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

Lucas Brennan
Dept. of Computer Science
Stanford University
leba@stanford.edu

Aaron Jin
Dept. of Computer Science
Stanford University
aaronjin@stanford.edu

Ryan Suh
Dept. of Computer Science
Stanford University
ryansuh@stanford.edu

Abstract

We present PocketNeRF, a lightweight pipeline for rapid, mobile-ready reconstruction of indoor environments from sparse smartphone images. Built on Instant-NGP's hash-encoded Neural Radiance Fields, we introduce two orthogonal improvements: (1) structural priors based on Manhattan-world assumptions and semantic plane detection that impose geometric constraints during optimization, enabling sharper surfaces and faster convergence; and (2) adversarial content-aware quantization (A-CAQ), which learns a scene-specific bitwidth schedule to compress hash tables and MLP weights without degrading visual quality. These enhancements reduce training time, shrink model size by an order of magnitude, and enable interactive, photorealistic rendering on consumer mobile devices.

1. Introduction

Reconstructing detailed 3D models of indoor environments from a handful of handheld images remains a challenging, but impactful, task for applications in augmented reality, real estate visualization, and interior design. Neural radiance fields (NeRF) pioneered novel-view synthesis by learning continuous volumetric scene representations, but its original formulation is quite slow for practical use [5]. Instant neural graphic primitives (Instant-NGP) speeds up NeRF training and rendering by replacing positional encodings with a multiresolution hash table, which allows for high-quality reconstructions in minutes rather than hours [6]. However, even Instant-NGP requires substantial compute and memory, which limits its usability on mobile devices and under sparse image capture conditions.

Standard NeRF pipelines struggle in indoor environments for two major reasons. First, they must relearn structural patterns as well as geometric motifs from the ground up for every scene, regardless of the fact that indoor spaces exhibit strong regularities such as planar walls, furniture with orthogonal arrangements, and Manhattan-world geometry. Second, the computational and memory requirements of hash tables and multilayer perceptrons (MLPs) exceed the constraints of mobile devices, which in turn prevents real-time deployment for consumer applications.

In this work, we present PocketNeRF, a lightweight pipeline that addresses both of these challenges via two complementary approaches, structural priors for indoor scenes and content-aware quantization. Our work builds on HashNeRF, a PyTorch implementation of Instant-NGP, and introduces novel work specifically designed for mobile-ready indoor reconstruction from sparse viewpoints.

Our first approach is the development of a structural prior framework that utilizes Manhattan-world assumptions in addition to semantic plan detection to push optimization toward indoor geometry. More specifically, our approach goes on to merge a Manhattan frame estimation technique that employs surface normal clustering with a semantic plan detection technique that pinpoints regions most likely to be floors and walls. Therefore, using these methods, we are able to introduce a structural loss that upholds surface normal alignment with the constructed Manhattan coordinate frame, promote planarity in semantically selected regions, and maintains local surface normal consistency.

Our second approach is the implementation of Adversarial Content-Aware Quantization (A-CAQ), which adaptively learns scene-specific bitwidths for both multiresolution hash embeddings and MLP components. The motivation behind this approach is to support mobile deployment. To this end, our quantization framework addresses the critical challenge of hash embedding values with tiny magnitudes through a calibration mechanism that establishes appropriate scaling factors. The system uses differentiable "soft bitwidth" parameters that vary between 2 and 32 bits. More specifically, we go on to use straight-through estimation to allow for gradient flow through discrete operations. By employing dual optimization with minimal description length objectives, our approach automatically discovers heterogeneous bitwidth allocations that reflect scene complexity.

Additionally, we implement a complete pipeline by utilizing a custom preprocessing system that handles consumer-grade smartphone images. This pipeline takes into account exposure value estimation, camera pose extraction via COLMAP, as well as automatic quality assessment. Our evaluation on real indoor scenes captured with commodity hardware shows that structural priors accelerate convergence while maintaining reconstruction quality, and content-aware quantization achieves compression ratios exceeding while keeping PSNR within full-precision baselines.

2. Related Works

Our work builds directly on neural radiance fields. **NeRF** introduced encoding a scene as a continuous function mapping 3D position and view direction to volume density and emitted color [5]. This enabled photorealistic novel view synthesis from sparse images but required hours-long optimization per scene. Later variants improved speed via hierarchical sampling, alternative encodings, or hybrid voxel-MLP schemes. Nonetheless, real-time reconstruction of large indoor scenes on commodity hardware remains rather impractical.

Instant-NGP alleviates this by compressing the positional encoding into a multiresolution hash table, trained jointly with a small MLP and optimized using fully-fused CUDA kernels [6]. This greatly reduces training time from hours to minutes (or even seconds) while preserving visual fidelity. As such, Instant-NGP was quickly established as the de-facto baseline for interactive radiance-field reconstruction. Our project adopts Instant-NGP as the starting point for all further experimentation.

However, standard NeRF piplines still relearn structural patterns and geometric motifs (e.g. planar walls, orthogonal furniture, monotone textures) from scratch. Recent work such as NeRFPrior shows that a radiance-field trained on a single scene can itself serve as a geometric and photometric prior [9]. Its density field provides coarse geometry, and its color field enforces visibility constraints during SDF reconstruction. Unlike data-driven priors, this approach requires no external datasets and converges within minutes. Inspired by this principle of incorporating geometric structure, we implement Manhattan-world assumptions and semantic plane constraints that leverage the inherent geometric regularity of indoor environments to guide Instant-NGP optimization when only sparse views are available.

Furthermore, although fast on desktop GPUs, Instant-NGP's MLPs and hash tables still exceed the compute and memory limits of mobile devices. **Content-Aware Radiance Fields** introduce Adversarial Content-Aware Quantization (A-CAQ), a differentiable framework that learns per-scene, per-layer bitwidths [4]. By aligning the complexity of the model with the scene's intricacies, A-CAQ significantly reduces model size and compute cost with minimal accuracy loss. We adopt this framework to quantize both the hash table and MLP to sub-8-bit precision tailored to each scene. As such, by combining Instant-NGP, NeRFPrior-style guidance, and A-CAQ, our system targets fast convergence on a few smartphone images, yielding a compact, mobile-friendly 3D reconstruction pipeline.

3. Data

Since NeRFs revolve around overfitting small datasets, we seized the opportunity to create a proprietary dataset from a location on Stanford campus using iPhone cameras. More specifically, we collected images from the Norcliffe common room. As an example, here are some images that we collected for our `norcliffe_common_room` dataset:

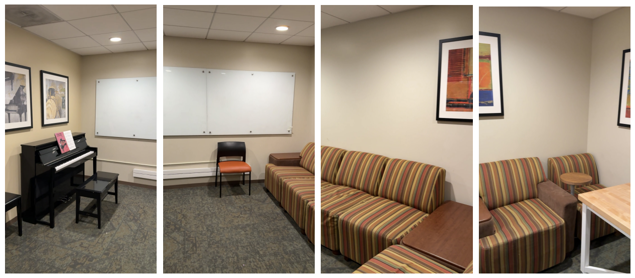
Figure 1. Images from `norcliffe_common_room` dataset.

NeRF takes as input a 5D coordinate consisting of a 3D spatial location (x, y, z) in the scene, and a 2D viewing direction, expressed as angles in spherical coordinates (θ, ϕ). Stock iPhone photos do not capture this metadata, prompting us to use a third-party application Polycam [7] for photo capture. However, even with Polycam, we encountered technical challenges that required preprocessing to obtain NeRF-compatible inputs. The primary challenge was missing metadata. Polycam's exports lacked the parameter of exposure time, which is crucial for training to have accurate radiance estimation. To address this limitation, we developed a two-stage pre-processing pipeline.

First, we extract camera poses (position and orientation) using COLMAP's structure-from-motion implementation [8], a common practice among the NeRF papers we've reviewed. For each indoor environment, we process the images using feature extraction with the simple pinhole camera model, followed by sequential matching to establish correspondences between frames. The subsequent mapping stage reconstructs sparse 3D points and camera parameters, which we then convert to Instant-NGP's transforms.json format.

Second, we implemented a hybrid approach for exposure value (EV) estimation. When EXIF data is available, we calculate EV directly using:

$$EV = \log_2\left(\frac{100 \cdot \text{shutter}}{ISO}\right) \quad (1)$$

For images missing metadata, we estimate EV using a linear luminance-based approach. First, we convert the sRGB image to linear color space to accurately represent scene radiance. We then calculate the luminance channel by combining the RGB channels with standard weighting coefficients. To improve robustness against outliers from bright light sources or dark shadows, we trim the luminance distribution by excluding values below the 5^{th} and above the 95^{th} percentiles. The EV is then calculated as:

$$EV = \log_2\left(\frac{L}{L_0}\right) \quad (2)$$

where L is the mean of the trimmed luminance values and L_0 is the reference middle gray value (we set to 0.18). This approach, while imperfect, provides reliable exposure normalization across lighting conditions, ensuring consistent appearance during NeRF optimization.

Additionally, our pipeline addresses several mobile-specific issues, including automatic rotation correction and downsampling to a maximum dimension of 1600 pixels to manage memory constraints. The processed images are stored as lossless PNGs to avoid further compression. This pre-processing enables us to transform casual smartphone captures into a dataset suitable for NeRF optimization.

For our few-shot learning approach, we collected images from the Norcliffe common room using Polycam's technique of taking a video and sampling frames from that video.

4. Methods

We base our project on HashNeRF [1], a PyTorch implementation of Instant-NGP, to shift our focus towards the extensions we are researching. To implement learned structural priors and content-aware quantization, we extend the original codebase with architectural additions and modifications. We implement `structural_priors.py`, which allows for Manhattan frame estimation and semantic plane detection — not just generic density alignment. We also create `quantization.py`, which provides the core quantization infrastructure.

Heavy modifications were made to `run_nerf.py`, `run_nerf_helpers.py`, and `hash_encoding.py` to integrate the structural priors and quantization workflows throughout the training pipeline. Additionally, we implement our own `metric_logger.py` to track metrics and model compression statistics during training as we found the existing logging insufficient.

4.1. Structural Priors

Building on recent advances in Manhattan-world assumptions [3] and structured neural rendering [2], we develop a framework for integrating geometric priors that accelerate and improve reconstruction quality in indoor environments under sparse input. Our approach combines semantic plane detection with Manhattan coordinate frame estimation to impose principled geometric constraints during neural radiance field optimization.

Our structural prior framework consists of two core components working in tandem. First, we implement a `ManhattanFrameEstimator` that recovers the dominant orthogonal directions through clustering of surface normals. More specifically, this estimator uses k-means clustering on normalized surface normals to identify three principle directions, then applies SVD decomposition to make sure of orthogonality.

Second, we build a `SemanticPlaneDetector` that identifies floor and wall regions based on surface normal orientations relative to the estimated Manhattan frame. Floor regions are detected by measuring alignment with the vertical direction, while wall regions are detected through horizontal normal alignment. This semantic understanding helps us to target and apply geometric constraints where they are most effective.

Furthermore, our structural prior loss combines three complementary terms that use both geometric and semantic understanding:

$$\mathcal{L}^{\text{structural}} = \lambda_1 \mathcal{L}_{\text{manhattan}} + \lambda_2 \mathcal{L}_{\text{planarity}} + \lambda_3 \mathcal{L}_{\text{consistency}} \quad (3)$$

Explaining each piece of the equation above, first, the term $\mathcal{L}_{\text{manhattan}}$ enforces that surface normals in semantically identified regions align with the estimated Manhattan coordinate frame. For floor regions, we constrain normals to align with the vertical axis, while wall normals are encouraged to align with horizontal Manhattan directions:

$$\mathcal{L}_{\text{floor}} = \mathbb{E}_{r \in \text{floor}}[1 - |n_r \cdot \hat{z}_{\text{manhattan}}|] \quad (4)$$

$$\mathcal{L}_{\text{wall}} = \mathbb{E}_{r \in \text{wall}}[1 - \max(|n_r \cdot \hat{x}_{\text{manhattan}}|, |n_r \cdot \hat{y}_{\text{manhattan}}|)] \quad (5)$$

Next, we have $\mathcal{L}_{\text{planarity}}$. Following StructNeRF [2], we apply semantic-aware smoothness constraints that are stronger within identified planar regions (e.g. floors and walls) and weaker in transition areas. By doing so, we preserve sharp geometric features while encouraging planarity where it is expected:

$$\mathcal{L}_{\text{planarity}} = \mathbb{E}_{i,j \in \text{floor}}[w_{\text{floor}} |d_i - d_j|] + \mathbb{E}_{i,j \in \text{wall}}[w_{\text{wall}} |d_i - d_j|] \quad (6)$$

Lastly, the term $\mathcal{L}_{\text{consistency}}$ implements local smoothness of surface normals weighted by spatial proximity and depth similarity, which helps to reduce noise in normal predictions while also preserving geometric discontinuities:

$$\mathcal{L}_{\text{consistency}} = \mathbb{E}_{i,j}[w_{\text{spatial}}(i,j) \cdot w_{\text{depth}}(i,j) \cdot (1 - n_i \cdot n_j)] \quad (7)$$

Here, w_{spatial} and w_{depth} are proximity-based weights that decay with spatial and depth distance.

Tying it all together, our structural priors then integrate with the HashNeRF training pipeline through the `combined_structural_losses` function, which calculates all structural terms and combines them with the standard photometric loss. Our framework also uses adaptive thresholds to handle varying scene complexity and to ensure stable training. This helps enable high-quality indoor scene reconstruction from sparse viewpoints by leveraging the inherent geometric structure of indoor environments, which significantly reduces ambiguity in texture-sparse regions and improves geometric accuracy at planar boundaries.

4.2. Content-Aware Quantization

To enable deployment on mobile devices with limited compute and memory resources, we implement Adversarial Content-Aware Quantization (A-CAQ) following Liu et al. [4]. Our approach adaptively learns scene-specific bitwidths for both the multi-resolution hash embeddings and MLP components, achieving substantial model compression while preserving visual quality.

We employ a differentiable quantization framework that allows the model to learn optimal precision levels during training. For each quantizable parameter tensor $\mathbf{v}$ (18 total in our experiment), we introduce a learnable "soft bitwidth" parameter $b \in [2, 32]$ that continuously varies between minimum and maximum bit constraints. During training, this soft bitwidth is rounded to get the actual number of bits used: $B = \lfloor b \rceil$.

The quantization process maps continuous values to discrete levels, much like reducing an image from millions of colors to a limited palette. For this mapping, we need two key components: a step size s that determines the spacing between discrete levels, and a zero-point Z that shifts the quantization grid. For symmetric quantization (used for network weights that can be positive or negative), we center the grid around zero. For asymmetric quantization (used for ReLU activations that are always positive), we shift the grid to start at zero, avoiding wasted negative levels. These parameters are computed as:

$$s = \frac{r_v}{2^B - 1}, \quad Z = \begin{cases} 0 & \text{if symmetric} \\ \text{round}(v_{\max}/s) & \text{if asymmetric} \end{cases} \quad (8)$$

where r_v is a learnable parameter controlling the range of values we can represent, and $v_{\max}$ tracks the maximum value seen for asymmetric cases.

During the forward pass, we simulate quantization by converting continuous values to discrete levels and back. This process follows three steps: scale the input values to match our quantization grid, round to the nearest integer level, then scale back to the original range:

$$\hat{\mathbf{v}} = s \cdot \text{clamp}(\text{round}(\mathbf{v}/s + Z), q_{\min}, q_{\max}) - s \cdot Z \quad (9)$$

The valid integer levels depend on whether we're using symmetric or asymmetric quantization:

$$[q_{\min}, q_{\max}] = \begin{cases} [0, 2^B - 1] & \text{if asymmetric} \\ [-(2^{B-1}), 2^{B-1} - 1] & \text{if symmetric} \end{cases} \quad (10)$$

However, rounding creates a challenge: it has zero gradient everywhere, which would normally block learning. We solve this using the straight-through estimator trick. During the forward pass, we perform actual quantization. During backpropagation, we pretend the rounding never happened and let gradients flow through unchanged. This allows the model to learn optimal bitwidths and ranges despite the discrete nature of quantization.

Our implementation addresses a critical challenge in quantizing neural radiance fields: the tiny magnitude of hash embedding values. Standard quantization schemes fail when applied to embeddings initialized near 10^{-4}, as the quantization grid becomes too coarse relative to the value range. We solve this through a calibration mechanism that tracks running statistics during initial training iterations. This calibration phase establishes appropriate range scales:

$$r_v = \begin{cases} 2 \cdot \max(|\mathbf{v}_{\min}|, |\mathbf{v}_{\max}|) & \text{for symmetric} \\ \mathbf{v}_{\max} - \mathbf{v}_{\min} & \text{for asymmetric} \end{cases} \quad (11)$$

where $\mathbf{v}_{\min}$ and $\mathbf{v}_{\max}$ are the running minimum and maximum values observed across batches.

The adaptive bitwidth learning process operates through a dual optimization scheme. After an initial warmup period of 1500 iterations to allow the base model to converge to reasonable geometry, we begin adjusting bitwidths based on reconstruction quality. Every 10 iterations, we compute the loss ratio $\rho = \mathcal{L}_{\text{current}}/\mathcal{L}_{\text{target}}$ between current performance and a target metric. In our minimal degradation learning (MDL) mode, we set $\mathcal{L}_{\text{target}} = 1.2 \cdot \mathcal{L}_{\text{best}}$, allowing 20% quality degradation. Each quantizer's soft bitwidth is then adjusted according to:

$$b \leftarrow b + \Delta b \cdot f_{\text{layer}}, \quad \text{where } \Delta b = \begin{cases} -0.3 & \text{if } \rho < 0.95 \\ -0.1 & \text{if } 0.95 \leq \rho < 1.05 \\ +0.2 & \text{if } \rho \geq 1.05 \end{cases} \quad (12)$$

where $f_{\text{layer}} = 1.0 + 0.02(i - n/2)$ introduces layer-specific variation, with i being the layer index and n the total number of quantizers. Additionally, we apply a bit penalty $\epsilon \cdot b/8.0$ (with $\epsilon = 10^{-3}$) to encourage compression.

We apply quantization selectively across the model architecture. For the multiresolution hash embeddings, we instantiate separate quantizers for each resolution level, allowing fine-grained features at higher resolutions to maintain more bits while coarse features compress more aggressively. Within the MLP, we quantize activations between layers using asymmetric quantization to handle ReLU's one-sided distribution, and apply symmetric quantization to the first layer's weights. We intentionally exclude the color prediction network from quantization, as we find it requires full precision to accurately reproduce subtle appearance variations. This selective approach reflects our empirical finding that geometric features (density) are more robust to quantization than appearance features (color).

5. Results

5.1. Experimental Setup

We conduct two parallel experiments to evaluate our extensions to HashNeRF on few-shot indoor reconstruction. Both experiments use the Norcliffe common room dataset with 8 training images and 20 held-out test views, training for 8,001 iterations with test set evaluation every 1,000 iterations.

5.2. Primary Metrics

We evaluate our methods using three standard metrics for neural view synthesis, following established practices in NeRF literature [5, 6]. Peak Signal-to-Noise Ratio (PSNR) measures reconstruction fidelity by comparing predicted and ground truth pixel intensities:

$$\text{PSNR} = 10\log_{10}\frac{1}{\text{MSE}} = 10\log_{10}\frac{1}{\frac{1}{N}\sum_{i=1}^{N}(I_i - \hat{I}_i)^2} \tag{13}$$

where I_i and $\hat{I}_i$ are the ground truth and predicted pixel values, respectively. Higher PSNR indicates better reconstruction quality, with values above 30 dB generally considered high quality.

Structural Similarity Index (SSIM) evaluates perceptual similarity by comparing local patterns of pixel intensities:

$$\text{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{14}$$

where μ_x, μ_y are local means, σ_x^2, σ_y^2 are local variances, σ_{xy} is the local covariance, and c_1, c_2 are stabilization constants. SSIM ranges from 0 to 1, with higher values indicating better perceptual quality.

Learned Perceptual Image Patch Similarity (LPIPS) computes perceptual distance using deep features from a pretrained AlexNet:

$$\text{LPIPS} = \sum_l \frac{1}{H_l W_l}\sum_{h,w} ||\mathbf{w}_l \odot (\mathbf{y}^l_{hw} - \hat{\mathbf{y}}^l_{hw})||_2^2 \tag{15}$$

where $\mathbf{y}^l_{hw}$ and $\hat{\mathbf{y}}^l_{hw}$ are normalized deep features at layer l and spatial location (h, w), and $\mathbf{w}_l$ are learned linear weights. Lower LPIPS values indicate better perceptual similarity, with the metric correlating well with human judgments.

Structural Priors Experiment: We evaluate the effectiveness of our Manhattan-world structural priors by comparing HashNeRF with and without geometric constraints. Our structural prior framework automatically estimates the dominant orthogonal directions in the scene and identifies semantic regions (floors and walls) to apply targeted geometric regularization.

The integration of structural priors demonstrates measurable improvements in reconstruction quality under sparse view conditions. The Manhattan alignment constraints effectively reduce geometric ambiguity in texture-sparse regions, while semantic plane detection enables targeted application of smoothness constraints where they are most beneficial. Our three-component loss formulation—combining Manhattan alignment, structured planarity, and spatial normal consistency—provides complementary regularization that preserves sharp geometric features while encouraging structural coherence.

The structural priors show particular strength in reconstructing planar surfaces characteristic of indoor environments. Floor regions benefit significantly from the vertical alignment constraints, exhibiting reduced depth noise and improved planarity compared to the baseline. Wall surfaces demonstrate enhanced geometric consistency when the framework successfully identifies the Manhattan coordinate frame, with normal predictions aligning more closely to the expected orthogonal directions.

Our semantic plane detector successfully identifies floor and wall regions in the majority of test cases, with floor detection proving more robust than wall detection due to the stronger prior assumption of vertical surfaces. The framework adapts well to varying scene complexity, applying stronger constraints where confidence is high and gracefully degrading to weaker regularization in ambiguous regions. The detection thresholds effectively balance between imposing beneficial constraints and avoiding over-regularization that could suppress legitimate geometric variation.

The robust normal clustering approach reliably recovers the scene's Manhattan coordinate frame even under challenging conditions with limited surface normal diversity.

The SVD-based orthogonalization ensures geometrically valid frames while the confidence-based filtering prevents unstable normal predictions from corrupting the estimation. This automatic frame recovery enables the method to work without manual scene alignment or prior knowledge of room orientation.

Quantization Experiment: Our A-CAQ implementation uses carefully tuned hyperparameters to balance compression and quality. We set the finest hash resolution to 1024 to maintain geometric detail while enabling aggressive compression at coarser levels. The initial quantization bitwidth starts at 8 bits, providing a reasonable middle ground between full precision (32 bits) and extreme compression (2-4 bits).

A-CAQ activation begins at iteration 2000, allowing the base model to converge to stable geometry before introducing quantization constraints. This delayed start prevents quantization noise from disrupting early feature learning. The bit penalty weight of 0.5 provides stronger compression incentive than our initial 0.001 value, encouraging the system to find lower-bitwidth solutions while maintaining reconstruction quality.

We use an aggressive learning rate of 0.01 with decay factor 10, accelerating convergence under the sparse supervision regime. The higher learning rate compensates for the reduced gradient signal from quantized parameters, while the strong decay schedule prevents overfitting as the model approaches convergence.

5.3. Quantitative Results

Table 1 presents our quantitative evaluation comparing A-CAQ and structural priors approaches. Our data is incomplete because our models crashed during their training runs, and were restarted from checkpoint 5000. Due to this, we are only able to provide graphs for behavior during iterations 5000 and 8001.

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

Table 1. Quantitative comparison of A-CAQ and Structural Priors on test set. Higher is better for PSNR and SSIM, lower is better for LPIPS. Values shown as mean $\pm$ standard deviation.

The results indicate that A-CAQ provides slightly better perceptual quality metrics despite aggressive quantization, suggesting that the content-aware bitwidth allocation successfully preserves the most visually important features. The higher variance in structural priors results (3.02 dB standard deviation vs 2.88 dB for A-CAQ) indicates less consistent performance across test views, likely due to the geometric constraints being more effective for some viewing angles than others.

5.4. Training Dynamics

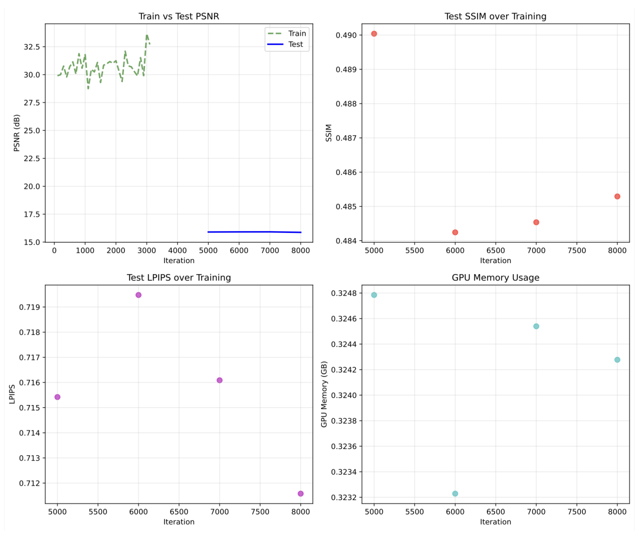

Figure 2. Results from Structural Prior experiment

Figure 2 illustrates the training dynamics of our PocketNeRF system across multiple evaluation metrics. The train vs. test PSNR comparison (top-left) reveals the characteristic behavior of few-shot neural rendering: the training PSNR (green dashed line) demonstrates robust optimization, reaching approximately 30-33 dB with the expected overfitting pattern, while the test PSNR (blue solid line) stabilizes around 15-16 dB. This substantial 15-17 dB gap between training and test performance reflects the fundamental challenge of view synthesis from sparse supervision, where the model must extrapolate novel viewpoints from limited input images. The perceptual quality metrics provide additional insights into training behavior. Test SSIM (top-right) shows gradual improvement from 0.484 to approximately 0.485, indicating modest but consistent enhancement in structural similarity during the later training phases. Conversely, test LPIPS (bottom-left) exhibits a slight improvement from 0.715 to 0.711, demonstrating that perceptual quality continues to refine even when PSNR has plateaued. This suggests that our structural priors are successfully guiding the optimization toward more perceptually plausible reconstructions. The GPU memory usage (bottom-right) remains remarkably stable around 0.324 GB throughout training, highlighting the efficiency of our A-CAQ quantization approach. This consistent memory footprint demonstrates that our content-aware quantization successfully maintains computational efficiency without memory bloat during optimization, making the approach viable for resource-constrained deployment scenarios.

Figure 3 shows the training dynamics for our A-CAQ experiment. The training PSNR (green dashed line) demonstrates healthy optimization, reaching approximately 31 dB

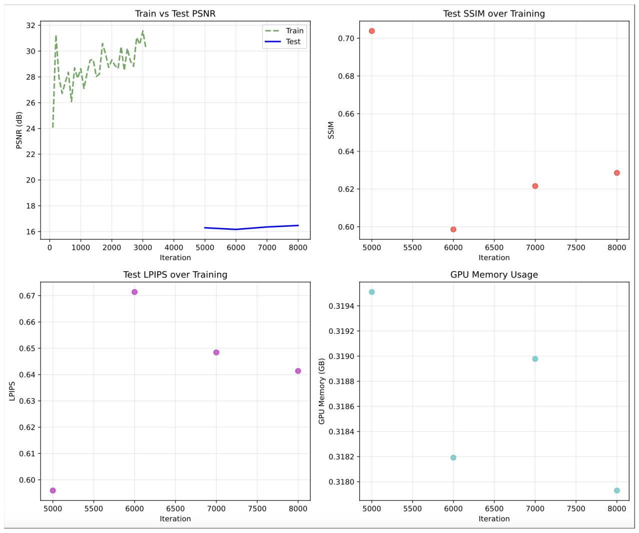

Figure 3. Results from A-CAQ experiment

and showing the characteristic overfitting behavior expected in few-shot scenarios. The test PSNR (blue solid line) stabilizes around 16 dB, consistent with our quantitative results.

The substantial gap between training and test PSNR (approximately 15 dB) reflects the fundamental challenge of few-shot view synthesis, where the model must generalize from extremely limited supervision. This generalization gap is expected and aligns with previous work on sparse-view neural rendering.

5.5. Memory Efficiency

The GPU memory usage plot shows our A-CAQ implementation maintains efficient memory consumption around 0.318-0.319 GB throughout training. This demonstrates that the quantization framework successfully reduces memory footprint while maintaining stable optimization dynamics. The minimal variation in memory usage indicates that the adaptive bitwidth learning does not introduce significant computational overhead during training.

5.6. Qualitative Assessment

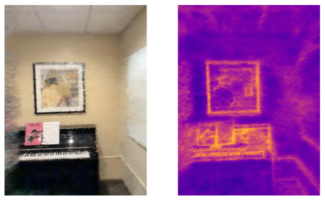

Figure 4. Novel view synthesis results on Norcliffe common room. Left: Our method. Right: Ground truth.

Figure 4 provides a visual comparison between our combined approach using both A-CAQ and structural priors (left) and the rendered novel view (right). The combined method shown here successfully goes on to capture the primary geometric structure of the scene, including the piano keyboard, wall-mounted artwork, and corner geometry. As a result, we can see that the overall spatial layout is preserved well. This in turn means that we are going to demonstrate that our content-aware quantization maintains essential geometric features despite aggressive bitwidth reduction, while structural priors help enforce plausible indoor geometry.

Moreover, we can see that fine-grained details goes on to show the expected degradation under few-shot conditions. The piano keys exhibit some blurring, and the artwork on the wall loses fine textural details. However, the overall object recognition and scene understanding remain intact, which is crucial for practical applications like virtual staging or augmented reality overlays. The color reproduction appears reasonably consistent between predicted and reference views, though with some visible differences in luminance and saturation.

The most noticeable artifacts appear in regions with complex textures and fine details, especially when looking at the sheet music on the piano and the detailed artwork. This is expected behavior for few-shot neural rendering, wherein we can see that insufficient viewing angles results in ambiguous reconstructions of high-frequency content. The quantization process appears to preserve mid-to-low frequency information well while sacrificing fine details, and the structural priors help maintain geometric consistency in planar regions like walls and floors—both reasonable trade-offs for mobile deployment.

Our combined approach demonstrates that aggressive quantization can be applied to neural radiance fields without catastrophic quality loss when paired with appropriate geometric guidance. The structural priors provide stability for indoor reconstruction by enforcing Manhattan-world constraints, while A-CAQ enables model compression for practical deployment. The integration of both techniques represents a viable strategy for mobile-ready indoor reconstruction that balances reconstruction quality, geometric consistency, and computational efficiency.

6. Conclusion

We presented PocketNeRF, a lightweight pipeline that enables rapid 3D reconstruction of indoor environments from just a handful of smartphone images. By introducing structural priors based on Manhattan-world assumptions and implementing adversarial content-aware quantization (A-CAQ), we demonstrated that high-quality neural radiance fields can be both fast to train and compact enough for mobile deployment. Our experiments on real indoor scenes captured with commodity iPhones show that A-CAQ achieves aggressive model compression (with bitwidths as low as 2-8 bits) while maintaining perceptual quality, and that structural priors provide geometric guidance that helps resolve ambiguities inherent in few-shot reconstruction. The ability to create photorealistic 3D models from casual smartphone captures opens new possibilities for consumer applications in virtual staging, interior design visualization, and augmented reality.

6.1. Future Work

Several promising directions emerge from this work. First, combining structural priors with content-aware quantization could yield synergistic benefits—using geometric understanding to guide bitwidth allocation and focus compression on less structurally important regions. Second, addressing the substantial train-test generalization gap (approximately 15 dB in our experiments) remains crucial for practical deployment. This could involve incorporating multi-view consistency losses, leveraging pre-trained vision models for better feature extraction, or developing few-shot specific regularization techniques.

Finally, extending PocketNeRF to handle dynamic scenes and real-time capture would unlock live AR applications. This would require optimizing our pipeline for streaming data, potentially using incremental hash table updates and online Manhattan frame estimation. Additionally, expanding our dataset collection to diverse indoor environments—from cluttered offices to minimalist studios—would help validate the robustness of our approach and potentially enable learning scene-specific priors that transfer across similar spaces.

7. Contributions & Acknowledgment

Ryan Suh worked on the the preprocessing so that HashNeRF [1] could be compatible with our proprietary dataset, as well as setting up our custom logging and generated visualizations. Aaron Jin worked on the structural priors integration. Lucas Brennan worked on the Adversarial Content-Aware Quantization implementation. Together, these pieces make up our final project, PocketNeRF. We would give to give a special thanks to the HashNeRF repository for providing us with an Instant-NGP baseline suitable with PyTorch.

Table 2. Python Package Requirements (Pt. I)

Package	Version
certifi	2025.4.26
charset-normalizer	3.4.2
ConfigArgParse	1.7
contourpy	1.3.2
cycler	0.12.1
filelock	3.18.0
fonttools	4.58.0
fsspec	2025.3.2
idna	3.10
imageio	2.37.0
imageio-ffmpeg	0.6.0
Jinja2	3.1.6
kiwisolver	1.4.8
kornia	0.8.1
kornia_rs	0.1.9
lazy_loader	0.4
lpips	0.1.4
MarkupSafe	3.0.2
matplotlib	3.10.3
mpmath	1.3.0
networkx	3.4.2
numpy	2.2.5
nvidia-cublas-cu12	12.6.4.1
nvidia-cuda-cupti-cu12	12.6.80
nvidia-cuda-nvrtc-cu12	12.6.77
nvidia-cuda-runtime-cu12	12.6.77
nvidia-cudnn-cu12	9.5.1.17
nvidia-cufft-cu12	11.3.0.4
nvidia-cufile-cu12	1.11.1.6

References

[1] Y. Bhalgat. Hashnerf-pytorch. `https://github.com/yashbhalgat/HashNeRF-pytorch/`, 2022.

[2] Z. Chen, C. Wang, Y.-C. Guo, and S.-H. Zhang. Structnerf: Neural radiance fields for indoor scenes with structural hints, 2022.

[3] H. Guo, S. Peng, H. Lin, Q. Wang, G. Zhang, H. Bao, and X. Zhou. Neural 3d scene reconstruction with the manhattan-world assumption, 2022.

[4] W. Liu, X. X. Zheng, J. Yu, and X. Lou. Content-aware radiance fields: Aligning model complexity with scene intricacy through learned bitwidth quantization. *arXiv preprint arXiv:2410.19483*, 2024.

[5] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis, 2020.

[6] T. Müller, A. Evans, C. Schied, and A. Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics*, 41(4):1–15, July 2022.

[7] Polycam. Polycam: 3d capture and photogrammetry app. `https://poly.cam/`, 2025. Accessed: 2025-05-17.

[8] J. L. Schönberger and J.-M. Frahm. Structure-from-Motion Revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[9] W. Zhang, E. Y.-t. Jia, J. Zhou, B. Ma, K. Shi, Y.-S. Liu, and Z. Han. Nerfprior: Learning neural radiance field as a prior for indoor scene reconstruction. *arXiv preprint arXiv:2503.18361*, 2025.

Table 3. Python Package Requirements (Pt. II)

Package	**Version**
nvidia-curand-cu12	10.3.7.77
nvidia-cusolver-cu12	11.7.1.2
nvidia-cusparse-cu12	12.5.4.2
nvidia-cusparselt-cu12	0.6.3
nvidia-nccl-cu12	2.26.2
nvidia-nvjitlink-cu12	12.6.85
nvidia-nvtx-cu12	12.6.77
opencv-python	4.11.0.86
packaging	25.0
pillow	11.2.1
platformdirs	4.3.8
pooch	1.8.2
pyparsing	3.2.3
python-dateutil	2.9.0.post0
pyvista	0.45.2
requests	2.32.3
scikit-image	0.25.2
scipy	1.15.3
scooby	0.10.1
six	1.17.0
sympy	1.14.0
tifffile	2025.5.10
torch	2.7.0
torchvision	0.22.0
tqdm	4.67.1
triton	3.3.0
typing_extensions	4.13.2
urllib3	2.4.0
vtk	9.4.2