BARFT: Bundle-Adjusting Neural Radiance Fields with Temporal Regularization

Ben Agro Quin Sykora Sourav Biswas Yiqian Qin

Abstract

Neural Radiance Fields (NeRFs) [13] and their various derivative works have become the de facto solution for 3D scene representation from RGB images. One limitation of NeRf is the requirement of accurate camera poses to learn the scene representation. Recently, Lin et al. proposed Bundle Adjusting Neural Radiance Fields (BARF) [12] to learn the scene representation and camera poses jointly; however, they only require minimal viewpoint changes or good camera pose initialization. To address these problems, we propose BARFT*; a method for training a NeRF on a chronological sequence of images (such as a video) without any known camera poses. We leverage the temporal information between consecutive frames to regularize the learned poses, allowing* BARFT *to learn accurate poses even with significant viewpoint changes and without any initialization. Our experiments show that* BARFT *outperforms the state-of-the-art BARF at learning a NeRF without camera poses, furthering the line of work towards learned visual localization systems and providing a robust and flexible alternative for training a NeRF on video data.* [1]

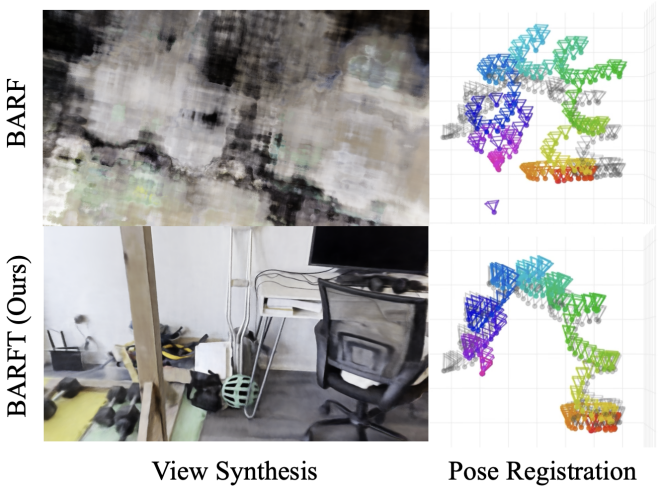

Figure 1. Novel view synthesis using BARF and BARFT, and learned camera poses (ground truth in grey). Leveraging Temporal information allows us to optimize the NeRF and camera poses jointly without requiring initialization.

1. Introduction

Neural Radiance Fields (NeRFs) [13] have proven to be a popular and powerful representation of 3D space. A neural network mapping from a 3D position and viewpoint direction to color and opacity can represent a 3D scene and be used to render novel viewpoints. The network is trained by comparing rendered images at given camera poses to their associated ground truth images, which encodes the scene in the network weights. NeRFs allow for high-fidelity reconstruction of the scene from multiple viewpoints.

NeRFs need accurate camera extrinsics (poses) for each image, a requirement which limits their widespread application [13]. The camera poses are either known for synthetic datasets or estimated in a pre-processing step using structure-from-motion (SfM) or Simultaneous Localization and Mapping (SLAM) methods [13] e.g., COLMAP [20, 21]. These methods are noise-sensitive, can produce sub-optimal solutions, and can be slow and cumbersome for use online [12].

Various works tackle NeRFs unknown camera poses, but either require initial pose estimates [12, 24], only work on scenes with minimal pose changes (commonly using a forward-facing camera with only planar translational movements) [12, 26], or they require depth camera measurements [9].

These shortcomings motivate our approach, BARFT, which leverages the temporal information inherent to RGB video data to optimize a NeRF and the camera poses jointly, with the ability to handle potentially complex camera movements (e.g., rotations, orbits, depth changes). The idea is simple and elegant: BARF [12] showed how to optimize the NeRF and camera poses jointly, but this only works with minimal viewpoint changes. However, considering a contiguous set of video frames sufficiently close in time, they

[1] Code Available at: https://github.com/souravb111/CSC2530GroupProject

will have minimal viewpoint changes. Thus, we design an architecture and training procedure that first learns the camera poses and updates the NeRF from only a short contiguous snippet of video that is close in time to build an initial "map" of the 3D scene and estimates of the camera poses. This initial NeRF can be used to localize subsequent camera poses that are nearby in time, and the NeRF will, in turn, be updated with the new RGB image observations. We can chain this procedure to learn the NeRF and camera poses for an entire RGB video. We perform extensive experiments on real-world video data with complicated camera movements, which shows that our method can jointly optimize the NeRF and camera poses and vastly outperforms the state-of-the-art BARF [12].

2. Related Work

Structure from Motion and SLAM Given a set of input images, SfM [1,18,19,22,23,27] and SLAM [4,7,15,16,28] systems aim to recover a 3D map and sensor poses simultaneously. SLAM systems are designed to run online, while SfM focuses on reconstruction quality. *Feature-based / Indirect* methods [4, 15] detect key-points and match them across frames. Modern versions use pre-trained neural networks as feature detectors [5]. *Dense/ Direct* methods [2,6] exploit photo-metric consistency of pixel intensities between adjacent images to estimate relative poses.

Our task falls under SfM, but in contrast to classical methods, we aim to use a learning-based optimization approach, encoding the scene with neural networks to obtain high-fidelity 3D reconstructions. BARFT is part of a recent line of work on the exciting avenue of rethinking visual localization for SfM/SLAM systems using view synthesis as a proxy objective.

NeRFs [13] proposed NeRF to synthesize novel views of static scenes given posed RGB images. A multi-layer perceptron (MLP) —— with inputs x, y, z, view direction and outputs of RGB color and opacity — is used to represent the scene. Images are rendered using this MLP by accumulating color and opacity along pixel rays, and the network is trained via a photometric loss between the given image and the rendered image.

NeRFs with Unknown Camera Parameters One line of work in NeRFs has been to relax the assumption of accurate camera poses for each image. Simultaneously reconstructing the 3D scene and the camera poses is a classic chicken-and-egg problem.

Inspired by image alignment, BARF [12] proposes a novel schedule for coarse-to-fine positional encoding throughout training. A key to the high-fidelity reconstructions of NeRF is its use of positional encoding, a deterministic mapping of input 3D coordinates to higher dimensions with different sinusoidal frequency bases [13]. However, the high-frequency bands of the positional encoding result in incoherent gradients on the query positions (x, y, z) and thus make it challenging to optimize for the camera poses. Thus, BARF proposed to apply a smooth mask on the encoding (from low to high) throughout training and showed that this allowed for the joint optimization of camera poses and NeRF [12]. However, BARF only performs well when the input images have minimal viewpoint changes (e.g., only forward-facing planar camera movement). If the viewpoint changes are complex, BARF requires fairly accurate pose initialization (e.g., from an SfM system). When these conditions do not hold, BARF does not learn accurate camera poses and instead "cheats" by learning to reproduce the RGB training images without learning the underlying 3D structure of the world (which we show in Sec. 4). Other recent works like NeRF– [26] and BADNeRF [24] face similar limitations.

Gaussian Splatting Kerbl et al. recently proposed Gaussian Splatting [10] as a promising alternative to NeRF for novel view synthesis and 3D scene reconstruction. The idea is to optimize the size, position, color, and opacity of millions of Gaussians in the scene. The Gaussians can be efficiently rendered by "splatting" them onto the image plane. Gaussian Splatting allows for super-realtime rendering (400FPS) [9], faster training, and more accurate novel view synthesis. The Gaussian positions can be randomly initialized or leverage point-cloud locations from SfM or a depth camera. This Gaussian representation provides an explicit map with spatial extent, which can be used to determine which parts of the rendered images are inaccurate and mask them out in the reconstruction loss [9]. SplaTAM [9] leverages these advantages to use Gaussian Splatting as a 3D map representation for online SLAM. However, they assume access to a depth camera to initialize the positions of the Gaussians.

3. Method

We begin with the intuition behind our method. The main idea is that BARF [12] cannot learn with images that have very different viewpoints (e.g., little to no overlapping features, large rotations, significant depth changes). However, in a video sequence, we know that nearby images in time should be similar to one another (if using a sufficiently high frame rate).

3.1. Overview

Our method uses an RGB video with N total frames ordered in time and known camera intrinsic parameters as input. Optimization of the NeRF and camera poses occurs over three stages, illustrated in Fig. 2, which we discuss below.

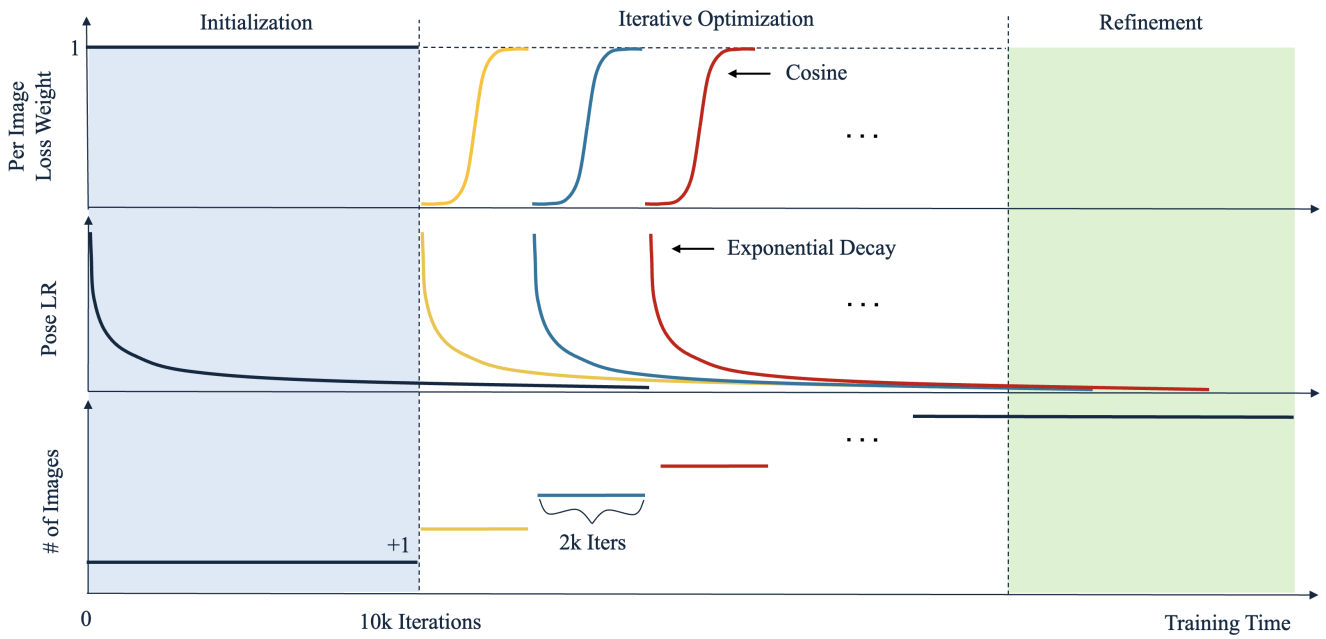

Figure 2. **BARFT Training Procedure**: The overall training procedure has three stages. In the *Initialization* stage, we use a short snippet from the beginning of the video to train both the NERF model and camera poses. We gradually add consecutive frames to the training data in the *Iterative Optimization* stage. Upon adding each new frame, we initialize the new camera pose to that of the previous frame, and its learning rate follows exponential decay. The loss weighting for the new image is smoothly increased over 2k iterations using a cosine scheduling function. In the *Refinement* stage, we train the entire model with all camera poses and video frames. Here, we gradually unmask higher frequency bands of the positional encoding to allow for high-fidelity NeRF results.

Initialization For `init_num_iters`, we optimize the NeRF using only the first `init_num_images` frames (e.g., using the first 1s of frames from a video for 10k iterations). We call camera poses that are being optimized and their associated RGB frames "active set". Here, only the lowest frequency band of the positional encoding is unmasked to make learning camera poses easier, as in BARF [12]. Because those images are close in time, we can learn both the NeRF and camera poses for the scene area observed in the initialization images [12].

Iterative Optimization After the *Initialization* period, we introduce new images into the active set chronologically. Concretely, we add the next video frame with camera pose initialized to that of the previous frame and optimize the NeRF and this new active set for `per_image_num_iters` (e.g., 2000 iterations). The idea here is that this new image is close in time to the pre-existing active set and thus will have a significant viewpoint overlap with the existing NeRF, and its camera pose should be close to the previous camera pose in the video sequence. This significant overlap makes it easy to localize the new camera pose with respect to the existing NeRF and update the NeRF to account for the new video frame observation. We make three additional changes from BARF to facilitate this optimization procedure.

BARF uses an exponential learning rate decay for the poses throughout training, allowing for large updates initially when the poses have significant errors and smooth convergence to a minimum later in training. Instead, we use an independent learning rate schedule for each pose because we introduce camera poses to the active set at different times during training. We choose an exponentially decaying learning rate that starts when the corresponding image is added to the active set (see Fig. 2). This learning rate schedule allows for large camera pose updates when the images are first introduced to the active set and only minor refinements later.

Secondly, introducing a new image initialized with the "wrong" pose can make it difficult to optimize the NeRF, as the new observation may disagree largely with the existing NeRF. To smooth the introduction of new images, we gradually increase their contribution to the overall loss using a cosine schedule increasing from a loss weighting of 0 to a loss weighting of 1 over `duration_increase` (we set this to be equal to `per_image_num_iters`) (see Fig. 2).

Refinement Finally, we include a *Refinement* period at the end of training that is `finalize_num_iters` (e.g., 200k) iterations in duration beginning after the active set includes all RGB frames. During this period, we unmask the NeRF

positional encoding as described by BARF [12]. This late unmasking allows the optimization to prioritize learning accurate poses while the active set grows and then transition to learning a high-fidelity NeRF during the refinement stage. We note that we restart the NeRF (exponential) learning rate schedule during the refinement phase to accommodate changes in positional encoding, which may require significant parameter updates.

3.2. Implementation Details

For BARFT, we train:

- *Initialization* for 10k iterations.
- *Iterative Optimization* for 2k steps per image. During this stage, for each new video frame, we exponentially decay from 3×10^{-4} pose learning rate exponentially decaying to 1×10^{-6}.
- *Refinement* for 200k iterations. During this period, we linearly unmask the positional encoding from $10\% = 20000$ iterations to $50\% = 100000$ iterations in the refinement stage, as in BARF [12].

The learning rate for the NeRF is exponentially decaying from 5×10^{-4} to 1×10^{-4} over *Initialization* and *Iterative Optimization*, and then re-started again from 5×10^{-4} to 1×10^{-4} over the *Refinement* stage.

We follow implementation details used in [12] for BARF and NeRF; we train them for 200k iterations with the NeRF learning rate exponentially decaying from 5×10^{-4} to 1×10^{-4}. In the case of BARF, the learning rate for the camera pose exponentially decays from 1×10^{-3} to 1×10^{-5} throughout training, and the positional encoding is unmasked during the period of 10% of training to 50% of training.

We train all methods on 1 GPU with 2048 random rays sampled per image.

4. Experiments

We test the performance of our method on three real-world videos captured with a handheld cellphone (see Sec. 4.1). More specifically, we evaluate our method on each video in two aspects: (i) accuracy for camera pose registration and (ii) view synthesis quality for the 3D scene representation.

4.1. Datasets

We conduct our experiments on three videos: WORKOUTAREA, PLANT, DESK. See Figs. 3 to 6 for an idea of the content of these videos. Each video is 30 frames per second, roughly one minute long, and has a 1080 by 1920 pixel resolution. The videos were collected manually using an iPhone 13 Pro video camera, and we estimated ground truth poses using COLMAP [20]. Semantically, each video captures different aspects of evaluation. The WORKOUTAREA dataset has significant translations during video capture and no overlap between the final and initial images. The PLANT dataset has a partial orbital rotation around a subject of interest, featuring camera rotations and novel viewpoints that BARF [12] cannot handle. DESK features significant depth changes throughout the video. We use video frames at times $\{0\,\text{s}, 0.5\,\text{s}, 1\,\text{s}, \dots\}$ for the training images, and video frames at $\{0.25\,\text{s}, 0.75\,\text{s}, 1.25\,\text{s}, \dots\}$ for the evaluation images, ensuring the two sets are non-overlapping but from a similar distribution.

4.2. Metrics and Evaluation

We measure performance along two axes: camera registration accuracy and view synthesis quality.

Camera Registration Following BARF [12], since the learned camera poses are variable up to a 3D similarity transform, we evaluate the registration quality by pre-aligning the optimized poses to the ground truth with Procrustes analysis on the camera locations. We use average translation errors (in meters) and rotation errors (in degrees). See appendix B of BARF [12] for more details on how to compute average rotation error. Note that the "ground truth" camera poses are provided by COLMAP [20] and are thus not perfectly accurate. Nevertheless, it provides some idea of the relative performance of BARF and BARFT.

View Synthesis Quality To evaluate the quality of novel view synthesis, we transform the test views to the coordinate system of the optimized poses by applying the 3D similarity transform. We follow BARF [12] in factoring out the pose error in evaluating view synthesis quality by running an additional step of test-time photometric optimization. Concretely, we learn a pose offset from the evaluation pose and render the novel view RGB image at this offset pose. The metrics used for evaluating view synthesis quality are:

- **Peak signal-to-noise ratio (PSNR):** Measures the mean-squared error per pixel between original and reconstructed images [8]. Higher PSNR implies less difference between the original and reconstructed image.
- **Similarity Index Method (SSIM):** Evaluates local structural similarity between images [25]. Higher SSIM values indicate higher similarity between original and reconstructed images.
- **Learned Perceptual Image Patch Similarity (LPIPS):** Evaluates deep features across different architectures and reflects the perceptual similarity between images [29]. Lower values correspond to higher perceptual similarity.

Scene	Camera Pose Registration				View Synthesis Quality								
	Rotation Error (°) ↓		Translation Error (m) ↓		PSNR ↑			SSIM ↑			LPIPS ↓		
	BARF	BARFT	BARF	BARFT	BARF	BARFT	NeRF	BARF	BARFT	NeRF	BARF	BARFT	NeRF
WORKOUTAREA	174.2	**6.55**	5.66	**0.47**	7.35	**21.66**	26.42	0.40	**0.78**	0.83	0.948	**0.390**	0.350
PLANT	178.1	**7.52**	6.09	**0.50**	10.49	**21.70**	26.78	0.58	**0.80**	0.83	0.884	**0.360**	0.364
DESK	24.6	**3.71**	1.38	**0.23**	9.51	**23.02**	25.34	0.53	**0.77**	0.79	0.820	**0.410**	0.420

Table 1. Camera pose registration accuracy and novel view synthesis evaluations of our model compared to baseline BARF. We include NeRF for reference, but a direct comparison to BARFT or BARF should not be made because NeRF has access to ground truth camera poses during training.

4.3. Quantitative Results

Comparison Against Baselines We present our quantitative results in Table 1, which compares BARF, BARFT (ours), and NeRF on the three datasets. Note that we present NeRF as a reference point for the "best" view synthesis metrics that could be achieved and should not be compared against directly because it has access to the "ground truth" poses.

Overall, our method dramatically outperforms BARF across all datasets and metrics and approaches the view synthesis quality of NeRF. We note that when presented with camera poses with significant viewpoint changes, BARF does not learn meaningful 3D structure or camera poses and instead overfits to the training images. This overfitting is evident in Fig. 4 and Fig. 5, where the novel view depth and RGB images and the learned poses are nonsensical. BARFT is robust to challenging video data, including orbital rotations, extensive translations, and significant depth changes.

Ablations We conduct ablation studies to analyze the impact of the per-image camera pose learning rate and final *Refinement* stage proposed in the Sec. 3. We use the PLANT dataset The results of our ablation experiments are shown in Tab. 2.

For ablating the proposed per-image camera pose learning rate schedule, we instead use the shared exponentially decaying camera pose learning rate used by BARF [12]. The proposed schedule is crucial for performance. Without it, the camera poses introduced later in training cannot be updated by the large amounts required to align with the learned NeRF. As seen with BARF, these inaccurate camera poses result in the training collapsing and the model overfitting without learning 3D structure.

For ablating the *Refinement* stage, we instead follow BARF in un-masking the positional encoding linearly during the period of 10% to 50% of total training iterations (instead of 10% to 50% of *Refinement* stage iterations). This earlier introduction of the high-frequency bands of the positional encoding causes issues in learning the camera poses, as illustrated by the worse pose registration results in Tab. 2, which in turn affects the view synthesis quality.

To ablate the proposed pose loss weighting schedule, we instead give every image an equal loss weighting. As shown in Tab. 2, our proposed loss schedule brings slight performance improvements on the PLANT dataset. In datasets with more considerable viewpoint changes between adjacent frames, we would expect this improvement to be even more prominent; newly introduced images would have considerable disagreements with the existing NeRF as their initial camera pose is inaccurate, which could cause inaccuracies to be learned in the NeRF if that image is weighted equally in the image reconstruction loss.

4.4. Qualitative Results

We provide some qualitative comparisons of BARFT to BARF and the oracle NeRF in Fig. 3. In all datasets, we observe BARF's tendency to overfit the training images and not learn the 3D structure of the world or accurate camera poses. This overfitting is reflected in the nonsensical RGB output of BARF when evaluated at the novel evaluation view and inaccurate depth map. On the other hand, BARFT learns an accurate depth map (comparable to that of NeRF) and high-quality RGB novel views.

Figs. 4 to 6 illustrate the poses learned by BARF and BARFT compared to the "ground truth" poses provided by COLMAP [20]. We see that throughout training BARFT learns the camera poses as the active set grows, while BARF learns nonsensical camera poses.

5. Future Directions

Our method is less accurate for camera pose registration than sophisticated SLAM and SfM methods. Future work should investigate how to bridge this gap. We see two potential areas to investigate.

Explicit Map: The learned NeRF is a learned map in some sense, but it does not have an explicit spatial frontier. Thus, we do not consider if a newly introduced image is looking at a region that has already been mapped or at a newly observed region in space, which results in ambiguity as to whether or not the map/NeRF should be updated or the camera pose. If the spatial frontier of the map was known, then when adding an image to the active set,

			Camera Pose Registration		View Synthesis Quality		
Pose LR	Refinement	Loss Schedule	Rotation Error (°) ↓	Translation Error (m) ↓	PSNR ↑	SSIM ↑	LPIPS ↓
✗	✓	✓	159.37	7.18	7.41	0.39	0.94
✓	✗	✓	16.04	0.91	15.35	0.70	0.57
✓	✓	✗	8.62	0.55	18.39	0.75	0.46
✓	✓	✓	**7.52**	**0.50**	**21.70**	**0.80**	**0.36**

Table 2. Training subsection ablation: we ablate our proposed pose learning rate schedule, refinement stage, and loss weight schedule described in Sec. 3 on the PLANT dataset. All components are important for final performance in both camera pose registration and view synthesis quality.

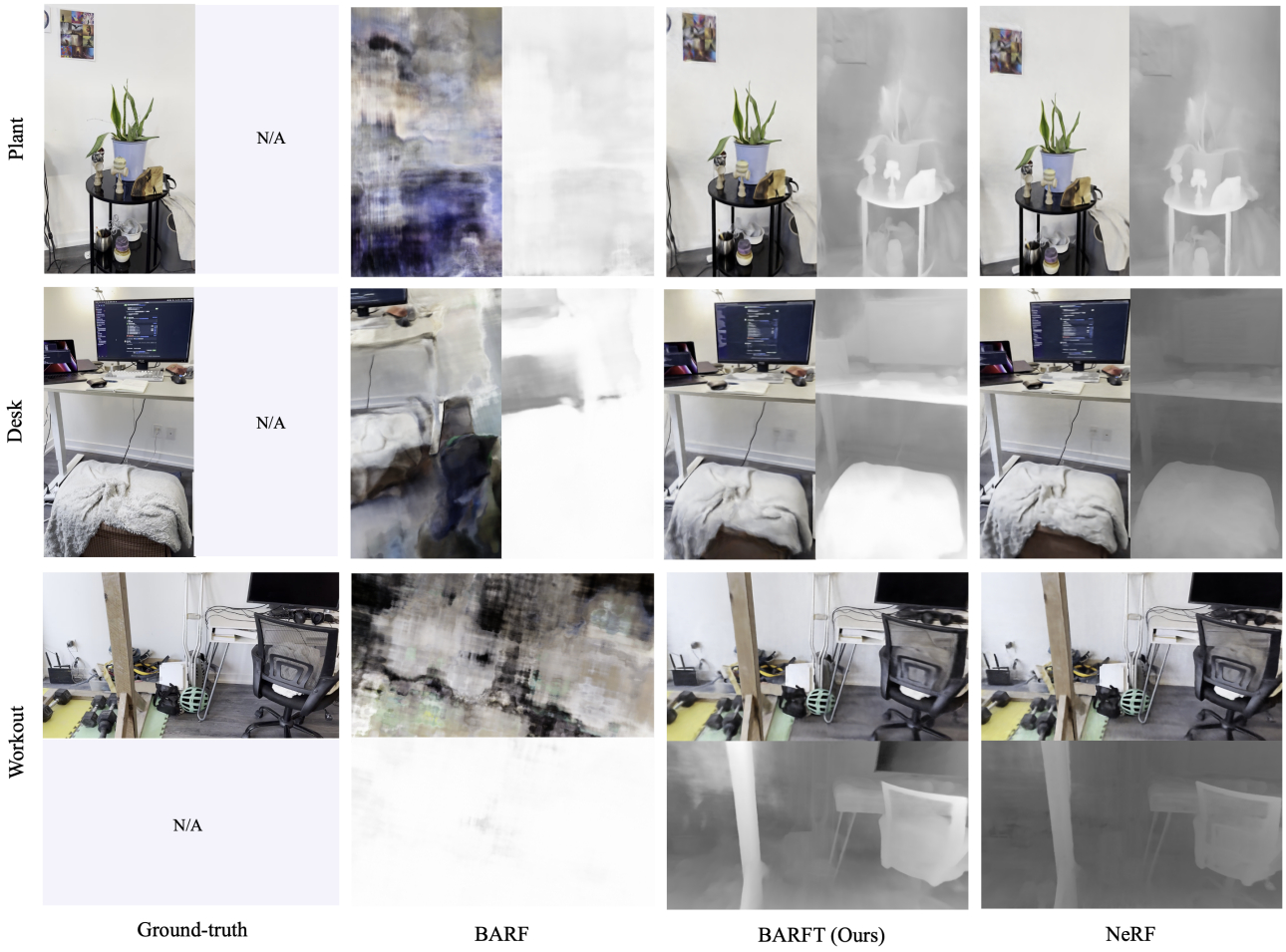

Figure 3. Novel view synthesis qualitative comparison. BARFT can produce reconstructions of the scene comparable to NERF trained with pre-computed camera positions. On the other hand, BARF struggles to learn the camera poses and NeRF jointly and fails to create a realistic 3D reconstruction.

we could use mapped regions to optimize the camera pose while filling in unmapped areas with the latest observation. As in SplaTAM [9] and discussed in Sec. 2, Gaussian splatting presents a promising alternative to NeRFs that has explicit spatial extent and could be used for learned SLAM.

Efficiency and Realtime Operation: Implemented entirely in Python, optimizing the NeRF and camera poses is far too slow to run online (approximately 8 hours of training time on an A6000 GPU). However, InstantNGP [14] showed they could speed up NeRF training by orders of magnitude. A similar approach could be applied to our method, which, along with other optimizations, could allow for realtime camera pose registration from a video stream.

Improved Motion Model: In our method, we naively initialize the newly added camera pose to the previous camera pose in the sequence. Future work could investigate

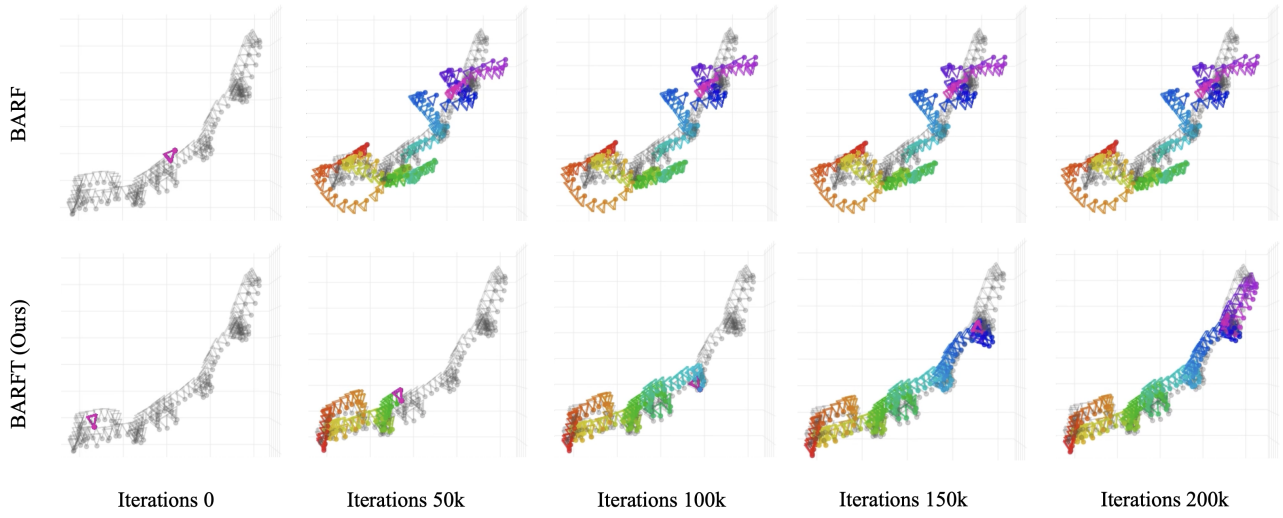

Figure 4. Camera Pose Registration for the PLANT dataset. The grey poses are the "ground truth", while the colored poses are learned. Despite the length of the camera trajectory, BARFT can learn a trajectory that very closely follows the true path.

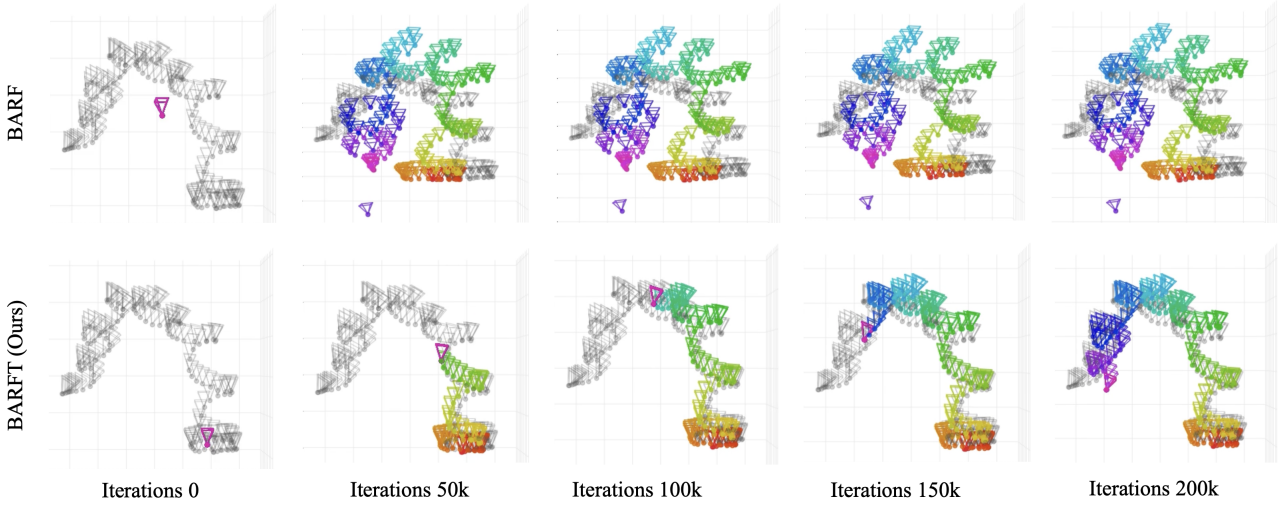

Figure 5. Camera Pose Registration for the DESK dataset. The grey poses are the "ground truth", while the colored poses are learned. BARFT accumulates significantly less error than BARF.

better initializations using a motion model (e.g., constant velocity), which could speed up optimization and improve performance by converging to a better minimum.

Loop closure: Mature SLAM systems often include a loop-closure mechanism, which allows for drastic map and pose updates when the same part of the map is observed again after some time (e.g., after the camera performs a loop) [3, 11, 17]. This mechanism is not present in BARFT, and as the NeRF trains, the map (and camera poses) becomes harder to drastically adjust if there is a new important observation, like after a loop closure (both because we reach a local minimum and the exponential learning rate decay). We observe this limitation in our results as we see the most significant pose errors later in the video sequence as the poses and map slowly drift and accumulate errors (see Fig. 5). Incorporating mechanisms like loop closure in SLAM to allow for significant adjustments could improve the performance of our system.

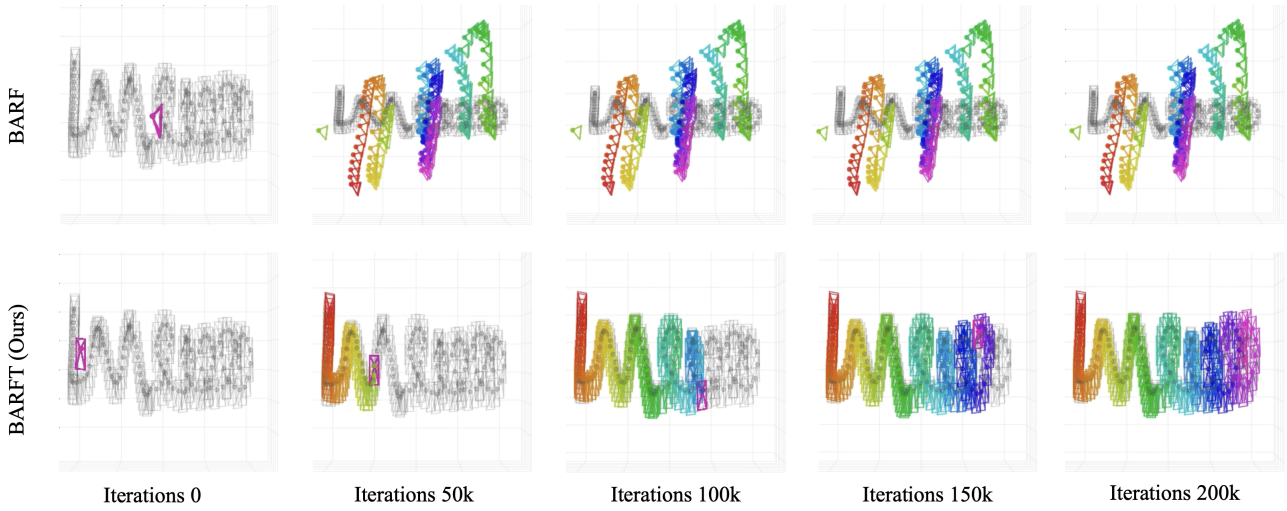

Figure 6. Camera Pose Registration for the WORKOUTAREA dataset. Unlike Fig. 5 and Fig. 4, we show a forward-facing view of the camera poses in this figure. We notice that BARFT tracks the ground truth accurately while BARF does not converge.

6. Conclusion

In summary we presented BARFT, which exploits temporal information between consecutive frames in videos to enable joint optimization of camera poses and NeRF without any camera pose initialization and with large camera viewpoint changes. We show experimental results of our method outperforming previous state of the art across several datasets exhibiting challenging camera movements. Finally we note the limitations of our work along with a discussion on next steps and future directions.

References

[1] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building rome in a day. *Communications of the ACM*, 54(10):105–112, 2011. 2

[2] Hatem Alismail, Brett Browning, and Simon Lucey. Photometric bundle adjustment for vision-based slam. In *Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part IV 13*, pages 324–341. Springer, 2017. 2

[3] Xieyuanli Chen, Thomas Läbe, Andres Milioto, Timo Röhling, Olga Vysotska, Alexandre Haag, Jens Behley, and Cyrill Stachniss. Overlapnet: Loop closing for lidar-based slam. In *Robotics: Science and Systems XVI*, RSS2020. Robotics: Science and Systems Foundation, July 2020. 7

[4] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):1052–1067, 2007. 2

[5] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018. 2

[6] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):611–625, 2017. 2

[7] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *European conference on computer vision*, pages 834–849. Springer, 2014. 2

[8] Jiatao Gu, Alex Trevithick, Kai-En Lin, Joshua M Susskind, Christian Theobalt, Lingjie Liu, and Ravi Ramamoorthi. Nerfdiff: Single-image view synthesis with nerf-guided distillation from 3d-aware diffusion. In *International Conference on Machine Learning*, pages 11808–11826. PMLR, 2023. 4

[9] Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. Splatam: Splat, track & map 3d gaussians for dense rgb-d slam. *arXiv preprint*, 2023. 1, 2, 6

[10] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023. 2

[11] Mathieu Labbé and François Michaud. Online global loop closure detection for large-scale multi-session graph-based slam. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2661–2666, 2014. 7

[12] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5741–5751, 2021. 1, 2, 3, 4, 5

[13] 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. *Communications of the ACM*, 65(1):99–106, 2021. 1, 2

[14] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022. 6

[15] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015. 2

[16] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In *2011 international conference on computer vision*, pages 2320–2327. IEEE, 2011. 2

[17] P. Newman and Kin Ho. Slam-loop closing with visually salient features. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 635–642, 2005. 7

[18] Marc Pollefeys, Reinhard Koch, and Luc Van Gool. Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters. *International journal of computer vision*, 32(1):7–25, 1999. 2

[19] Marc Pollefeys, Luc Van Gool, Maarten Vergauwen, Frank Verbiest, Kurt Cornelis, Jan Tops, and Reinhard Koch. Visual modeling with a hand-held camera. *International Journal of Computer Vision*, 59:207–232, 2004. 2

[20] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-Motion Revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 4, 5

[21] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise View Selection for Unstructured Multi-View Stereo. In *European Conference on Computer Vision (ECCV)*, 2016. 1

[22] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM siggraph 2006 papers*, pages 835–846. 2006. 2

[23] Noah Snavely, Steven M Seitz, and Richard Szeliski. Modeling the world from internet photo collections. *International journal of computer vision*, 80:189–210, 2008. 2

[24] Peng Wang, Lingzhe Zhao, Ruijie Ma, and Peidong Liu. Bad-nerf: Bundle adjusted deblur neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4170–4179, 2023. 1, 2

[25] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 4

[26] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf–: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021. 1, 2

[27] Changchang Wu. Visualsfm: A visual structure from motion system. *http://www. cs. washington. edu/homes/ccwu/vsfm*, 2011. 2

[28] Anqi Joyce Yang, Can Cui, Ioan Andrei Bârsan, Raquel Urtasun, and Shenlong Wang. Asynchronous multi-view slam. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5669–5676. IEEE, 2021. 2

[29] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 4