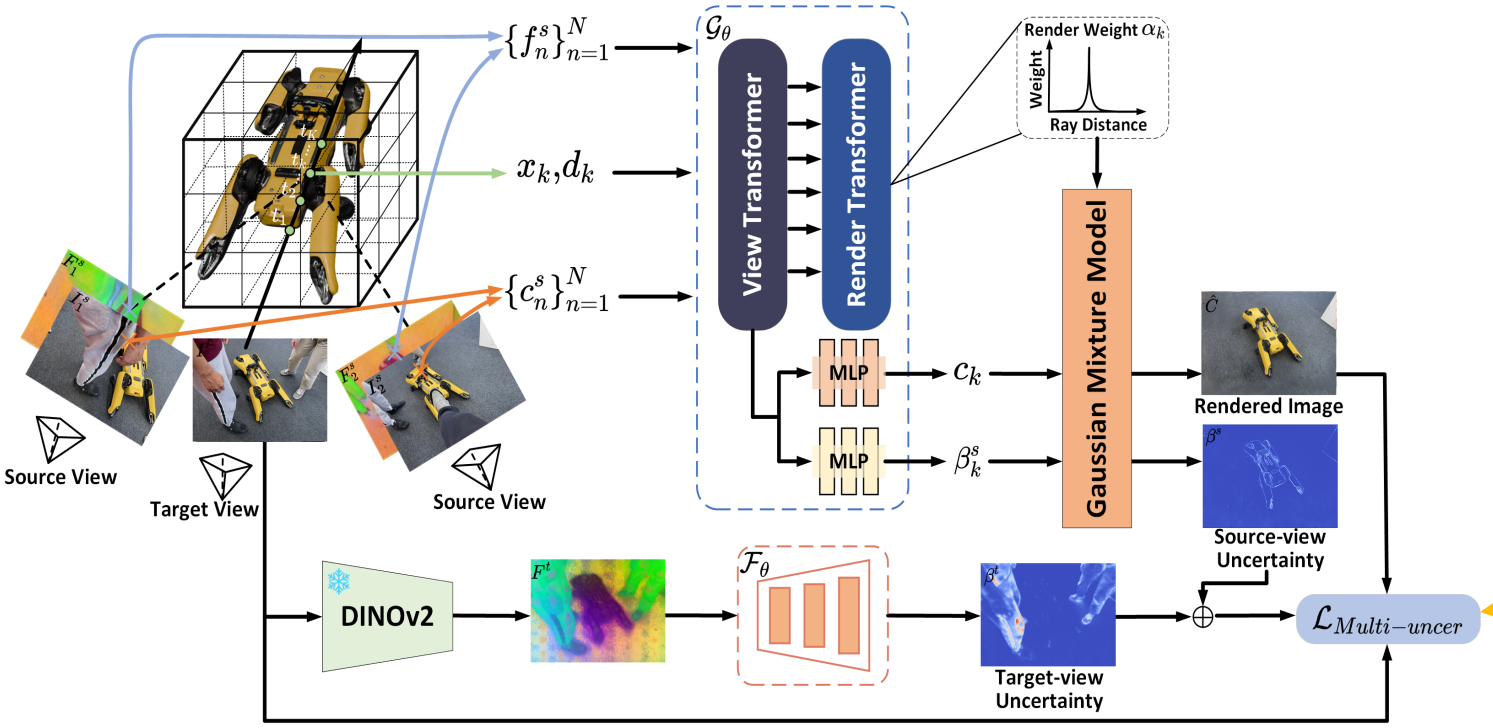


Figure 1. **Proposed Multi-view Uncertainty-guided distractor-aware methods.** Two complementary uncertainties are estimated to support stable training in dynamic scenes. Source-view Uncertainty captures structural inconsistencies across source views arising from occlusion or dynamic factors, while Target-view Uncertainty localizes the spatial distribution of distractors within the target view. Together, they synergistically form a heteroscedastic reconstruction loss that guides the model to adaptively modulate supervision and enhances the robustness of geometric modeling.

- **Source-view Uncertainty**: Captures structural discrepancies across source views (occlusion, dynamics)
- **Target-view Uncertainty**: Localizes transient distractors in the target view
- **Heteroscedastic Reconstruction Loss**: Adaptively modulates supervision, enabling robust geometry learning in dynamic scenes



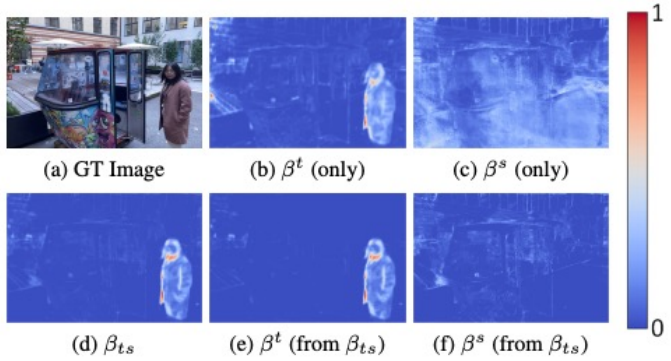


Figure 3. **Visualization results under different uncertainty modeling strategies.** (a) The target image. (b) Modeling only Target-view Uncertainty β^t tends to misjudge static regions as distractors. (c) Modeling only Source-view Uncertainty β^s fails to localize distractors in the target view. (d) Jointly modeling both β^s and β^t and weighted fused into β_{ts} under the multi-view uncertainty framework. (e) and (f) are the separate visualizations of these two components to demonstrate their complementary roles: under synergy, β^t accurately locates transient distractors, while β^s effectively captures static structural discrepancies. Ultimately, β_{ts} integrates the advantages of β^t and β^s, enabling more accurate supervision modulation. The vertical bars show normalized uncertainty (0 to 1), with warmer colors representing higher uncertainty.

$$\mathcal{L}_{\text{Multi-uncer}} = \frac{\mathcal{L}_{\text{SSIM}}(P(r), \hat{P}(r)) + \mathcal{L}_{\text{MSE}}(P(r), \hat{P}(r))}{2\beta_{ts}^2(r)} + \lambda \log \beta_{ts}(r) \tag{8}$$

where $\beta_{ts} = \omega \cdot \beta^t + (1-\omega) \cdot \beta^s$, ω is weight coefficient. $P(r)$ and $\hat{P}(r)$ are patches from the ground truth $C(r)$ and rendered images $\hat{C}(r)$, respectively. The loss comprises both Structural Similarity Index Measure (SSIM) [50] and Mean Squared Error (MSE) terms.

vs. Generalizable Methods (ReTR, MuRF):

- Consistently superior, with or without fine-tuning
- MuRF suffers severely — inter-view modeling misled by distractors

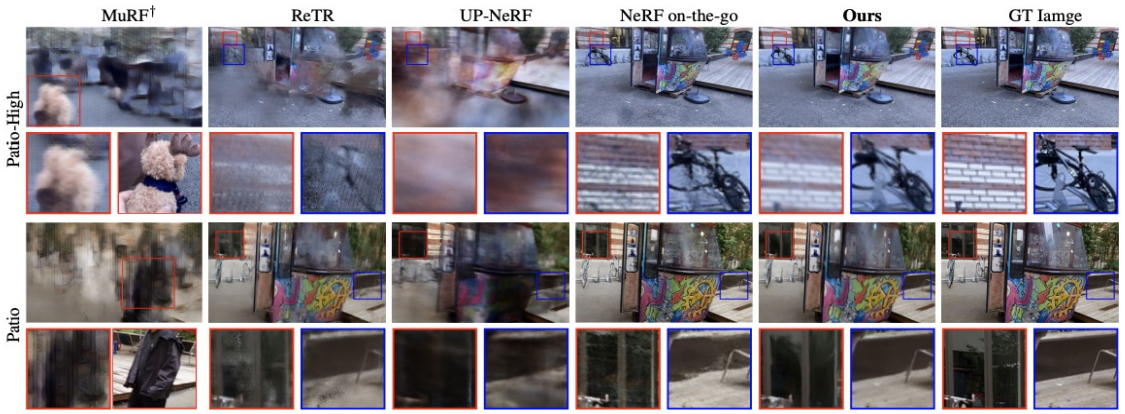


Figure 4. **Qualitative comparison on On-the-go after fine-tuning.** MuRF† implicitly models inter-view consistency, making it susceptible to transient distractors and causing its rendered results to exhibit distractor artifacts.

Method	Setting	Corner			Patio			Spot			Patio-High		
		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
ReTR (NIPS 2023)	No optimized per-scene	16.33	0.545	0.541	16.39	0.418	0.568	17.43	0.475	0.502	15.66	0.327	0.593
MuRF (CVPR 2024)		13.41	0.348	0.692	11.78	0.266	0.647	14.18	0.342	0.657	11.88	0.234	0.707
MU-GeNeRF (Ours)		**17.96**	**0.597**	**0.502**	**18.63**	**0.543**	**0.454**	**19.35**	**0.510**	**0.469**	**17.76**	**0.446**	**0.507**
ReTR ft (NIPS 2023)	Optimized per-scene	19.76	0.611	0.439	17.97	0.485	0.494	18.52	0.495	0.483	16.88	0.381	0.566
MuRF ft (CVPR 2024)		13.57	0.356	0.685	12.26	0.290	0.630	15.02	0.382	0.627	12.01	0.276	0.693
MuRF† ft (CVPR 2024)		14.03	0.363	0.677	13.22	0.286	0.616	15.15	0.384	0.621	12.42	0.280	0.688
UP-NeRF (NIPS 2023)		19.34	0.619	0.443	15.78	0.491	0.624	16.71	0.398	0.606	14.52	0.379	0.723
NeRF on-the-go (CVPR 2024)		**23.15**	**0.751**	**0.252**	**21.35**	**0.717**	**0.276**	**23.03**	**0.727**	**0.246**	**20.99**	**0.687**	**0.263**
MU-GeNeRF (Ours) ft		21.77	0.669	0.338	20.72	0.644	0.290	21.38	0.553	0.424	20.33	0.564	0.311

Table 1. **Quantitative comparison on On-the-go.** ft denotes per-scene fine-tuning. † indicates that the pretrained weights released by the original work are used; methods without this symbol are trained from scratch. The underline indicates that our method is only inferior to NeRF on-the-go.

vs. Distractor-free NeRFs (UP-NeRF, NeRF On-the-go):

- Outperforms UP-NeRF by clear margin
- Comparable to NeRF On-the-go, but: per-scene optimization, ~48 hrs; feed-forward GeNeRF, ~2 hrs

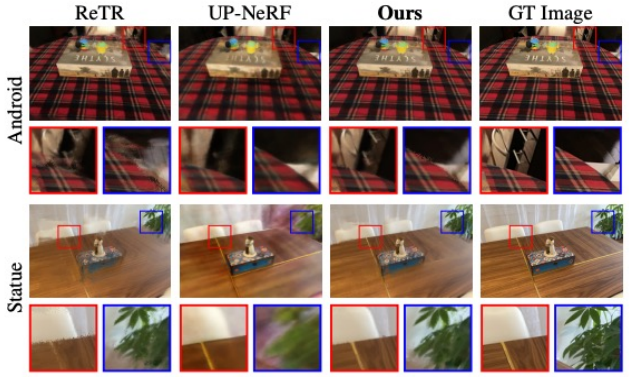


Figure 5. **Qualitative comparison on RobustNeRF after fine-tuning.**

Method	Statue		Android	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑
ReTR (NIPS 2023)	16.84	0.500	18.63	0.531
MuRF (CVPR 2024)	12.88	0.337	11.94	0.370
MU-GeNeRF (Ours)	**18.28**	**0.577**	**19.87**	**0.583**
ReTR ft (NIPS 2023)	18.77	0.585	20.29	0.601
MuRF ft (CVPR 2024)	13.10	0.351	12.34	0.389
MuRF† ft (CVPR 2024)	13.51	0.366	12.93	0.401
UP-NeRF (NIPS 2023)	18.10	0.629	20.45	0.664
NeRF on-the-go (CVPR 2024)	**21.25**	**0.732**	**23.17**	**0.756**
MU-GeNeRF (Ours) ft	19.97	0.651	22.34	0.701

Table 2. **Quantitative comparison on RobustNeRF.**

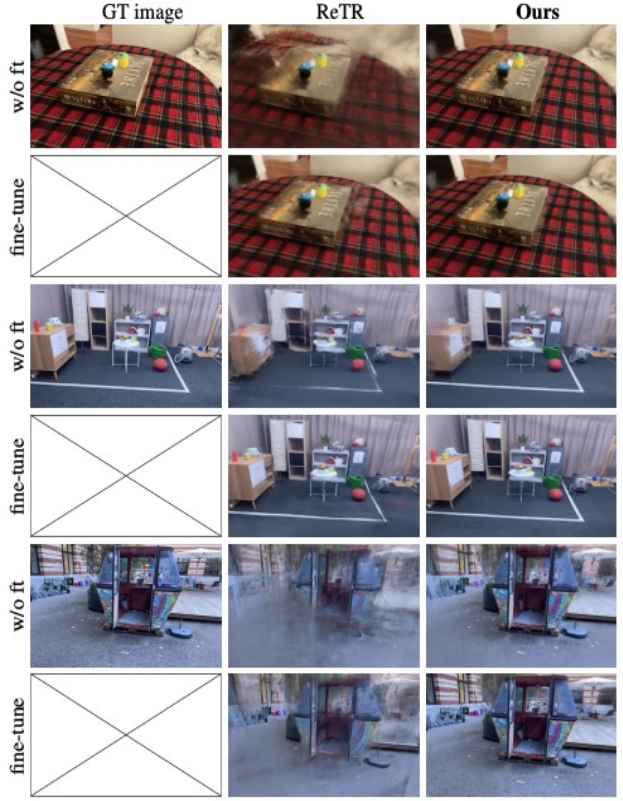


Figure 12. **Comparison of rendering results before and after fine-tuning.**

Ablation Study: Each Component Matters

- Removing any uncertainty: Performance drops significantly
- Single uncertainty (either β^s or β^t): Insufficient alone
- MSE or SSIM alone: Fails to fully constrain optimization
- Conclusion: Both uncertainties + both losses are essential

	β^s	β^t	$\mathcal{L}_{MSE}$	$\mathcal{L}_{SSIM}$	Corner PSNR↑	Corner SSIM↑	Patio-High PSNR↑	Patio-High SSIM↑
0			✓	✓	20.20	0.625	14.92	0.332
1	✓		✓	✓	19.85	0.618	15.33	0.341
2		✓	✓	✓	20.73	0.640	19.55	0.496
3	✓	✓	✓		16.84	0.563	13.07	0.256
4	✓	✓		✓	14.87	0.497	14.10	0.292
Full	✓	✓	✓	✓	**21.77**	**0.669**	**20.33**	**0.564**

Table 3. **Quantitative comparison of different ablation components.**

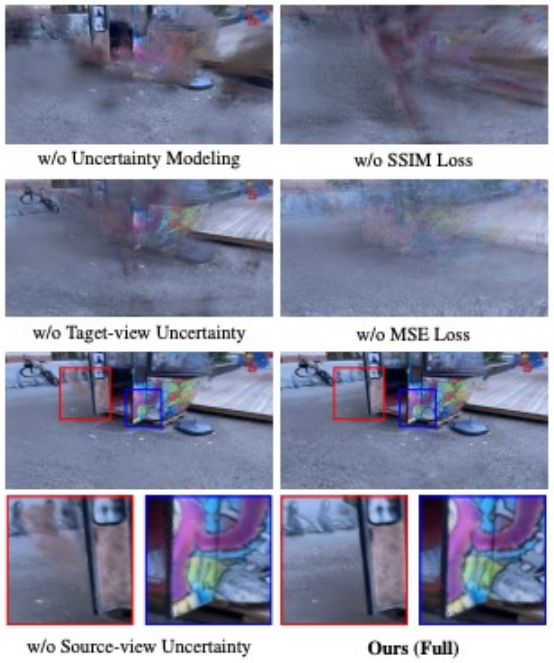


Figure 6. **Qualitative comparison of view rendering results under various ablation components.**

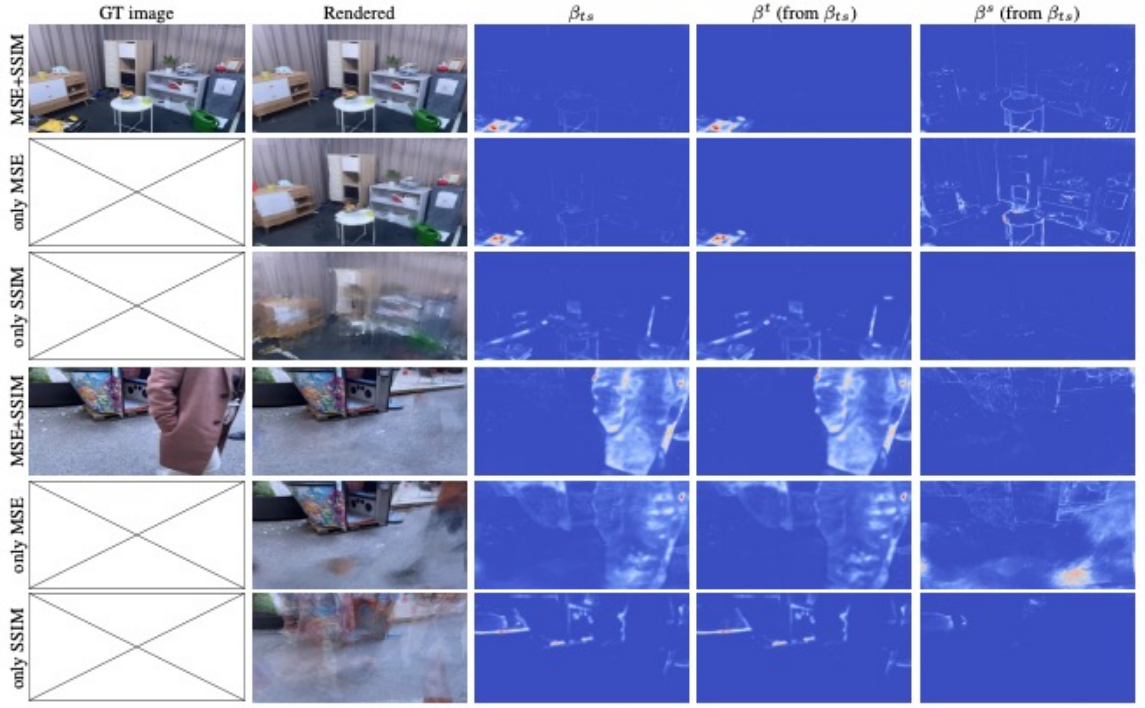


Figure 10. **Comparison of loss term ablation experiments,** showing the rendering results and uncertainty visualizations using MSE loss only, SSIM loss only, and the combination of both losses.

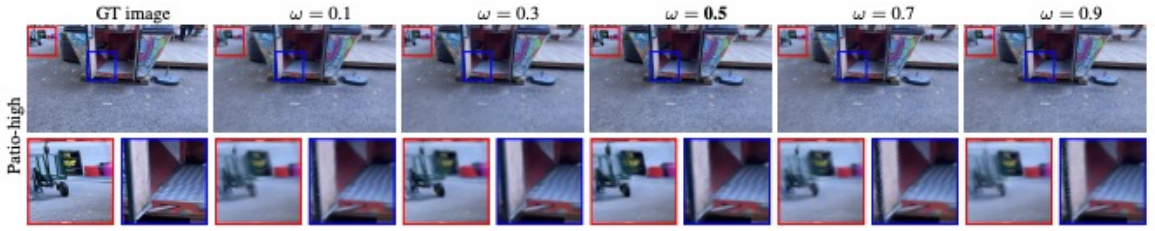


Figure 11. **Qualitative Comparison under Different Weight Settings.**

Uncertainty Visualization: How β^s and β^t Complement Each Other

- β^t alone: Localizes distractors but misjudges inconsistent static structures → detail loss
- β^s alone: Captures cross-view conflicts but blind to target-view distractors → residual artifacts
- β_{ts} (Ours): Combines both strengths, driven by geometric consistency, not semantic memorization

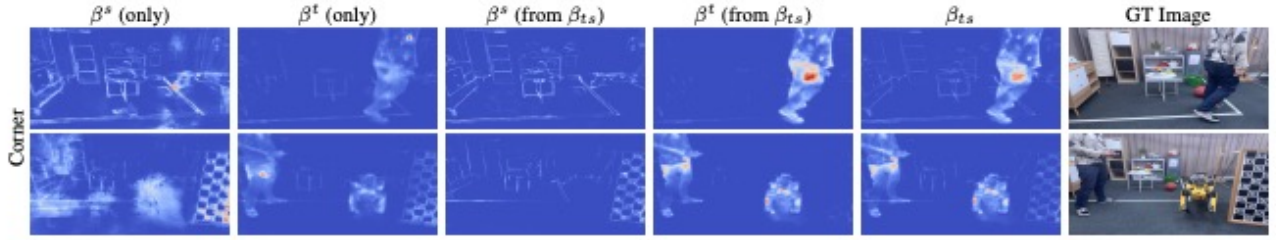


Figure 8. **Visualization Comparison of Different Uncertainty Modeling Strategies.**

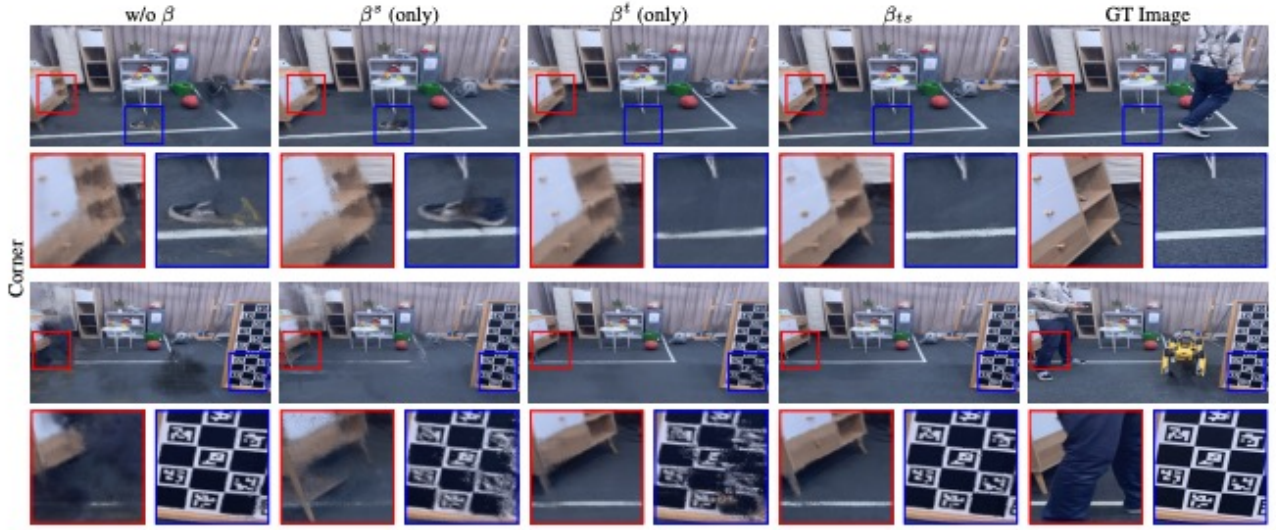


Figure 9. **Distractor Suppression Performance under Different Uncertainty Modeling Strategies.** w/o β denotes that no uncertainty is modeled.

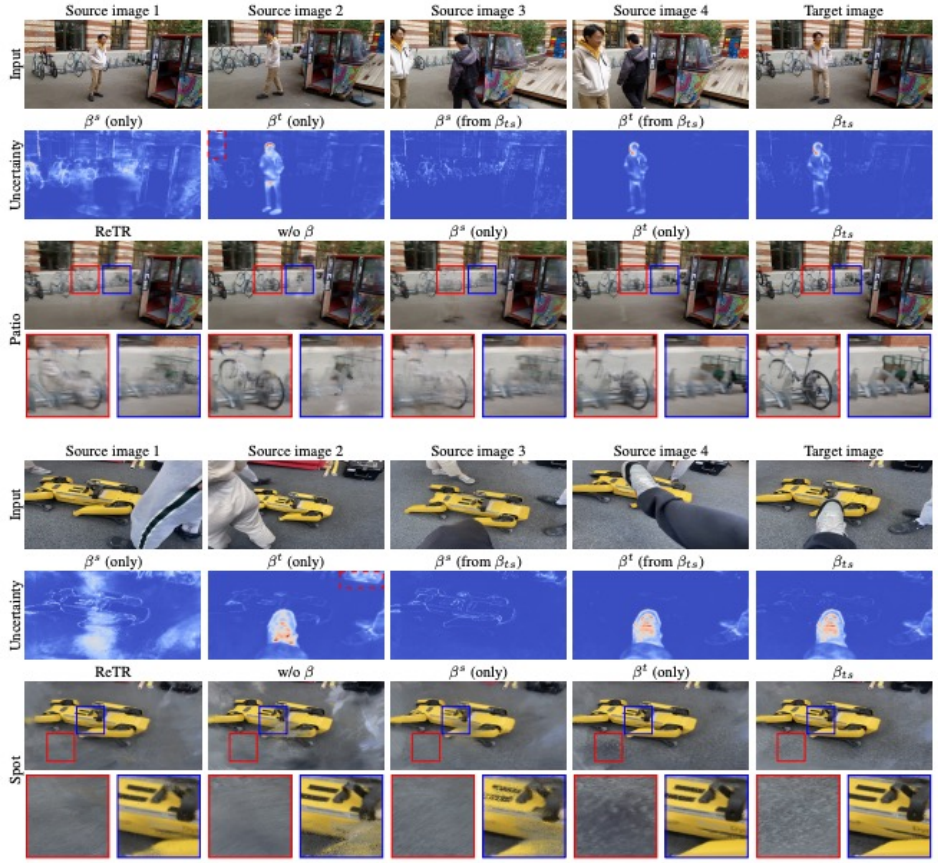


Figure 16. **Qualitative comparison of rendering results in distractor-containing scenes.**

Thanks for listening!

Code Link

Group’s Homepage