Optimizing NeRF: An Ablation Study of Nerfacto on Training Efficiency and Quality

Advanced Computer Graphics

Zijie Cai
University of Maryland, College Park

September 7, 2025

Implementation available on Google Colab:

Abstract

Neural Radiance Fields (NeRF) enable high-quality 3D scene reconstruction by representing the scene using a deep neural network. However, this method is computationally intensive and highly dependent on the quality and quantity of input data. In this project, I performed ablation studies with the goal to optimize the default 'nerfacto' model implementation from the nerfstudio framework. Specifically, the following model elements are being explored:

- Hyperparameter tuning: hidden dimensions, learning rates, batch sizes.
- Initial sampling strategies: Uniform Sampler vs Piecewise Sampler.
- Various number of input images and scene resolution.

The optimized model configuration demonstrates improved performance while maintaining the model efficiency for training. This provides a deeper understanding of trade-offs between computational efficiency and reconstruction quality.

1 Introduction

While NeRF is widely used nowadays for high-quality 3D scene reconstruction, it faces many challenges in model training efficiency as it often requires a lot of input images to obtain good results, and sensitivity in input data as a sparse camera set of input images could lead to motion blur which is not explicitly handled by the original NeRF model.

The goal of this project is to optimize the default 'nerfacto' model, an open-source implementation of an optimized version of the original NeRF, 'vanilla-NeRF' model. The reason for this is due to the extensive training time of the 'vanilla-NeRF' model, where training can take up to 2 days. In comparison, the default 'nerfacto' model only need 30 minutes of training for the same scene with only 6 GB memory usage. The implementation is based on the Google Colab notebook, which connects to a free-tier T4 GPU at run time.

Specifically, this report documents the following:

- The impact of hyperparameter tuning (hidden dimensions, learning rates, batch sizes) on training performance and quality.
- Comparison between the sampler choice (Uniform Sampler vs Piecewise Sampler).
- The effect of input image quantity and scene resolution on reconstruction quality.

The project provides visualizations of scene reconstruction and plots of model training metrics comparisons with different model configurations. Based on both visual and numerical analysis of model training performance, the optimized configurations for the default 'nerfacto' model configurations based on nerfstudio framework is also provided.

2 Related Work

The original Neural Radiance fields (NeRF) was introduced by Mildenhall, Srinivasan, Tancik et al. (2020). At the time, it was a novel approach to 3D scene reconstruction utilizing deep neural networks for scene representation instead of traditional computational imaging methods or relying on additional sensors.

The implementation of this project is based on the nerfstudio framework, which is an open-source research project launched by a group of students at Berkeley research lab. The framework provided a user-friendly interface for monitoring model training of many popular existing 3D reconstruction models based on both NeRF and Gaussian Splatting. The performance results included in this report are collected through streamlined tools : viewer for scene reconstruction and wandb for numerical results.

The Google Colab notebook provides the entire pipeline, from environment setup to model training and rendering output, to replicate all the results documented in this report. Through ablation studies, the project builds upon previous studies with optimizations.

3 Approach and Methodology

3.1 Datasets

Three nerfstudio datasets were used for model training evaluation:

- **Vegetation**: 463 images, resolution 720x960.
- **Floating Tree**: 130 images, resolution 3008x2000.
- **Giannini Hall**: 458 images, resolution 3008x2000.

These three datasets include a wide range of diverse characteristics for evaluating model training with various input image quantity and quality.

3.2 Ablation Studies

In this project, the ablation studies are performed through three stages optimizing model configurations and analyzing model training efficiency to help identify both advantages and disadvantages. Here are more details focusing on each study:

- **Hyperparameter Tuning:** Here are the three model hyperparameters tuned and evaluated for this project:
  - **Hidden Layer Dimensions:** Evaluated hidden layers size options of 64 (default) and 128 neurons in the hidden layers for various model complexity.
  - **Learning Rates:** Evaluated learning rate options of 0.005, 0.01 (default) and 0.015 for training convergence speed.
  - **Batch Sizes:** Evaluated batch size options of 1024, 2048 and 4096 (default) for training efficiency and reconstruction quality.

  The default 'nerfacto' model has hyperparameters values of the following: hidden layer dimension = 64, learning rate = 0.01, batch size = 4096. For this project, this default 'nerfacto' model configurations with three hyperparamters values were used as a baseline for evaluation comparison. The outcome of this section will provide a set of fine-tuned hyperparameters values of hidden layer dimension, learning rate, and batch size, which achieves better training performance and efficiency than the baseline by nerfstudio framework for selected datasets.

- **Initial Sampling Strategy:** By default, 'nerfacto' model utilizes an initial piecewise sampler for selecting rays for training. For this section, model training was performed using both piecewise sampler and uniform sampler separately and results were compared and evaluated based on the Vegetation dataset with the optimized set of hyperparameters from the previous section.

- **Input Images and Scene Resolutions:**
  - **Number of Input Images:** The Floating Tree (130 images) and Giannini Hall (458 images) datasets were used for evaluating the effects of various number of input images for model training since both datasets have the same scene resolution to ensure consistency.
  - **Scene Resolutions:** The Vegetation (720x960) and the Giannini Hall (3008x2000) were used for evaluating the trade-offs between model training efficiency and reconstruction quality with varying scene resolutions since both datasets almost have the same number of input images: The Vegetation (463 images) and Giannini Hall (458 images).

Model training commands for all ablation studies with different model configurations were generated for executing in terminal and reproducing results in this report.

4 Experiments and Results

4.1 Hyperparameter Tuning

- **Optimal Configuration (Purple line in Figure 2):** Hidden layer dimensions = 64, learning rate = 0.015, batch size = 4096. The only difference from the default 'nerfacto' model hyperparameters is the learning rate is a bit higher than the default (0.015 vs. 0.01). This yields not only faster convergence for training based on the number of rays trained per second and training time per iteration, and better reconstruction results based on evaluation loss.

- **Observations:** There are many interesting observations after interpreting the training metrics results for 5000 steps on the Vegetation dataset.

  Based on the right plot in Figure 1, the hyperparameter batch size is associated with the initial GPU memory usage directly. In other words, the larger the batch size is, the higher initial GPU memory usage is. For a batch size of 4096, the initial GPU memory usage is the red line which is above the blue line for a batch size of 2048. Based on the left plot in Figure 1, the batch size is also related to training time, where a larger batch size leads to longer training time.

  Additionally, based on the center plot in Figure 1 and the left plot in Figure 2, the optimized model configurations (purple line) achieve the best evaluation loss among all evalu2ated configurations, where the red line is the baseline 'nerfacto' model.

  Another finding based on the center plot and the right plot in Figure 2 is that the optimized model configurations seem to have consistent model training efficiency (purple line), where the default baseline (red line) had one downward spike between 2000 and 3000 steps, and model with higher hidden layer dimension of 128 (blue line)'s training efficiency downgraded significantly after 3000 steps.

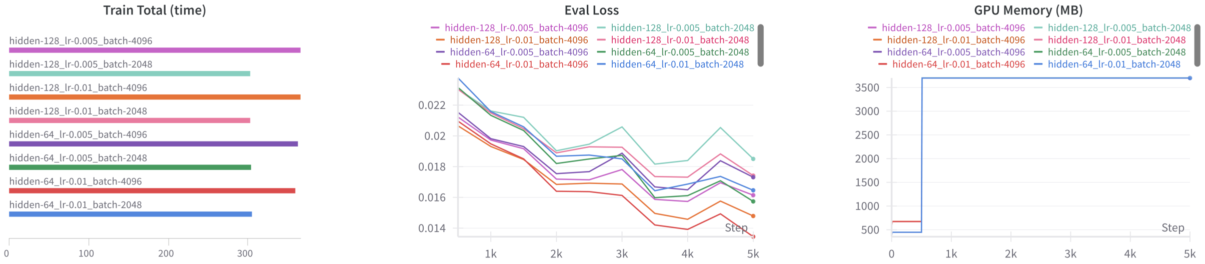

Figure 1: Training results for 5000 steps with hidden layer dimensions (64 vs. 128), learing rate (0.005 vs. 0.01), and batch size (2048 vs. 4096): Total training time (secs) (left), Evaluation loss (center), and GPU memory usage (right).

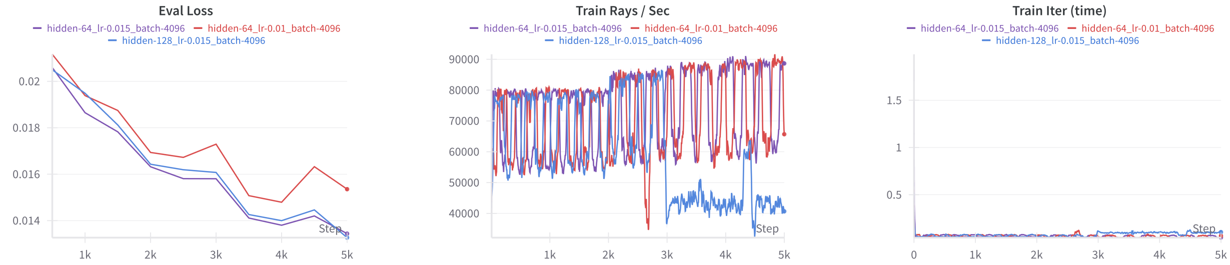

Figure 2: Training results for 5000 steps with hidden layer dimensions (64 vs. 128), learning rate (0.01 vs. 0.015), and batch size (4096): Evaluation loss (secs) (left), Training Efficiency (center), and Training time per step (right).

4.2 Sampling Strategies

- **Piecewise Sampling:** Based on evaluated results in Figure 3, while using an initial piecewise sampler leads to a slightly longer training time, it converges much faster than uniform sampling as demonstrated in the evaluation loss plot on the right in Figure 3.

- **Uniform Sampling:** Although the training loss seems to converge using an initial uniform sampler in the pipeline after 5000 steps and the training time is slightly faster, the model with a uniform sampler does not seem to converge after 5000 steps on the same scene input and same model configurations based on the evaluation loss plot in Figure 3, where the uniform sampler (blue line) does not show a decreasing trend, but the piecewise sampler (red line) shows convergence.

  Figure 4 shows the visual reconstruction comparison after the model was trained for 5000 steps. On the left is the result with a uniform sampler where there are many artifacts in the center of the final reconstruction. On the right is the result with a piecewise sampler, where there are no such artifacts. This is likely caused by sparse camera input views, where the NeRF model using a uniform sampling strategy handles all regions equally likely, but the model using a piecewise sampling strategy focuses on regions that contribute to the reconstruction quality the most which makes the piecewise sampler more robust for sparse input data. In other words, NeRF model training with a piecewise sampler leads to faster convergence speed for reconstruction.

4.3 Number of Input Images

- **Results:** For this section, the ablation study of NeRF model training with various number of input images is evaluated on the Floating tree and Giannini Hall dataset at the same scene resolution. In Figure 5, on the left is the reconstruction result of Floating Tree scene with 130 input images, and on the right is the reconstruction result of Giannini Hall scene with 458 input images. Based on visual interpretation, the reconstruction results of a model trained with more images seem to produce an overall more consistent reconstruction result, whereas the reconstruction results of a model trained with fewer images will lose a bit more fidelity from the original data.

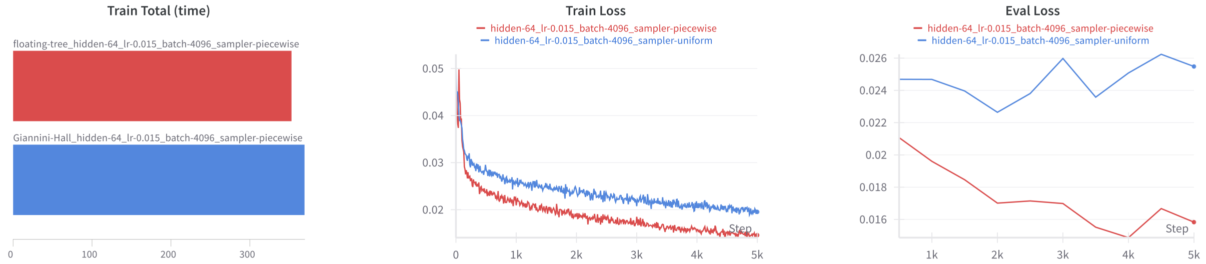

Figure 3: Comparison of initial sampling strategies [Uniform (red) vs. Piecewise (blue)]: Total train time (left), Training loss (center), and Evaluation loss (right).

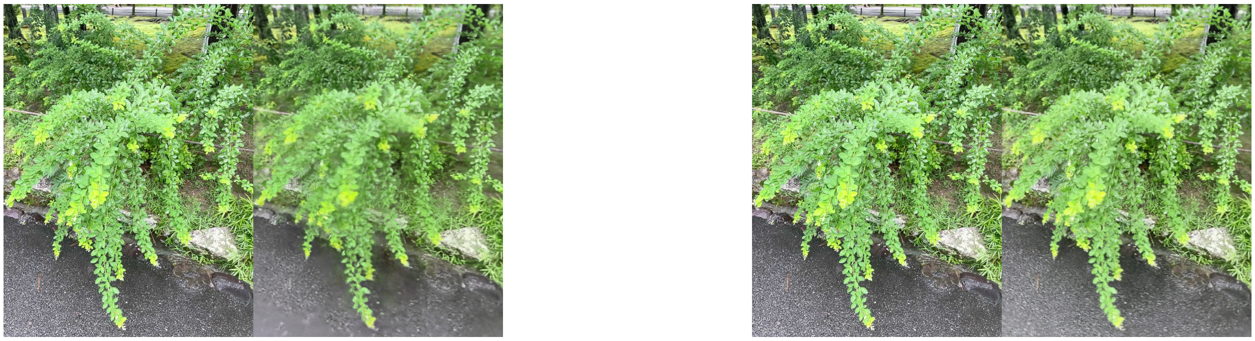

Figure 4: Comparison of sampling strategies: Uniform (left) vs. Piecewise (right).

4.4 Scene Resolution

- **Datasets:** For this section, model training performance is evaluated with varying input scene resolution on the Vegetation dataset at original resolution of 720x960 with downscale factor = 1, 2, 4, 8 and the Giannini Hall dataset at original resolution of 3008x2000 with downscale factor = 4, 8.

4.4.1 Vegetation Dataset

- **Results:** Based on Figure 6, higher scene resolution yields to longer training time and GPU memory usage. It is notable that the relationship is not linear. In other other words, other than training the model at original resolution 720x960 with downscale factor = 1 which is significantly more computationally intensive as other three downscale factors use relatively similar amount of resources without spikes during training.

4.4.2 Giannini Hall Dataset

- **Results:** Similarly, based on Figure 7, while the model training time do not differ too much for model training on the Giannini hall dataset (3008x2000) at downscale resolution factor of 4 and 8. The GPU memory usages is significantly less for training with lower scene resolution. In fact, based on visual interpretation in Figure 8, while

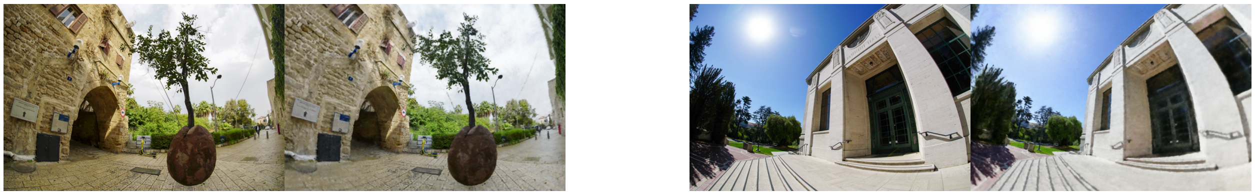

Figure 5: Various number of input images: Floating Tree dataset with 130 images (left) and Giannini Hall dataset with 458 images (right).

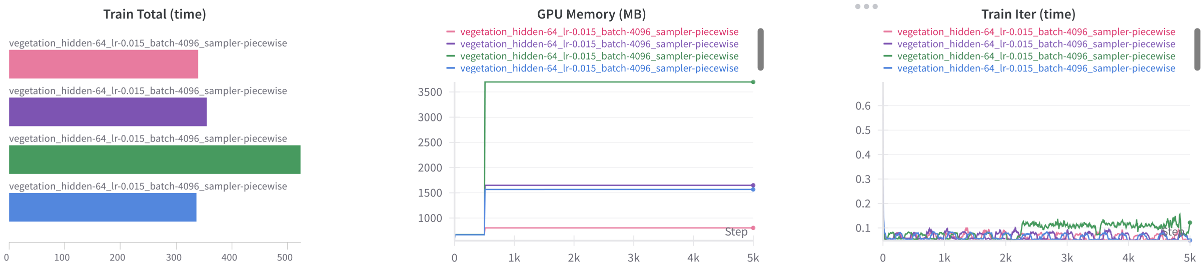

Figure 6: Performance metrics on Vegetation dataset at varying scene resolutions: Training time (left), GPU memory usage (center), and training efficiency (right).

higher input scene resolution leads to sharper reconstruction, it also introduces artifacts, which are smoothed out in the reconstruction with lower input scene resolution.

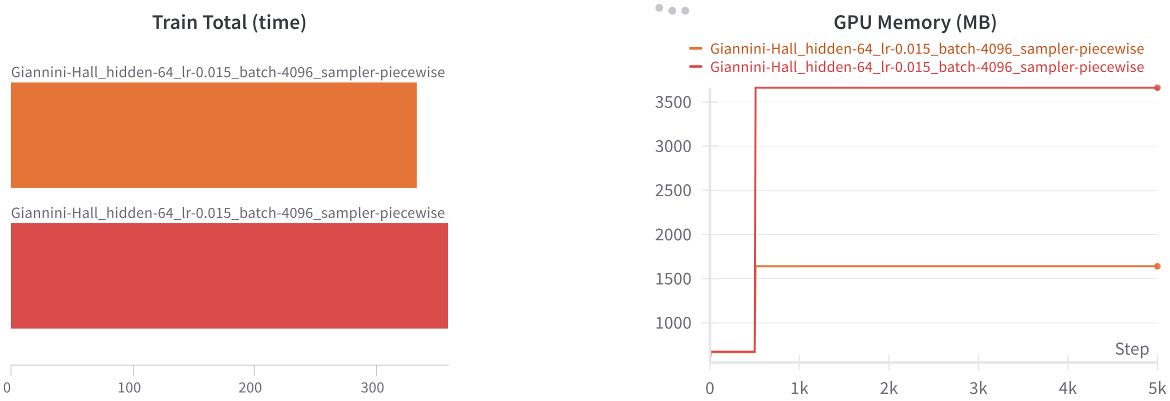

Figure 7: Performance metrics for Giannini Hall dataset at varying scene resolutions: Training time (left) and GPU memory usage (right)

5 Conclusions

This project provides optimization of the default 'nerfacto' model through three ablation studies: hyperparamaters tuning, initial sampling strategy, input data quantity, and varying input scene resolutions. Here is a recap of key findings, limitations, and future directions:

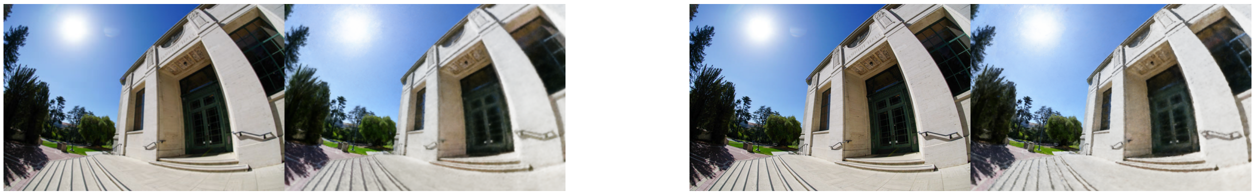

Figure 8: Visual results for Giannini Hall (3008x2000) dataset at varying scene resolutions: Downscale factor 4 (left) and downscale factor 8 (right).

Key Findings

- Hyperparameter Tuning: After fine-tuning the three model training hyperparamters: hidden layer dimensions, learning rate, and batch size, the optimal set of configuration was found to be: hidden layer dimension = 64, learning rate = 0.015, and batch size = 4096. This configuration shows improved and more consistent training efficiency and better reconstruction quality for 5000 steps on the Vegetation dataset compared to the baseline default 'nerfacto' model configuration: hidden layer dimension = 64, learning rate = 0.01, and batch size = 4096.

- Initial Sampling Strategies: Using Piecewise Sampler as initial sampling strategy for model training proves faster convergence speed due to its robustness against sparse input data by prioritizing regions that contribute the most to final reconstruction quality which yield to quicker training and better quality compared to Uniform Sampler.

- Varying Number of Input Images and Scene resolutions: The ablation study shows that NeRF model training with more input images yields better reconstruction quality with better data fidelity compared to training with less input images.

  For varying scene resolutions, study shows that model training with higher scene resolutions leaded to sharper reconstructions with slightly more computational resources. However, the trade-offs for sharper reconstruction is the introduction of artifacts in reconstruction and loss of data fidelity while model training with lower scene resolution smoothens out these artifacts.

Limitations

- Computational Limitations: Ablation studies on larger input image datasets and higher scene resolutions for model training are not included due to the intensive nature of training. A better GPU connection is required for more extensive training.

- Data Dependency: NeRF model's reconstruction quality greatly depends on the quality and quantity of input data. For this project, the optimized hyperparameter configuration was only evaluated on the Vegetation dataset. Its general performance for other datasets may be affected.

  Future work directions can include studying more advanced initial sampling strategy and utilizing more powerful GPU to enable training with more input images and higher scene resolutions for more robust ablation studies.

References

[1] Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., & Ng, R. (2020). NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. *Proceedings of the European Conference on Computer Vision (ECCV).*

[2] Tancik, M., Weber, E., Ng, E., Li, R., Yi, B., Kerr, J., Wang, T., Kristoffersen, A., Austin, J., Salahi, K., Ahuja, A., McAllister, D., & Kanazawa, A. (2023). Nerfstudio: A Modular Framework for Neural Radiance Field Development. *ACM SIGGRAPH 2023 Conference Proceedings.*