PocketNeRF: Fast-Converging Neural Radiance Fields for Indoor Reconstruction from Few-Shot Mobile Images

Lucas Brennan, Aaron Jin, Ryan Suh
Department of Computer Science

Problem

Objective: Rapid, mobile-ready 3D reconstruction of indoor scenes from sparse smartphone images.

- Create a **pipeline** that takes as input N = 8 images which we convert into 5D coordinates (x, y, z, θ, Φ).
- The output should be a **compressed Neural Radiance Field** that supports interactive novel-view synthesis on mobile hardware.

Evaluation: We measure reconstruction fidelity and perceptual quality using three standard metrics:

- **Peak Signal-to-Noise Ratio (PSNR),**
- **Structural Similarity Index (SSIM),**
- **Learned Perceptual Image Patch Similarity (LPIPS).**

Background

Neural Radiance Fields: NeRF first enabled creating realistic 3D models from photos, but required **hours of training** per scene.

Instant-NGP: Instant-NGP **accelerates training** using multi-resolution hash tables, from hours to minutes.

Related Approaches: Standard NeRF is too slow + heavy for mobile use. Recent work introduces (1) **geometric priors** for faster guided training; (2) **Adversarial Content-Aware Quantization (A-CAQ)** for model compression.

Dataset

Sample of dataset: Norcliffe common room images captured by sampling frames from an iPhone-recorded video.

Methods

Experiment 0: Preprocessing

Created dataset from iPhone + Polycam.

- **Camera Pose Extraction:** Run COLMAP's Structure-from-Motion
- **EV Normalization:** Compute exposure value (EV) from EXIF
- **Mobile Optimization:** Apply automatic rotation correction, downsample images to maximum dimension, save lossless PNGs

Experiment 1: Structural Priors

Implemented Manhattan-world geometric constraints to accelerate convergence from sparse views. Approach:

- Applied K-means clustering of surface normals + SVD orthogonalization for Manhattan frame detection.
- Identified floor/wall regions via surface normal alignment with estimated coordinate frame.
- Structural loss: $\mathcal{L}^{\text{structural}} = \lambda_1 \mathcal{L}_{\text{manhattan}} + \lambda_2 \mathcal{L}_{\text{planarity}} + \lambda_3 \mathcal{L}_{\text{consistency}}$

Experiment 2: A-CAQ

Implemented differentiable quantization network to compress hash embeddings and MLP weights for mobile deployment. Approach:

- Added 18 learnable bitwidth parameters $b \in [2, 32]$ (init = 8)
- Calibrated quant scales w/ min/max to scale hash values (1e-4)
- Adjust bitwidths based on loss ratio $\rho = L_{cur} / L_{target}$

Trained for 8001 iterations total with 1500 iters warm up →A-CAQ begins at 2000 iters, updating bitwidths every 10 iters.

Analysis

Structural Priors:

- Higher variance (vs. A-CAQ) means constraints work better for some angles than others.
- Successfully identified floor/wall regions, with floor detection more robust due to vertical assumptions.
- Manhattan frame estim. recovered orthogonal directions.

A-CAQ:

- All quantizers converged to the min or max precision, suggesting high sensitivity to hyperparameter selection.
- Despite extreme quantization, maintained competitive test metrics (PSNR = 16.47) with a stable 319MB model size.
- 2-bit quantization did not catastrophically fail, suggesting NeRF features may be more compressible than expected.

Conclusions & Future Work

PocketNeRF enables fast 3D reconstruction from smartphone images using **structural priors** and **A-CAQ quantization**. It achieves aggressive compression (2-8 bits), while maintaining perceptual quality. For next steps, we would like to:

- Synergize structural priors + quantization for compression.
- Tackle 15 dB train-test gap via multi-view consistency losses.
- Expand datasets across other indoor environments.

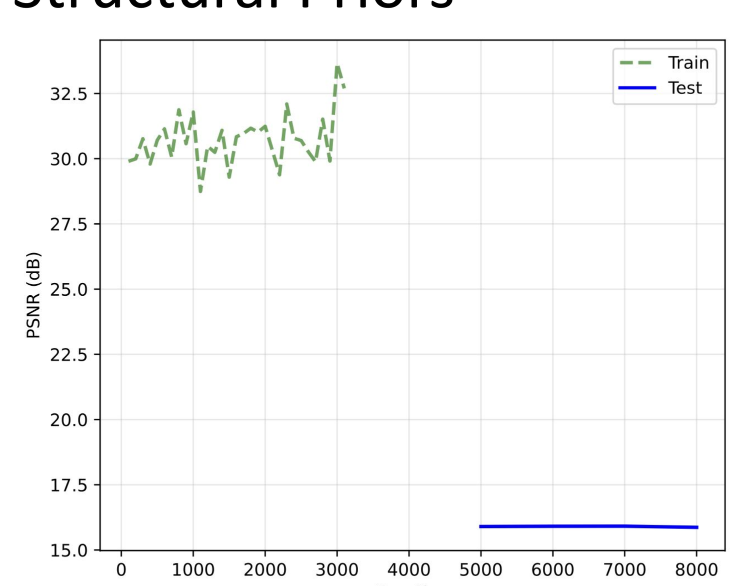
Experiment Results

Final Test Statistics

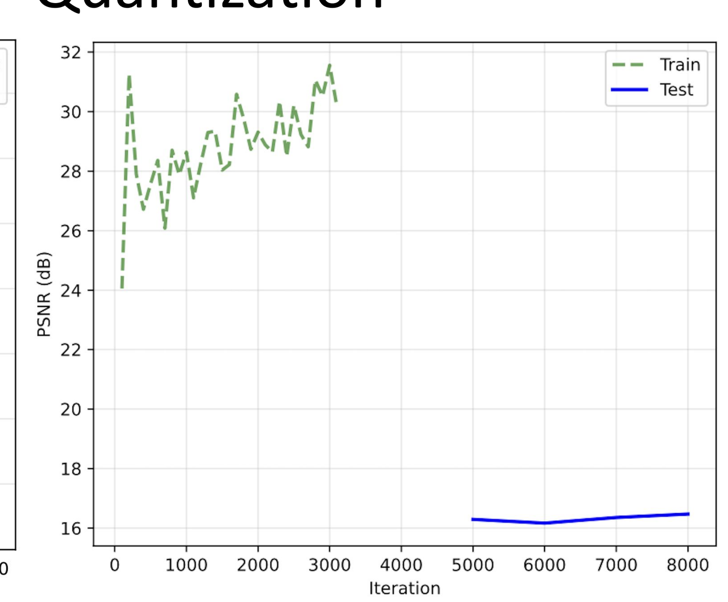
Method	PSNR (dB) ↑	SSIM ↑	LPIPS ↓
A-CAQ	16.47 ± 2.88	0.629 ± 0.104	0.641 ± 0.171
Struct Priors	15.87 ± 3.02	0.485 ± 0.173	0.712 ± 0.137

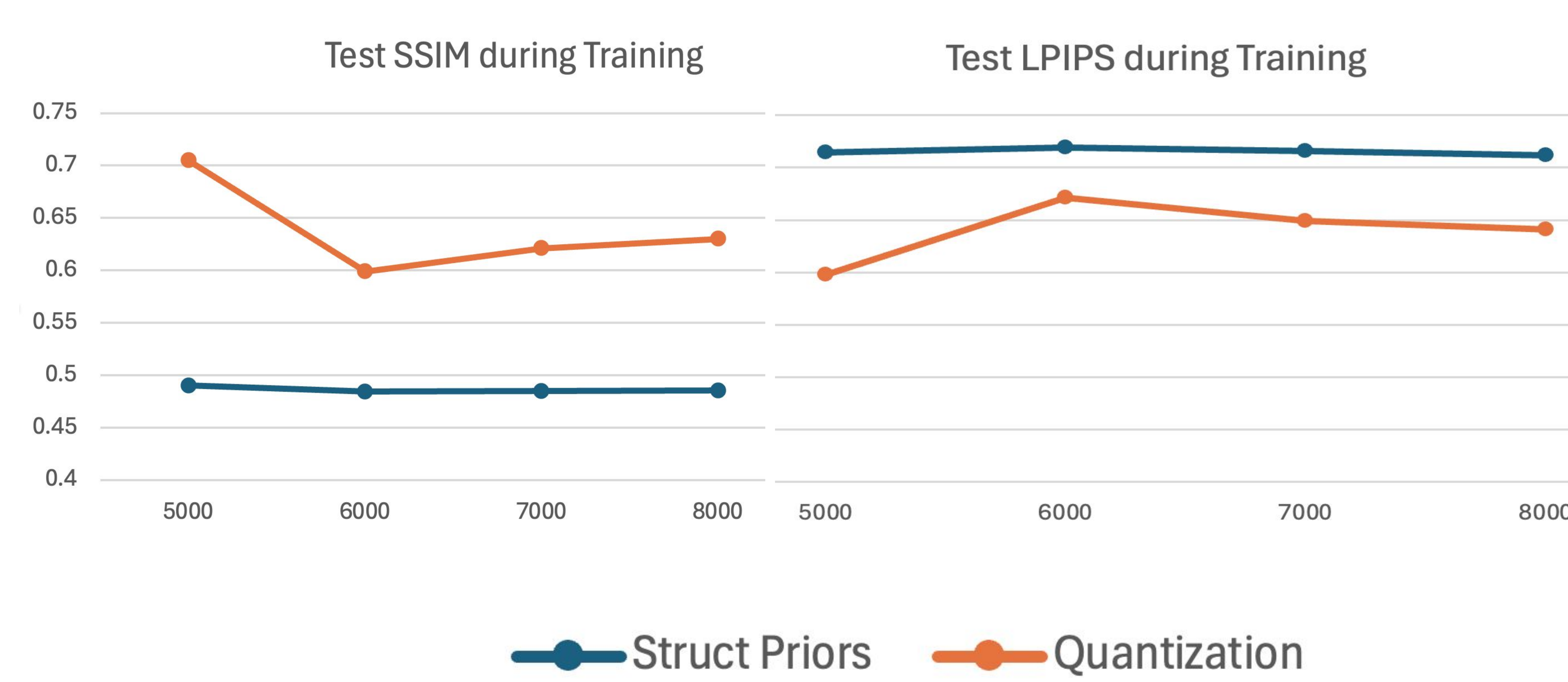
Train Versus Test PSNR:

Structural Priors



Quantization





Note: Our models crashed during training, and were restarted from checkpoint 5000. Due to this, we are only able to provide graphs for behavior between iterations 5000 and 8001.