Augmenting with NeRFs: Fast Relocalization on Densified Datasets

Michael Tomadakis[1] Rebecca Borissova[1] Yuxuan Zhang[1] Sanjeev Koppal[1,2*]
[1]University of Florida [2]Amazon Robotics
{m.tomadakis, rborissova, zhangyuxuan, sjkoppal}@ufl.edu

Abstract

We reinterpret NeRFs as a resource for extreme data augmentation to advance the field of camera relocalization. Our approach lets us automatically render a massive, densified dataset of novel views, given only sparse ground-truth viewpoints. We introduce a filtering strategy that, compared to existing novel-view-synthesis-focused relocalizers, does not rely on custom or specific NeRF backbones. This filtering strategy allows for significant spatial extrapolation within the scene, without compromising novel view quality. As a result, training a lightweight off-the-shelf vision backbone as a pose regressor on our expanded datasets significantly improves accuracy, uniquely enables relocalization of very spatially-novel views, and performs well on portable-scale hardware.

1. Introduction

Relocalization—positioning a camera view within a known scene—is a foundational challenge for countless applications from photogrammetry to augmented reality. In this work, we address the gap in accuracy and robustness among the few real-time-capable relocalization models. Our key idea is to leverage the high-quality view synthesis of neural radiance fields to move the mapping process *offline*, and automatically perform extreme data augmentation that extrapolates far beyond its initial spatial domain.

This vast augmentation lets us train light, high-sample-complexity models from the UniRepLKNet [12] family, frontload the computational burden of scene understanding, and decouple the original training set size from final model accuracy. Our augmentation **more than doubles the accuracy of multiple architectures** relative to an unaugmented baseline, and requires minimal hand-tuning, if any. Using Instant-NGP [27] as our NeRF, we apply our method to train UniRepLKNet on traditional datasets 7Scenes [14] and Cambridge-Landmarks [19] demonstrating the relative impacts of our backbone, augmentation, and filtering strategy relative to various relevant SotA relocalizers.

Our model is robust to sparse data and extreme viewpoint changes, a challenge we term "faraway relocalization" (FR). Scenes like stadiums or warehouses exemplify this common challenge: spatially-limited training data may exist despite localization objectives in distant regions (e.g. soccer field vs stadium seats). Yet the most common relocalization benchmarks, which fuel intense competition in pursuit of SotA performance, often contain spatially adjacent train/test sets, leaving FR underexplored. We introduce a simple filtering strategy to remove low-quality samples that deter comparable approaches from such extrapolation, and contribute multiple new FR-focused synthetic scenes to show that sparsity and FR challenge or completely break otherwise competitive methods like DSAC* and DFNet, but not ours. We show that only around 200 views are sufficient to train our model in a large, detailed environment, and that the large-kernel UniRepLKNet backbone is uniquely resilient to our patterned, symmetric, or textureless scenes.

Additionally, our method handles challenging real-life environments while remaining fast enough for applications in robotics. We demonstrate our success on mobile hardware in a repetitive office environment that challenges even classical structure-from-motion (SfM) approaches. With minimal tuning of view sampling parameters, our trained model runs on an Nvidia Orin at 50+ FPS and successfully disambiguates nearly-identical regions of a scene.

We summarize our contributions below, and outline our proposed pipeline in Figure 1:

- Demonstrate NeRF-based augmentation to dramatically improve lightweight relocalizer accuracy while maintaining efficiency (Section 4.1)
- Reveal a gap in FR among existing techniques through synthetic environments in which our method excels (Section 4.1, Section 4.2)
- Showcase real-time portable hardware performance (∼50 FPS on Nvidia Orin) in a real demo (Section 4.3)

*Sanjeev J. Koppal holds concurrent appointments as an Associate Professor of ECE at the University of Florida and as an Amazon Scholar at Amazon Robotics. This paper describes work performed at the University of Florida and is not associated with Amazon.

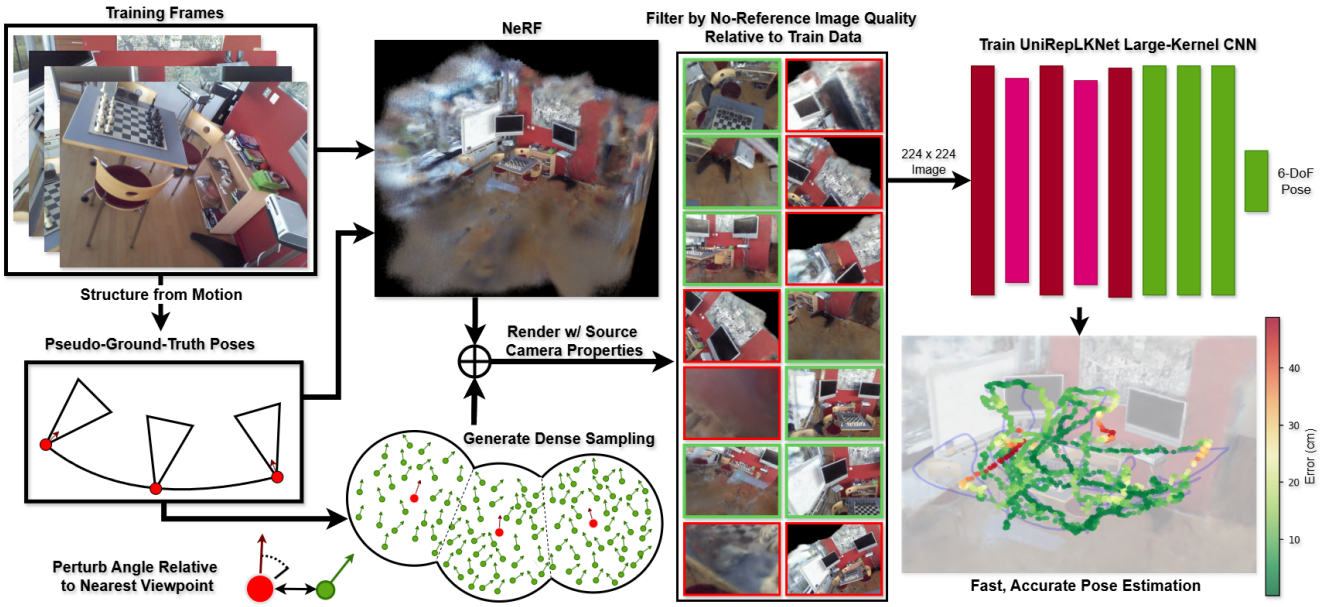


Figure 1. We outline our proposed pipeline from unposed frames to a trained relocalization model. We emphasize our sample-generation strategy which produces novel poses in the neighborhood of ground-truth poses as well as our filter strategy which mitigates negative outcomes caused by over-extrapolating.

Table 1. Model Feature Comparison

Feature	DUSt3R	DFNet / DSAC*	PoseNet	Our Method
< 1GB Memory at Inference	×	✓	✓	✓
Faraway Relocalization	✓	×	×	✓
Inference Time	∼100ms	∼5-25ms	5-10ms	5-10ms
Relative Error on Trad. Benchmarks	0.06x	0.35-0.06x	1x	0.38x

2. Background

Classical relocalization, first introduced in 1981 [13], relies on feature matching and bundle adjustment, a mainstay of structure-from-motion pipelines. Many variations have since emerged both in academic literature and commercial software for photogrammetry and camera tracking, like Meshroom [17], COLMAP [32], and RealityCapture.

Early Neural Relocalization In 2016, PoseNet [19] first demonstrated successful one-shot neural relocalization, mapping single images directly to a 6-DoF pose estimate through a convolutional neural network (GoogLeNet [34]), with minimal compute and memory overhead, but low accuracy compared to classical approaches. An array of academic literature has emerged since, pushing the frontier in neural relocalization primarily through a combination of RANSAC and neural mapping, as in [3–5, 21]. Others perform RANSAC/ PnP on dense features, like [35].

Scene Coordinate Regressors Feature matching and PnP of early neural relocalizers evolved toward direct 3D coordinate prediction per pixel. These approaches decouple the perspective-n-point [13] problem from image identification or feature recognition. In [3–5], Brachmann et. al. explore training a CNN using reprojection loss to predict scene coordinates from images. In DSAC*, they show that this CNN learns an implicit scene representation without a depth prior, by slowly converging to a consistent scene coordinate output. Later advancements like Ace and Acezero [2, 6] improve upon the organization and processing time of these features but leverage the same fundamental DSAC* scene-coordinate-regression. Despite their state-of-the-art accuracy critically rely on a small receptive field that struggles with repetitive features (81×81 pixels in the case of DSAC* and its successors) [5], unlike our choice of UniRepLKNet which exhibits a global receptive field. Furthermore, related approaches have traded PoseNet's simplicity for costly iteration, often querying a map at test-time, neural or explicit, which occupies a costly memory footprint, as in [21, 31, 36].

Large Transformers Most recently, works like DUSt3R [40], Mast3R [20], and VGGT [38] have pushed boundaries in relative camera localization through approaches mirroring trends in large language models: "troves" of data, and massive but straightforward transformer architectures. These approaches boast inference times as low as 50ms, and generalize from massive datasets which enable relative localization of numerous camera poses in wholly unknown environments, no training poses needed. However, they are unsuited to the task of lightweight relocalization due to a significant memory footprint for suitable queries (*poten-*

tially over 40GB at inference), and lacking training context, cannot absolutely metrically relocalize views. Marepo [9], a recent work from the authors of [2, 3, 5, 6], overcomes the lack of metric pose in the other large transformer models and boasts greater inference efficiency, lower than 20ms. However, it too is prone to profound memory constraints (V100 recommended for inference), excluding it from the lightweight / portable regime of models we emphasize in this work.

NeRFs and Augmentation-centric Relocalizers Neural Radiance Fields [23] enable realistic novel view synthesis (NVS). Fast implementations which leverage Instant-NGP's (INGP) hash encoding [27], make large-scale view synthesis tractable. An array of literature leveraging NeRF NVS in localization has recently emerged. Several fall victim to slow test-time iteration [8, 41, 43, 44] (5-20+ seconds/query on desktop hardware), exempting them from the lightweight regime we target. Others perform their NVS offline, like ours [10, 30, 39], but suffer from shortcomings in sampling strategies such as constraining to a horizontal plane [26, 30], or lack a robust filter to permit scene extrapolation [10]. Our closest competitors are DFNet and LENS [10, 26]. DFNet, biases synthetic training data towards ground truth, unlike our uniform sampling, and, lacking a filter, cannot extrapolate for FR. LENS uses a proprietary, unavailable model from its authors, CoordiNet [25], and constrains its synthesis to a plane, catering to traditional benchmark datasets.

While transformer-based models like DUSt3R [40] and NeRF-based methods such as PNeRFLoc [43] and CaLDiff [33] offer great relocalization accuracy, their computational footprints limit their feasibility in real-time or portable applications (*seconds* per frame on desktop hardware), while lightweight competing models are not robust to repetitive scenes or spatially-novel views.

3. Methods

Our method consists of two main stages: NeRF-based data augmentation and pose regressor training.

3.1. NeRFs as a Dataset Expansion Tool

Our proposed pipeline begins with computing a NeRF of each scene. While the choice of NeRF architecture may benefit convergence, our sampling strategy is agnostic to the architecture used and depends optionally on efficient NeRF rendering, in order to make the generation of new samples computationally tractable at scale.

NeRF Training / SfM Producing a NeRF requires ground-truth poses for the training dataset of imagery. Our method therefore relies on existing SfM pipelines, but the technique for recovering these poses falls outside the scope of this work. We use Alicevision/Meshroom [17] almost exclusively due to its performance and accessibility. We apply INGP [27] as our NeRF for all experiments for similar merits, and utilize INGP's full suite of optimization options (pose, exposure, intrinsics, latents) where they were found to improve convergence. No depth priors are used in any experiments, nor do we compare against RGB-D relocalizers. All NeRFs are trained strictly using the training subset of data, including those of synthetic scenes, in which synthetic data is rendered using Blender [11]. Additionally, periodic resets of the INGP optimizer are performed, and the standard INGP learning rate is reduced by a factor of 10 to improve convergence. We note that while a convergent NeRF is preferable, even noisy NeRFs produce adequate results due to both the low resolution of imagery used in training and our backbone's robustness to noise. We plot final training losses of our NeRFs against the best relocalization results achieved on their datasets in Figure 4 to show the lack of correlation.

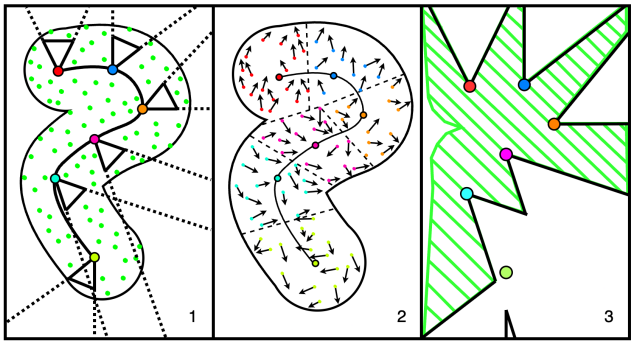


Figure 2. Our data augmentation method extrapolates, rather than interpolates, to capture an enlarged volume of the scene: we sample new camera poses near ground-truth viewpoints (1) with perturbed orientations relative to nearest-source-pose neighbor (2) to dramatically expand scene coverage (3). This results in training data with a more thorough coverage of 3D features, though at the cost of perceiving unconvergent regions of NeRF which must be addressed through filtering.

Sampling Having recovered ground-truth poses and a NeRF, we uniformly sample a region within some radius of those poses, placing new cameras, and aligning them to the nearest ground-truth pose's orientation, which we subsequently perturb by some angle (detailed in Section 4). We then re-image the NeRF from this new set of sample poses. This process goes beyond traditional data augmentation (e.g., affine transforms, color jitter), which cannot synthesize the truly novel viewpoints necessary to address FR, a property we emphasize in Figure 2. Unless otherwise noted, we render 10,000 new perturbed views of each scene configured with matching camera parameters to the test set.

Filtering The augmented data is filtered in order to omit images from cameras which have strayed far enough from the training domain to image NeRF regions with poor convergence. We first annotate synthesized images by black

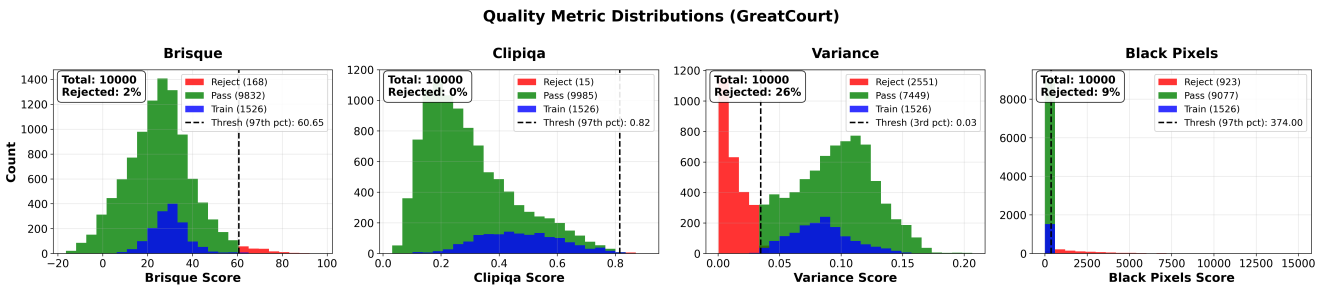


Figure 3. We illustrate filter threshold selection based on the distribution of referenceless image quality metrics in the original training data, and the effects of the selected cutoffs on the *augmented* data from the Cambridge-GreatCourt scene.

pixel count, image variance, BRISQUE score [24], and ClipIQA score [37]. While BRISQUE and ClipIQA help identify semantically meaningful or photo-like structure, variance is responsible for omitting the bulk of imagery, since most problematic samples manifest as noise. The black pixel count helps omit photos that look beyond the edge of the NeRF domain – rays that do not intersect with the volume.

For each score, we experiment with filtering at two different thresholds: (1) at the 99.5th percentile values of the training data (i.e., images with more black pixels than the top 99.5th-percentile-by-black-pixels image in the training set, or lower variance than the bottom 0.5th percentile of the training set); and (2) more conservatively, at the 97th/3rd percentile of the training data. We find that image filtering decouples model accuracy from the degree of sample perturbation/radius. Image filtering ensures that even a naively-sampled scene with gross extrapolation produces results comparable to a conservatively-sampled scene, thereby reducing the need to tune sampling parameters. This sets us apart from approaches like DFNet [10], which *cannot filter out poor-quality poses* and therefore must apply conservative sampling that demonstrably prevents its utility in FR. We also contrast to LENS [26], which filters poses based on the NeRF density field. LENS' approach is prone to noise and challenges related to scene geometry (bounds, manifold, inside/outside). Furthermore, LENS constrains novel views to a plane in order to cater to its benchmarks, unlike our comparatively robust, efficient, tuning-free, and broadly applicable sample selection strategy.

We plot the impacts of filtering by each metric on the cumulative data distributions of one of the "traditional" datasets in Figure 3. Although the volume of images rejected by filtration implies its importance, our results show that, in several cases, choosing not to filter is superior (Table 2). This is due to the reduction in training data resulting from filtering and UniRepLKNet's robustness to otherwise unfiltered noisy or featureless images.

Filtering can be problematic in extreme cases. Combinations of gross extrapolation in sampling, aggressive filtration, and limited sample counts will result in very few new samples surviving the filter, producing comparable results to an unaugmented baseline. Thus, while filtering reduces the need for precise tuning, selecting reasonable sampling parameters (e.g., avoiding extrapolation far beyond a scene's rough bounding box) remains beneficial.

3.2. UniRepLKNet Pose Regression

We choose the UniRepLKNet architecture [12] as our pose regressor for its favorable performance, global receptive field, and demonstrated resilience in low-texture or patterned environments where competitors struggle. Unless otherwise noted, we apply UniRepLKNet-A, pretrained on ImageNet-1K, in all experiments. It is smaller (4.4M params) than PoseNet's GoogLeNet backbone (6.6M params), yet performs comparably. We append a regression head to predict 6-DoF pose (parametrized as XYZ WXYZ, with the quaternion explicitly normalized as a final step), and train for 100 epochs on each dataset, merging the newly sampled imagery with the provided training imagery and splitting 90:10 training to validation. We use cosine annealing [22] with a single warm restart halfway through training, and ADAM as our optimizer. Our loss function equally penalizes euclidean distance (m) and orientation error (°). We downscale images to a short-side length of 256px, then center-crop to 224^2 following [19] and due to observed insignificance of higher resolutions.

Several minor experiments were performed to inform these decisions: higher resolutions, deeper UniRepLKNet architectures (-B, -P, -N), and loading of pretrained weights each appeared not to contribute meaningfully to accuracy beyond chosen settings.

4. Experiments and Results

We report three categories of experiments: *Traditional Datasets*, in which we compare to prior work; *Synthetic Scenes*, in which we demonstrate FR in plausible drone flight scenarios and describe ablations; and *Hardware Demonstration*, in which we demonstrate efficiency

and symmetry/repetition-robust relocalization in a real-life demo.

4.1. Traditional Datasets

We present median translation and angle accuracy of comparable state-of-the-art methods on the 7Scenes and Cambridge Landmarks datasets, as reported in [40]. Each dataset provides an array of scenes with posed cameras and corresponding images, and designates standard training and testing sets. We provide dataset details including inconsequential adjustments made to the Cambridge Landmarks dataset in Appendix C in the supplementary material.

For each scene, we perform four augmentations, sampling within varying radii and perturbations of the training poses. Sampling details can be found in Appendix C in the supplementary. For each choice of sampling, we experiment with two levels of filtration as described in Section 3.1. This results in a total of eight trained models per scene. We report best results by selecting the model with lowest validation loss among each scene's augmentation-trained models. Our strategy of mixing augmented and original samples for both training and validation *succeeds* in selecting the best test-result model in most cases, or comes close.

We find that augmenting improves UniRepLKNet accuracy uniformly by at least 2x, outperforming PoseNet by 2x-4x, while remaining similarly real-time on portable hardware. Importantly, we show that UniRepLKNet *alone, without augmentation,* contributes *minimally* to the improvement over PoseNet, in some cases even underperforming as in Cambridge-KingsCollege. This cements augmentation as the driver behind the improvement in accuracy.

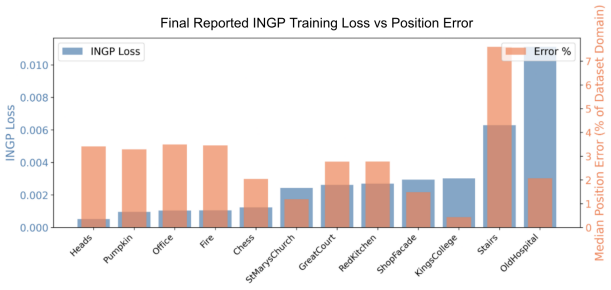


Figure 4. The final reported checkpoint loss of our various NeRFs does not correlate with final model performance, hinting at the noise-tolerance of UniRepLKNet and the success of filtering.

4.2. Synthetic Datasets

To test challenging FR situations, we produce two synthetic datasets: "Hangar", and "Dronerace", designed with repetitive features and spatially distant train/test sets (as shown in Figure 5). Further dataset details may be found in Appendix C in the supplementary material.

In the **Hangar** scene, which emphasizes challenging symmetry and reflective surfaces, our model relocalizes a simulated drone flight path with an error of only 9m/9°, despite test views being, on average, 25m from the nearest training view. DSAC* fails to converge entirely, with a catastrophic median error of 53.3°and 7km, a consequence, we hypothesize, of the patterned environment. Dust3R also fails to handle more than a small fraction of the dataset at once for pairwise localization, and must be paired with image databases and retrieval to overcome memory constraints. We conduct several ablations illustrated in Figure 6, showing that increased sample density, higher training resolution, and larger model variants did not improve performance further.

Our **Dronerace** scene is created with the intent of simulating the constraints of autonomous drone racing, an area to which our approach is well-suited. We note a recent autonomous drone racing event which featured Nvidia Orin NX GPUs on drones as the sole source of compute [1], hardware we later experiment with in Sec. 4.3. This scene, though less, repetitive, symmetrical and smaller than our Hangar dataset, features only 200 ground-truth views designed to blanket the environment from a spatially limited region and challenge existing models on sparsity. We achieve a mean translation error of 4.6m. Keeping with the theme of the dataset and intending to show the practicality of our approach, we fuse model predictions to a realistically-simulated IMU [16], showing that outlier predictions may be constrained, and median error lowered to 4.2m with an optimally-tuned Kalman filter. We again experiment with DSAC* and DFNet on this dataset. DSAC* fails to converge with 121.9°and 561.3m median errors. DFNet also succumbs to the sparse initial poses though less catastrophically as we show in Tab. 3. DFNet, even with the same far-reaching novel view-synthesis (NVS) parameters to our model and a comparable scene reconstruction quality, is unable to filter out poorly-posed views.

4.3. Hardware Demonstration

Beyond emulating the drone racing scenario through synthetic scenes, we also explore a real-life hardware relocalization demo. We capture an office environment with challenging repetitive scenery through a fisheye-lens camera with very high FOV. The office scene contains numerous repeating features including identical chairs, computers, and large identical and feature-limited cubby structures. We drive a drone [42] through the scene with the same camera, producing a ground-level test perspective.

We use Meshroom/Alicevision [17] to localize both the train and test set, each spanning around 1000 images. Notably, SfM fails on the aforementioned cubby structures, also pictured in Figure 7, and requires manual splicing and significant finetuning to recover accurate poses. We follow

Table 2. Accuracy and Compute Comparisons

	Inference Memory Usage, Latency	7Scenes							Cambridge				
		C	F	H	O	P	RK	S	GC	KC	OH	SF	SMC
PoseNet	50MB, 5-10ms Nvidia Titan Black	32, 6.60	47, 14.0	30, 12.2	48, 7.24	49, 8.12	58, 8.34	48, 13.1	N/A	166, 4.86	262, 4.90	141, 7.18	245, 7.96
DUSt3R (Best of 224/512)	12+ GB, 50-250ms H100	3, .97	3, .95	1, 1.00	3, 1.01	4, 1.14	4, 1.34	11, 2.84	36, 0.24	**11, .2**	17, 0.33	6, .26	**7, .24**
PixLoc	N/A, 88ms, RTX 2080ti	2, .80	**2, .73**	1, .82	3, .82	4, 1.21	3, 1.2	5, 1.30	30, 0.14	14, .24	16, 0.32	**5, .23**	10, .34
HSCNet++	>80MB, 85-130ms	**2, .63**	2, .79	**1, .8**	**2, .65**	**3, .85**	**3, 1.09**	**3, 0.83**	**28, .2**	19, .3	18, .3	6, .3	9, .3
DSAC*	180MB, 20-30ms RTX 4080	2, 1.10	2, 1.24	1, 1.82	3, 1.15	4, 1.34	4, 1.68	3, 1.16	49, .3	15, .3	**15, .3**	5, .3	13, .4
DFNet	100mb, **5ms**, RTX A6000	5, 1.88	17, 6.45	6, 3.63	8, 2.48	10, 2.78	22, 5.45	16, 3.29	N/A	73, 2.37	200, 2.98	67, 2.21	137, 4.03
DFNet$_{dm}$	100mb, **5ms**, RTX A6000	4, 1.48	4, 2.16	3, 1.82	7, 2.01	9, 2.26	9, 2.42	14, 3.31	N/A	43, 0.87	46, 0.87	16, 0.59	50, 1.49
UniRepLKNet*	**45MB, 5-10ms RTX4080, (20ms Orin)**	16, 5.6	42, 11.31	16, 12.86	26, 6.36	30, 5.30	31, 6.24	54, 11.58	648, 4.9	815, 6.6	249, 1.47	140, 5.74	247, 5.33
* Augmentation	**45MB, 5-10ms RTX4080, (20ms Orin)**	6, 1.80	14, 4.47	8.27, 4.07	12, 2.68	15, 2.59	16, 2.97	36, 3.25	463, 4.21	59, 0.96	111, 1.42	45, 1.48	113, 2.76
* Augmentation + Filtering	**45MB, 5-10ms RTX4080, (20ms Orin)**	8, 2.1	17, 4.96	7, 3.98	19, 3.17	14, 2.60	16, 3.03	25, 3.06	395, 4.16	66, 0.96	129, 2.24	46, 1.54	124, 2.99

Our model improves 2x-4x over PoseNet with real-time inference and a tiny memory footprint. While DSAC* and DFNet compare in efficiency, both are prone to limitations explored in Section 4.2, and we exceed the baseline DFNet without the direct-feature-matching component, indicating a plausible superiority in our augmentation and filtering scheme. Dataset scene titles are abbreviated for readability. Absolute best results in each column are bolded/underlined, while the best of *our* results are highlighted in green, and comparably worse results from competing models in red.

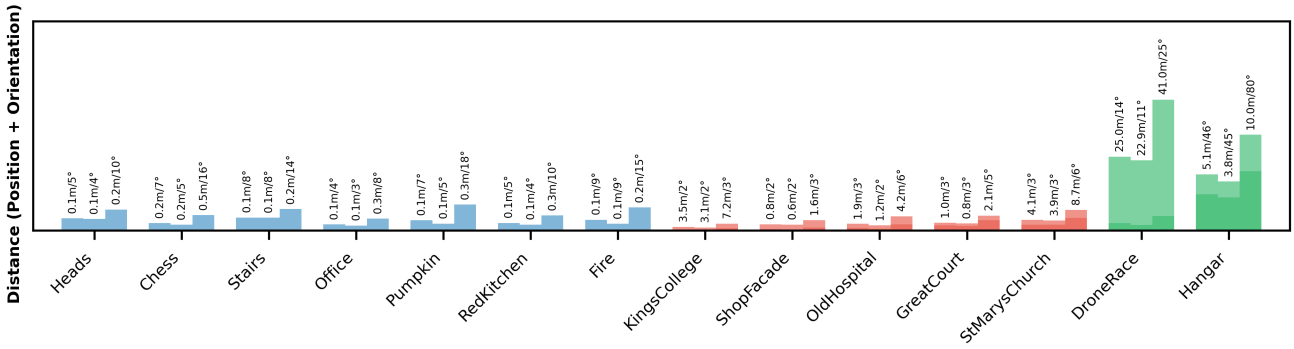


Figure 5. We show the mean, median, and 90th-percentile-highest distances to the nearest train pose per test pose of each "traditional dataset" scene, and compare to our synthetic datasets whose train/test sets differ significantly. This highlights our synthetic scene's unique exploration of faraway relocalization.

Method (NVS Perturbation)	Mapping		Inference	
	PSNR	Time	Error (m / °)	Perf.
Ours (15m + 180°)	**22.80**	**2m**	**4.09 / 12.80**	10ms / **50MB**
DFNet (15m + 180°)	22.26	3h 54m	20.27 / 136.31	**5ms** / 100MB
DFNet (3m + 7.5°)	22.26	3h 54m	19.56 / 129.66	**5ms** / 100MB
DFNet (0.2m + 10°)	22.26	3h 54m	19.47 / 137.22	**5ms** / 100MB

Table 3. DFNet trained on poses synthesized with matching parameters to our own, as well as with parameters used in its original paper for Cambridge and 7Scenes, fails to converge, revealing its inability to relocalize distantly-posed views.

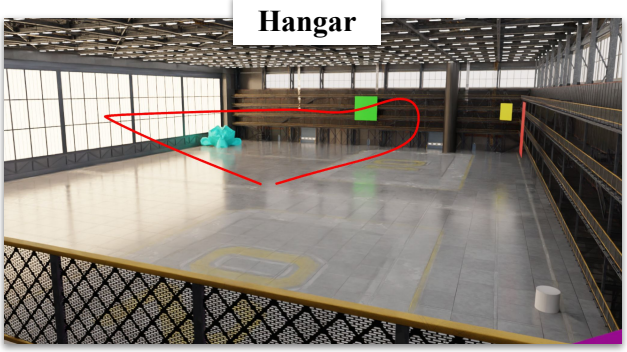
our usual procedure of NeRF training and sampling of the scene within a 1m radius of ground-truth train poses and with up to 40 degrees of perturbation from the training data, producing 10,000 additional samples.

In order to emulate real-time relocalization, we test our model on the drone's onboard computer, an Nvidia Orin NX 16GB, measuring VRAM and inference. Uncompiled, our model achieves real-time framerates of ∼50FPS (22ms), with peak VRAM utilization under 50MB. We illustrate the trajectory and report accuracies in Figure 7, highlighting our model's success in localizing ambiguous imagery in the same cubby regions that SfM failed to disambiguate. We postulate this success stems from UniRepLKNet's global receptive field, which can leverage the peripheral context of the fisheye lens used on resolutions higher than 224x224, unlike DSAC* or PoseNet.

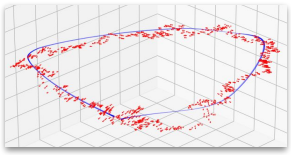
5. Ablation on Backbone

Though our experiments and other ablations (Figure 6) exclusively feature UniRepLKNet as our backbone of choice, we note our method is broadly compatible with pose regressors, generating simple labeled RGB images. However, the choice to apply UniRepLKNet is well-motivated. This architecture is known to exhibit transformer-like per-

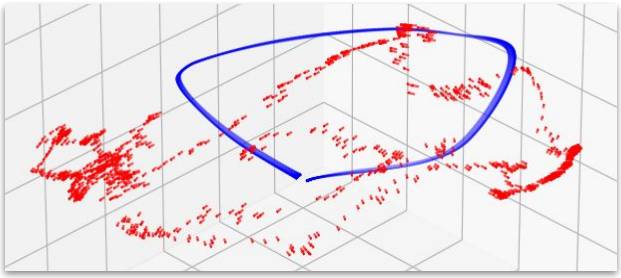
Figure 6. Synthetic Datasets



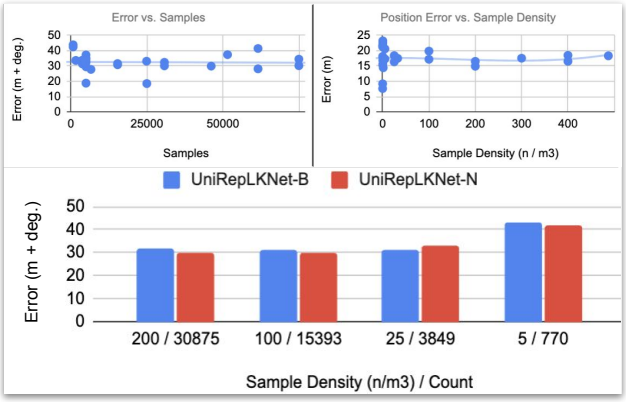
Above / right, we depict the test trajectory in red, and the train trajectory in green.



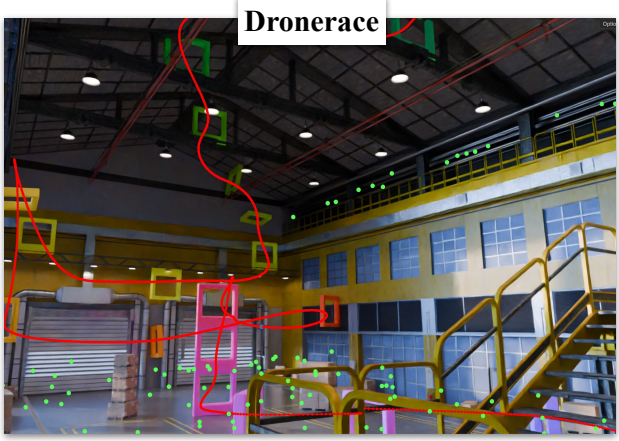
We achieve a mean accuracy of 6.68m/7.85°, via UniRepLKNet-N and 25,000 boosted samples. However, as illustrated below, various combinations of sample count, density, and model depth have little effect on outcomes. We compare against the same train/test scenario without boosting, training only from the images in the periphery of the hangar. Without boosting, our model never achieves a test error lower than 20m, averaging closer to 40m and 15 degrees of position and orientation error, clearly visible below.



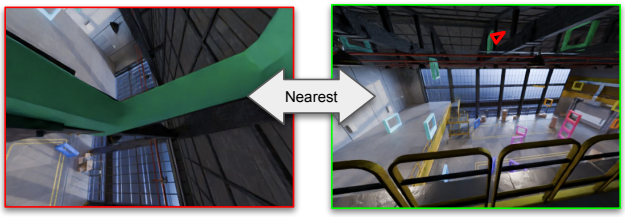
Ablations



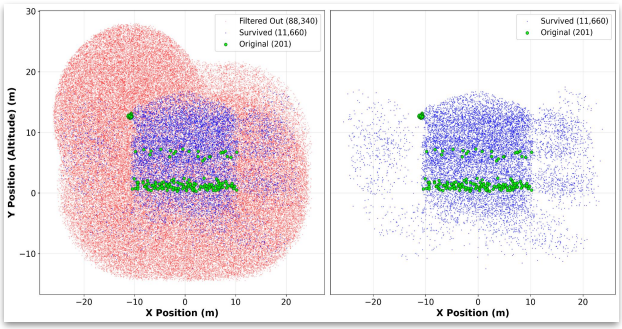
Our Hangar environment allows us to ablate on sample density (as measured out of the conveniently rectangular hangar volume) and sample count. No significant trend is noted above ~1000 boosted samples (below which performance degrades towards baseline). We also compare UniRepLKNet-B to -N. The -B variant of the architecture has 98m parameters, compared to -N's 18.3m, but no significant difference in performance is observed.



Dronerace demonstrates our model's ability to handle FR and sparse training data. We depict train poses in green and our test poses in red. An example of challenging faraway relocalization is shown below.

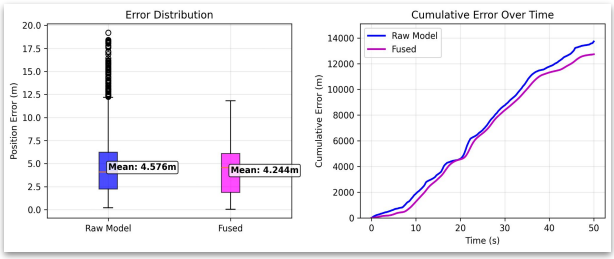


Our dronerace scene also helps illustrate the ***importance of filtering*** as deduced from the structure and gross reduction in samples depicted below:

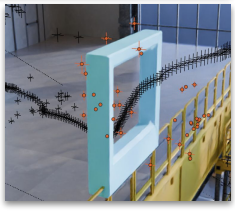


Simulated Sensor Fusion

We show that our model is suitable for grounding of inertial localization by fusing it to a simulated IMU via optimally-tuned Kalman filter.



Sensor fusion reduces our model error, especially among outliers. We depict the fused trajectory (black) intersecting a drone racing checkpoint, and the noisy model predictions in its vicinity (orange). This is one of the few checkpoints intersected by our model, hinting at the difficulty and credibility of the FR challenge we introduce.



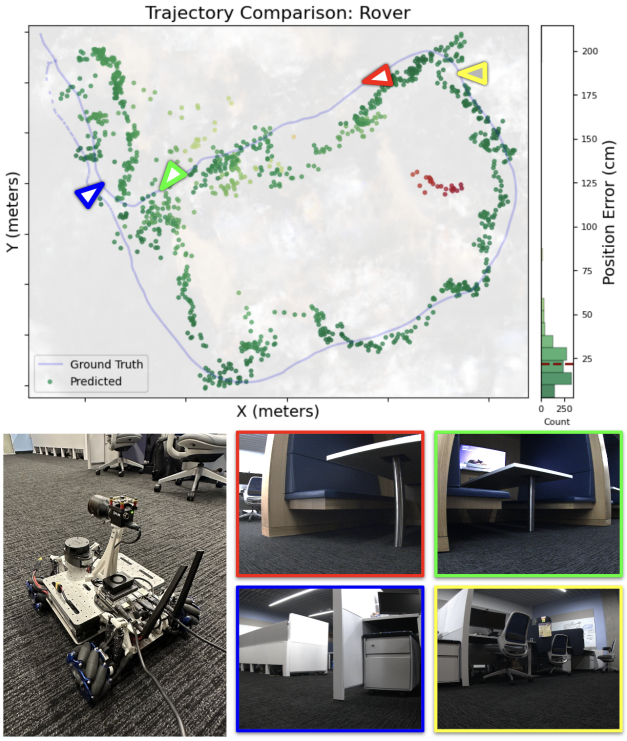


Figure 7. We visualize predictions in our hardware demo, emphasizing the red/green and blue/yellow cameras/views which showcase the successful relocalization of two nearly identical but different regions. We demonstrate localization that handles these repetitive regions well, with a respectable median error of 20cm / 6°, and limited hyperparameter tuning. Depicted LIDAR was unused.

formance, such as a need for larger volumes of data, a global receptive field, and superior performance per compute. To validate our decision, we replace UniRepLKNet with two backbones - an attempted reimplementation of CoordiNet [25], a closed-source model used in [26], and ResNet-152 [18], as provided by PyTorch [29]. As results in Tab. 4 demonstrate, our model is compatible with the technically complex CoordiNet pose regressor implementation, as well as the baseline ResNet-152. Critically, despite being the lightest, UniRepLKNet-A achieves the best performance improvement from our augmentation method, exceeding that which the LENS paper reports on its own closed implementation of CoordiNet.

6. Limitations and Conclusion

We introduced Augmenting with NeRFs, a method that improves the accuracy of lightweight, single-shot relocalization models. Our approach sidesteps many limitations of existing methods, especially an inability to localize familiar regions from spatially novel perspectives, by augmenting relocalization datasets with many additional, often-extrapolative samples. We show that augmenting reduces

Table 4. Alternative Regressors

Method	AVG, Median Error	Relative Error After Augmentation	Params, Mem.
CoordiNet (Paper)	92, 2.6	0.43x, 0.45x	N/A, N/A
CoordiNet + LENS (Paper)	**39, 1.2**		
Our "CoordiNet"	203, 4.5	0.47x, **0.39x**	11.5m, 61mb
+ Our Augmentation	100, 1.8		
Our ResNet-152	268, 5.4	0.47x, **0.39x**	62m, 249mb
+ Our Augmentation	127, 2.2		
Our UniRepLKNet-A	363, 4.8	**0.25x**, **0.39x**	**6.7m, 45mb**
+ Our Augmentation	91, 1.9		

Best results bolded, green shows best result of ***our*** experimental backbones, relative to inferior ablations highlighted in red.

error in UniRepLKNet by as much as 70% compared to an unaugmented baseline across many datasets. We also qualitatively demonstrate that the wholly-encompassing receptive field of UniRepLKNet, combined with wide-angle cameras, can handle repetitive structures and integrate peripheral details into predictions where even classical SfM fails, making it well-suited to relocalization. In synthetic scenes with faraway-relocalization objectives, our approach stands alone in accuracy and robustness.

Our method has several limitations. First is reliance on often-brittle SfM pipelines for otherwise-unposed ground-truth, a recurring theme in relocalization works. Second, our sample placement is a brute-force inefficiency—rendering 100,000 views to discard 90,000—that could be reconciled by NeRF-uncertainty-aware sampling (e.g., Bayes Rays [15]), but still challenges related papers [7, 28]. Future work may look to the strategy SPPNet exploits on feature point clouds to more efficiently mine new poses, though this too requires ample pruning [30]. This inefficiency is mitigated by our model's compatibility with hash-encoding-accelerated NeRF backbones like Instant-NGP [27] and its successors, which enable us to trivially generate such vast amounts of synthetic views (see appendix in supplementary material for training times). Third, we introduce new hyperparameters (sampling radius/perturbation) which, while controlled by filtering, may still require tuning and can modestly impact results. Finally, VRAM limitations prevent us from exploring resolutions higher than 512x512, or architectures deeper than UniRepLKNet-B. Further ablations and VRAM-unlimited explorations may reveal desirable compute-performance tradeoffs or a more precise lower bound for sample count.

Despite these limitations, pushing NeRF augmentation orders of magnitude higher is a promising direction for the relocalization field that may be used to complement the training of ***many*** other approaches which innovate on pose regressors. We hope our work brings attention to simpler parsimonious architectures and lightweight relocalization methods that improve relocalization on portable robotics like FPV drones, AR/VR, and mobile devices.

Acknowledgements

The authors gratefully acknowledge Davide Tirindelli for his hangar environment art, and the partial support by the National Science Foundation (NSF) grants 1942444 and 2330416 and the Office of Naval Research (ONR) grants N000142312429 and N000142312363.

References

[1] Abu Dhabi Autonomous Racing League (A2RL). A2RL Technical Specifications. https://a2rl.io/press-release/9/artificial-intelligence-triumphs-in-worlds-most-sophisticated-autonomous-drone-race-in-abu-dhabi, 2025. 5

[2] Eric Brachmann, Tommaso Cavallari, and Victor Adrian Prisacariu. Accelerated coordinate encoding: Learning to relocalize in minutes using rgb and poses. In *CVPR*, 2023. 2, 3

[3] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. DSAC-Differentiable RANSAC for camera localization. In *CVPR*, 2017. 2, 3

[4] Eric Brachmann and Carsten Rother. Learning less is more - 6D camera localization via 3D surface regression. In *CVPR*, 2018. 2

[5] Eric Brachmann and Carsten Rother. Visual camera relocalization from RGB and RGB-D images using DSAC. *TPAMI*, 2021. 2, 3

[6] Eric Brachmann, Jamie Wynn, Shuai Chen, Tommaso Cavallari, Áron Monszpart, Daniyar Turmukhambetov, and Victor Adrian Prisacariu. Scene coordinate reconstruction: Posing of image collections via incremental learning of a relocalizer. In *ECCV*, 2024. 2, 3

[7] Le Chen, Weirong Chen, Rui Wang, and Marc Pollefeys. Leveraging neural radiance fields for uncertainty-aware visual localization, 2023. 8

[8] Shuai Chen, Yash Bhalgat, Xinghui Li, Jiawang Bian, Kejie Li, Zirui Wang, and Victor Adrian Prisacariu. Neural refinement for absolute pose regression with feature synthesis, 2024. 3

[9] Shuai Chen, Tommaso Cavallari, Victor Adrian Prisacariu, and Eric Brachmann. Map-relative pose regression for visual re-localization. In *CVPR*, 2024. 3

[10] Shuai Chen, Xinghui Li, Zirui Wang, and Victor Prisacariu. Dfnet: Enhance absolute pose regression with direct feature matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 3, 4

[11] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. 3

[12] Xiaohan Ding, Yiyuan Zhang, Yixiao Ge, Sijie Zhao, Lin Song, Xiangyu Yue, and Ying Shan. Unireplknet: A universal perception large-kernel convnet for audio, video, point cloud, time-series and image recognition, 2024. 1, 4

[13] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981. 2

[14] Ben Glocker, Shahram Izadi, Jamie Shotton, and Antonio Criminisi. Real-time rgb-d camera relocalization. In *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 173–179, 2013. 1

[15] Lily Goli, Cody Reading, Silvia Sellán, Alec Jacobson, and Andrea Tagliasacchi. Bayes' Rays: Uncertainty quantification in neural radiance fields. *CVPR*, 2024. 8

[16] Rodrigo Gonzalez and Paolo Dabove. Performance Assessment of an Ultra Low-Cost Inertial Measurement Unit for Ground Vehicle Navigation. *Sensors (Basel)*, 19(18):3865, Sep 2019. 5

[17] Carsten Griwodz, Simone Gasparini, Lilian Calvet, Pierre Gurdjos, Fabien Castan, Benoit Maujean, Gregoire De Lillo, and Yann Lanthony. Alicevision meshroom: An open-source 3d reconstruction pipeline. In *Proceedings of the 12th ACM Multimedia Systems Conference*, MMSys '21, page 241–247, New York, NY, USA, 2021. Association for Computing Machinery. 2, 3, 5

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 8

[19] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization, 2016. 1, 2, 4

[20] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r, 2024. 2

[21] Jianlin Liu, Qiang Nie, Yong Liu, and Chengjie Wang. Nerf-loc: Visual localization with conditional neural radiance field, 2023. 2

[22] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts, 2017. 4

[23] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 3

[24] Anish Mittal, Anush Krishna Moorthy, and Alan Conrad Bovik. No-reference image quality assessment in the spatial domain. *IEEE Transactions on Image Processing*, 21(12):4695–4708, 2012. 4

[25] Arthur Moreau, Nathan Piasco, Dzmitry Tsishkou, Bogdan Stanciulescu, and Arnaud de La Fortelle. Coordinet: uncertainty-aware pose regressor for reliable vehicle localization, 2021. 3, 8

[26] Arthur Moreau, Nathan Piasco, Dzmitry Tsishkou, Bogdan Stanciulescu, and Arnaud de La Fortelle. Lens: Localization enhanced by nerf synthesis, 2021. 3, 4, 8

[27] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, July 2022. 1, 3, 8

[28] Xuran Pan, Zihang Lai, Shiji Song, and Gao Huang. Activenerf: Learning where to see with uncertainty estimation, 2022. 8

[29] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison,

Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019. 8

[30] Pulak Purkait, Cheng Zhao, and Christopher Zach. Spp-net: Deep absolute pose regression with synthetic views, 2017. 3, 8

[31] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019. 2

[32] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[33] Rashik Shrestha, Bishad Koju, Abhigyan Bhusal, Danda Pani Paudel, and François Rameau. Caldiff: Camera localization in nerf via pose diffusion, 2023. 3

[34] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions, 2014. 2

[35] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. Inloc: Indoor visual localization with dense matching and view synthesis, 2018. 2

[36] Shitao Tang, Sicong Tang, Andrea Tagliasacchi, Ping Tan, and Yasutaka Furukawa. Neumap: Neural coordinate mapping by auto-transdecoder for camera localization, 2023. 2

[37] Jianyi Wang, Kelvin CK Chan, and Chen Change Loy. Exploring clip for assessing the look and feel of images. In *AAAI*, 2023. 4

[38] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer, 2025. 2

[39] Jialu Wang, Kaichen Zhou, Andrew Markham, and Niki Trigoni. Wscloc: Weakly-supervised sparse-view camera relocalization, 2024. 3

[40] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy, 2024. 2, 3, 5

[41] Lin Yen-Chen, Pete Florence, Jonathan T. Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. Inerf: Inverting neural radiance fields for pose estimation, 2021. 3

[42] Yuxuan Zhang, Adnan Abdullah, Sanjeev J. Koppal, and Md Jahidul Islam. Cliprover: Zero-shot vision-language exploration and target discovery by mobile robots, 2025. 5

[43] Boming Zhao, Luwei Yang, Mao Mao, Hujun Bao, and Zhaopeng Cui. Pnerfloc: Visual localization with point-based neural radiance fields. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 7450–7459, 2024. 3

[44] Qunjie Zhou, Maxim Maximov, Or Litany, and Laura Leal-Taixé. The nerfect match: Exploring nerf features for visual localization, 2024. 3