

WELL-NeRF: ENSURING WELL-POSED NEURAL RADIANCE FIELDS VIA VIEW FRUSTUM AND SHADOW ZONE BASED REGULARIZATION

Anonymous authors

Paper under double-blind review

Supplementary Materials for Rebuttal

CONTENTS

1	Near-Far Parameter Robustness	2
2	Effectiveness of Dynamic Lambda	3
3	Visualization of Frustum score	4
4	Dataset Proposal: Randomized Structures and Patterns	5
5	Video Demonstration of NeRF Rendering and Loss Curve	9

1 NEAR-FAR PARAMETER ROBUSTNESS

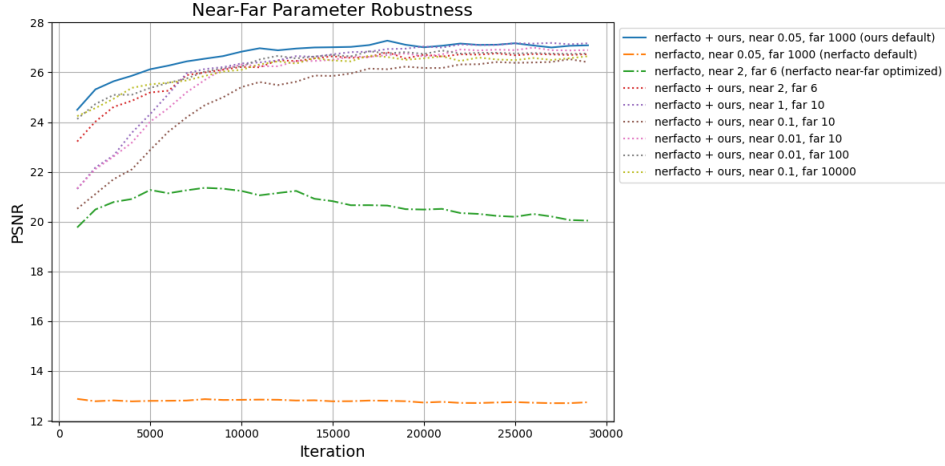


Figure 1: PSNR vs. Iterations Under Various Near-Far Parameter Settings (Evaluation Metric)

Our model is designed to be near-far parameter-free. As shown in Figure 1, our model demonstrates robustness to near-far parameter settings. This is achieved by incorporating spatial boundary conditions into the learning process through Frustum Score Regularization. In contrast, Nerfacto fails to learn effectively under broad near-far conditions and exhibits suboptimal performance even under optimized near-far settings.

2 EFFECTIVENESS OF DYNAMIC LAMBDA

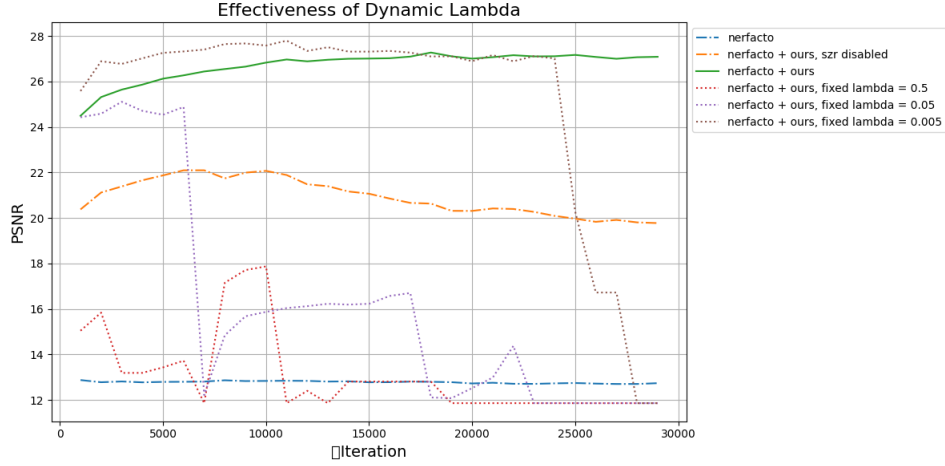


Figure 2: Contribution of Dynamic λ to Performance and Model Stability

To prevent over-regularization, we employed a dynamic adjustment of λ . The RGB loss is duplicated and reused as the Shadow Zone Regularization loss. This approach ensures that the model functions identically to the base model during the mid-to-late stages of training, minimizing the risk of over-regularization or training collapse.

Regularization methods designed based on human intuition are prone to counterexamples and carry a significant risk of over-regularization. In contrast, our method mitigates these risks, preserving the original potential of the model to the fullest extent. We consider this an additional effective contribution of our approach.

3 VISUALIZATION OF FRUSTUM SCORE

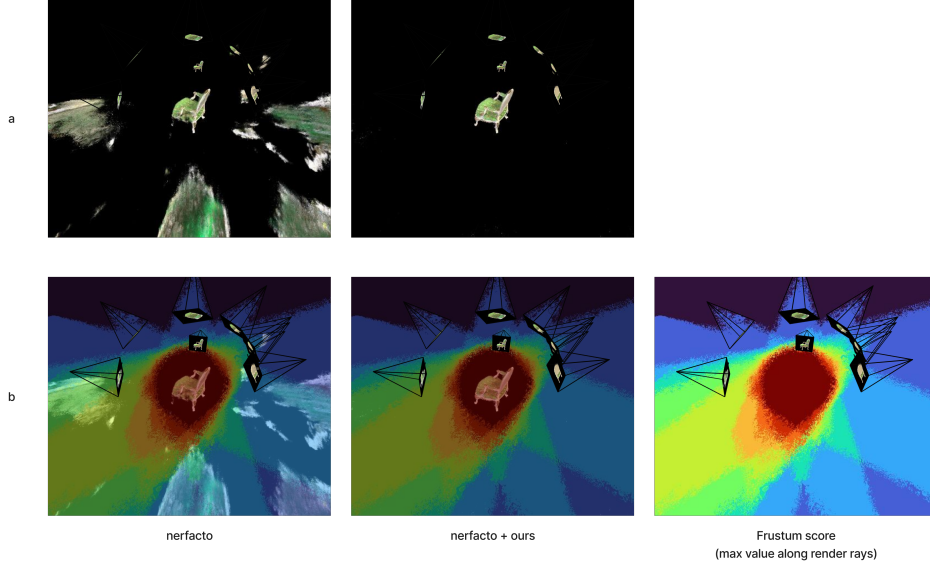


Figure 3: Visualization of Frustum Scores Overlaid on Rendering Results

The view Frustum Score was visualized and overlaid onto the rendering. Misformed fragments of objects in regions where position inference is impossible can be observed. However, since the training loss converges without issues, the rendered results appear normal from the input view positions. We believe the root cause of this phenomenon lies in improperly defined spatial boundary conditions during training.

When solving differential equations with incorrectly specified boundary conditions, incorrect results do not imply a flaw or limitation in the equations themselves. Similarly, if there are no inherent flaws in the general NeRF design, we believe that its potential can be further unlocked with more rational configurations, including address

4 DATASET PROPOSAL: RANDOMIZED STRUCTURES AND PATTERNS

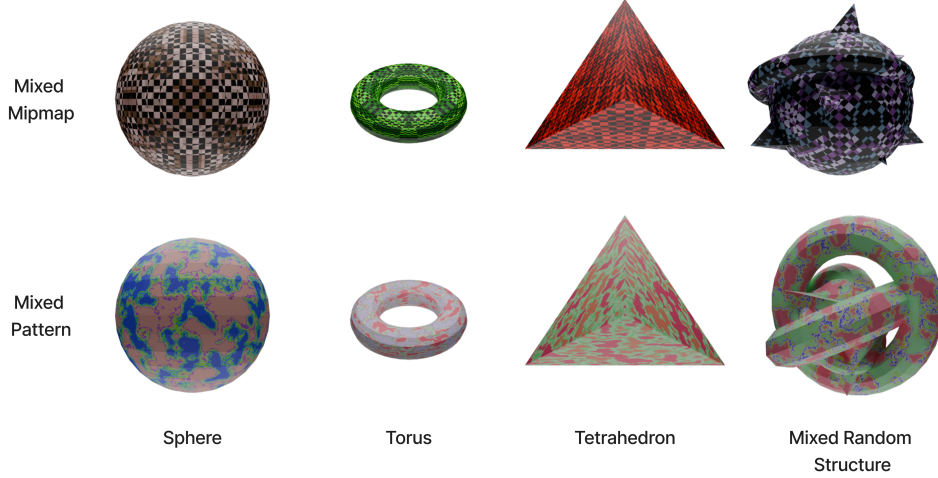


Figure 4: Examples of Base Structures and Patterns.

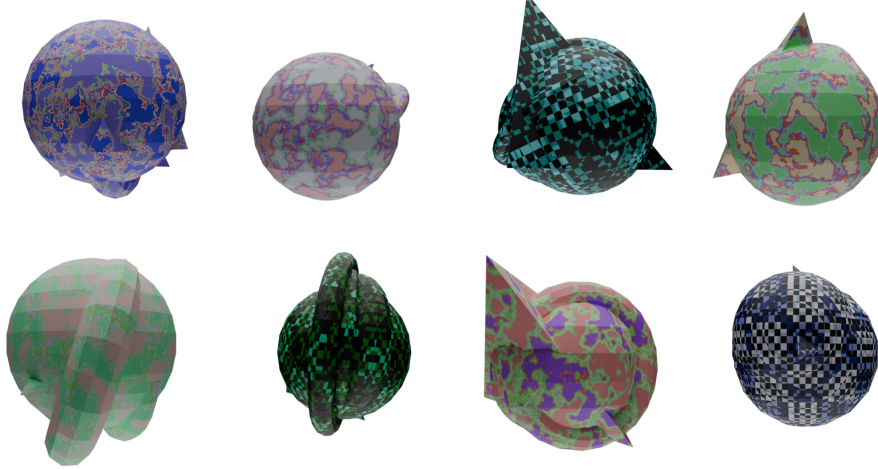


Figure 5: Examples of Randomly Generated Structures and Patterns.

We propose a new dataset generator for evaluating the performance of NeRF. Existing datasets are often confined to a very limited number of classes, which negatively impacts researchers’ ability to objectively assess their methods, especially in the context of the sparse input problem.

In pursuit of better performance metrics, researchers may unintentionally design overly regularized methods biased toward specific datasets, potentially degrading

the model’s generalizability. Additionally, it is often observed that different parameters are set for each dataset to improve average performance.

To address this issue, we have designed a random structure and pattern-based dataset generator that allows for a clearer evaluation of a model’s generalization performance. The final structures are composed by randomly generating and combining base structures and patterns.

Base Structures: Sphere, torus, tetrahedron, etc. (two basic topologies and structures with edges)

- **Base Structures:** Sphere, Torus, Tetrahedron, etc. (two basic topologies and structures with edges)
- **Base Patterns:** Noise, Mip-map (checkerboard pattern)
- **Camera Positions:**
 - **Training Cameras:** Centers of the faces of a cube enclosing the object (total of 6)
 - **Evaluation Cameras:** Centers of the edges and vertices of the cube (total of 20)

Since the input data are far apart at 90-degree angles from each other, this is quite a challenging dataset (more sparse compared to datasets concentrated in a hemisphere or a quarter sphere). Even in such circumstances, our model consistently succeeds in converging the object within a normal range.

The code for this generator will be publicly released, and it is designed to allow users to generate datasets by specifying the desired level of randomness and scale, as well as to easily add custom patterns and structures. In the future, we plan to update the generator to automatically produce more reasonable and diverse random datasets by advancing mathematical definitions, topology, patterns, geometry, and so on, regarding patterns and geometric structures.

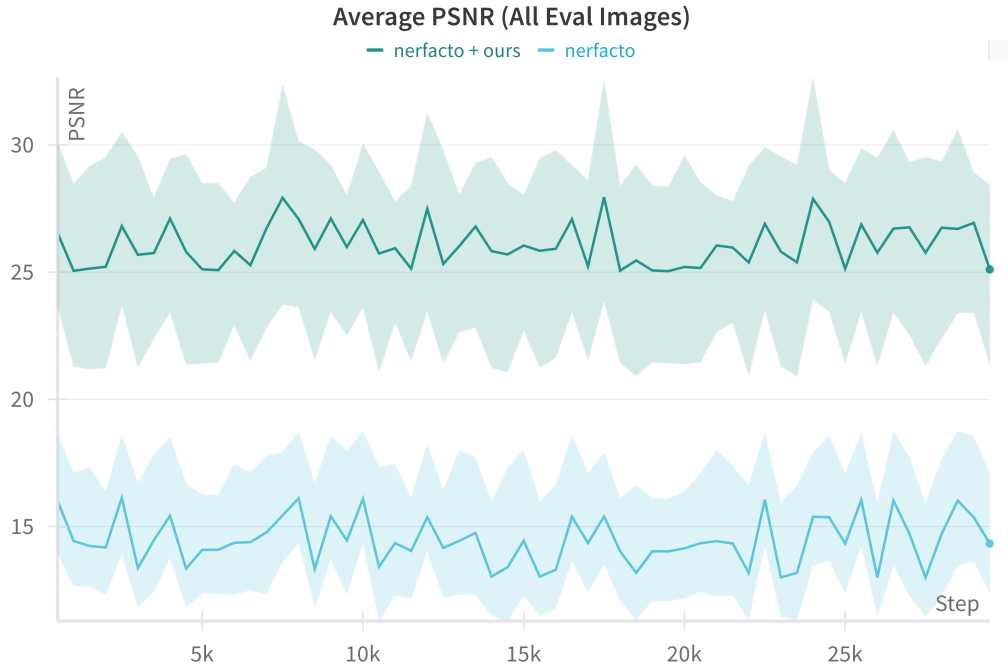


Figure 6: Average PSNR results for 16 random structure training runs.

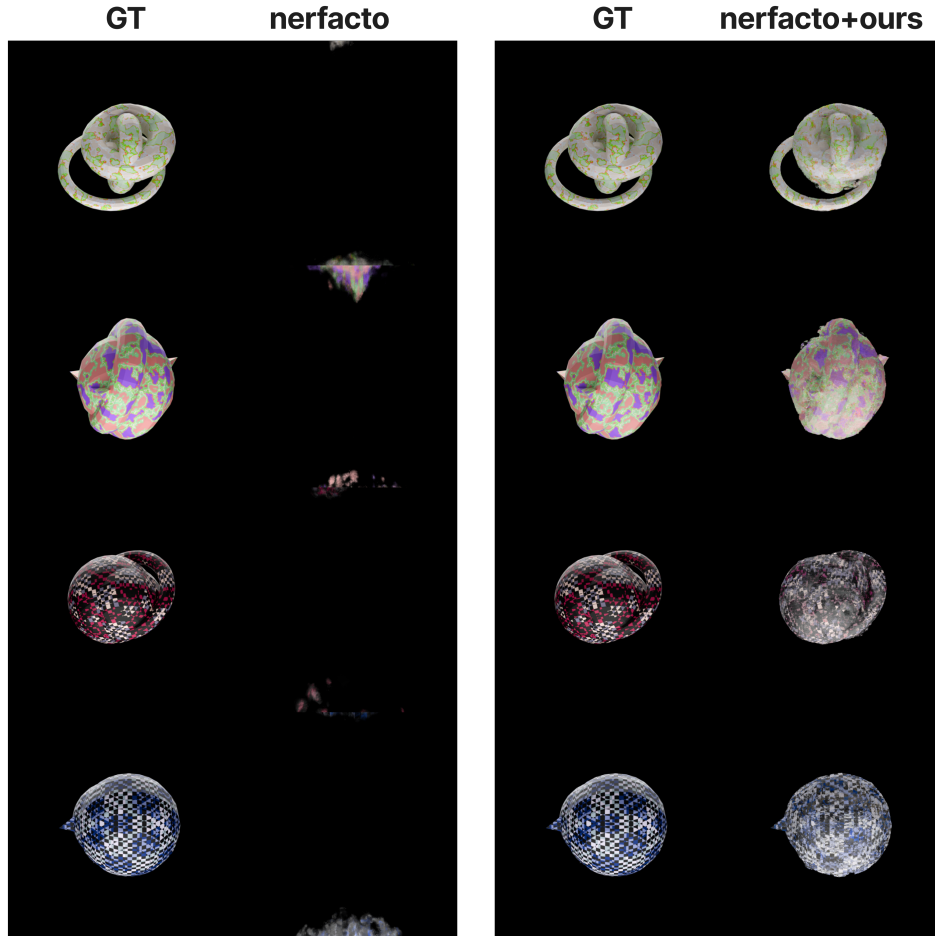


Figure 7: Rendering results from 4 samples out of 16 training runs.

5 VIDEO DEMONSTRATION OF NeRF RENDERING AND LOSS CURVE

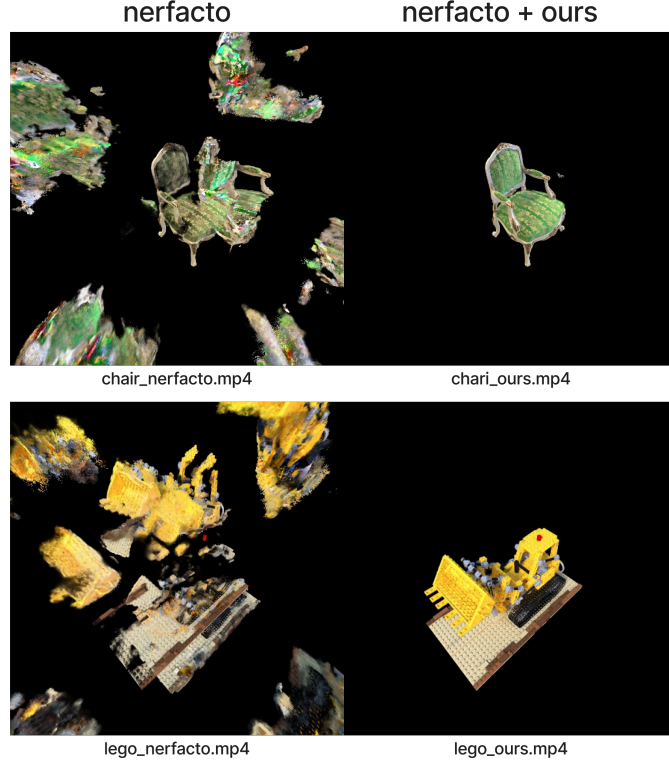


Figure 8: Thumbnails of Rendering Video

Through the video materials, the fragments described in the Visualization of Frustum Score section can be observed in greater detail.

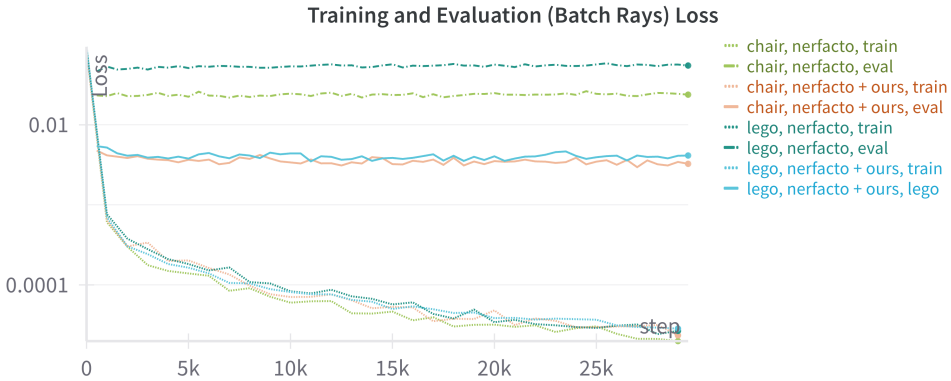


Figure 9: Loss results from experiments included in the video materials.

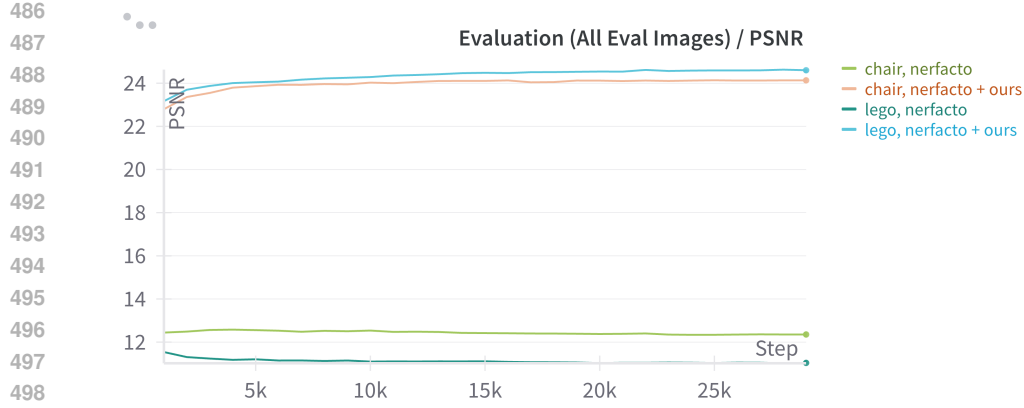


Figure 10: PSNR results from experiments included in the video materials.

The figure shows the RGB Loss and PSNR (Batch Rays) curves corresponding to the video experiment. Our model achieves rapid initial convergence, which can make the curve appear nearly horizontal at a broader scale.

When applying our method, there is a significant performance improvement (in evaluation metrics) compared to Nerfacto, while the training loss curves remain nearly identical. This indirectly demonstrates that our method minimally disrupts the learning mechanism of the base model.