SIREN: Semantic, Initialization-Free Registration of Multi-Robot Gaussian Splatting Maps

Ola Shorinwa[1], **Jiankai Sun**[2], **Mac Schwager**[2], **Anirudha Majumdar**[1]
[1]Princeton University [2]Stanford University
siren-robot.github.io

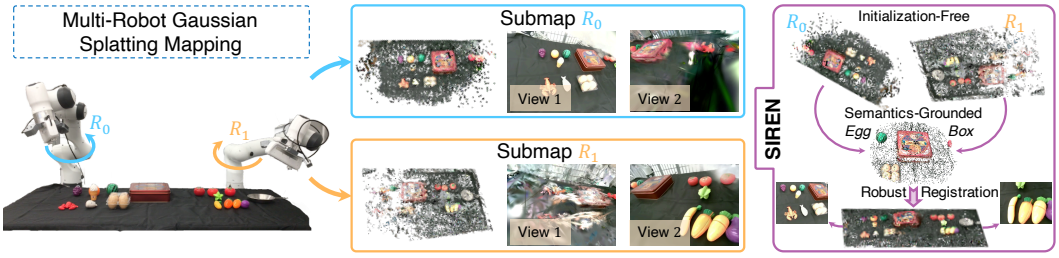

Figure 1: SIREN enables robust registration (i.e., fusion) of multi-robot Gaussian Splatting maps, with no access to camera poses, images, and inter-map relative poses, via semantics-grounded optimization centered on feature-rich regions of each map.

Abstract: Despite remarkable advances in Gaussian Splatting (GSplat), GSplat mapping remains primarily limited to single-robot, single-deployment use-cases. Existing methods generally require a good initialization or global information on inter-map transforms, limiting their effectiveness in practice. To address these fundamental challenges, we present *SIREN* for registration with **zero** access to camera poses, images, and inter-map transforms for initialization or fusion of local submaps. SIREN introduces three key innovations: (i) semantic feature extraction and matching to identify feature-rich regions, eliminating the need for any initialization; (ii) robust geometric optimization for Gaussian-to-Gaussian matching, even in noisy maps; and (iii) photometric refinement with novel-view synthesis for high-fidelity fusion. We demonstrate the superior performance of SIREN compared to competing baselines across a range of real-world datasets, and in particular, across the most widely-used robot hardware platforms, including a manipulator, drone, and quadruped. In our experiments, SIREN achieves about 90x smaller rotation errors, 300x smaller translation errors, and 44x smaller scale errors in the most challenging scenes, where competing methods struggle.

Keywords: Multi-Robot Mapping, Gaussian Splatting, Map Registration.

1 Introduction

In robotics, traditional map representations such as point-cloud and voxel maps constitute a critical component of the robotics stack, enabling downstream behavior prediction, planning, and control across many problem domains, e.g., navigation and manipulation. However, these map representations often lack the expressiveness required to capture high-fidelity visual details and semantics [1], limiting their applications in fine-grained robotics tasks, e.g., in dexterous open-vocabulary manipulation [2]. To address these fundamental limitations, recent robotics research has adopted radiance fields as flexible, high-fidelity 3D scene representations, e.g., in robot navigation [3, 4] and manipulation [5, 6]. Radiance fields, e.g., neural radiance fields (NeRFs) [7] and Gaussian Splatting (GSplat)

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

[1], are trained entirely from monocular images, typically collected by a single robot on a single deployment.

However, practical real-world robot mapping requires multiple deployments and multiple robot platforms, especially when mapping large-scale areas. For example, mobile robots have a limited battery life, while fixed-base robotic manipulators have a limited workspace, making map registration a necessity for covering large-scale areas. Fusing map information across multiple robot platforms and deployments remains a key challenge, particularly with radiance field maps. Prior work has explored fusing multiple radiance field maps [8, 9]; however, these methods either require a good initialization of inter-map correspondences or access to the camera poses and images, which is often unavailable. Moreover, these methods often fail in unstructured real-world environments, an important operational domain for robots. Specifically, the core challenge in map registration can be largely attributed to the difficulty in identifying accurate correspondences between points [10]. In fact, given accurate correspondences, the map registration problem can be solved efficiently in closed-form.

To address these challenges, we introduce *SIREN*, a semantic, initialization-free registration algorithm for multi-robot Gaussian Splatting maps. Although often unexploited, many real-world scenes contain rich semantic information, e.g., associated with objects such as vehicles, people, and vegetation. SIREN leverages this key insight to derive a robust map registration algorithm, consisting of three central components, illustrated in Figure 1—centered on feature-rich areas of the local submaps. First, SIREN trains a semantic GSplat to directly embed semantic features in GSplat maps and subsequently uses the inherent semantics in the local maps for semantic feature extraction and matching centered on feature-rich regions. Second, SIREN formulates a geometric optimization problem for a coarse Gaussian-to-Gaussian registration, which is solved efficiently in closed-form. Third, SIREN leverages novel-view synthesis for photorealistic map fusion via photometric refinement, using a semantics-based image filter for outlier rejection.

We demonstrate the superior effectiveness of SIREN compared to both existing GSplat registration methods and classical point-cloud registration methods across different real-world datasets, including standard benchmarks for radiance fields and data collected across three different robot hardware platforms: a quadruped, drone, and fixed-base manipulator. In almost all settings, SIREN achieves lower rotation, translation, and scale errors compared to all baselines, especially in the quadruped mapping task, where SIREN achieves about 90x lower rotation error, 300x lower translation error, and 44x lower scale errors.

2 Related Work

Semantic Radiance Fields. Large vision-language models, e.g., CLIP [11] and DINO [12, 13] have demonstrated the effectiveness of large-scale pretraining in learning robust visual and language features, enabling object detection [14, 15], object segmentation [16, 17], and image captioning [18, 19]. Prior work has examined grounding the 2D image-language features from vision-language foundation models in 3D radiance fields. CLIP-NeRF [20], DFF [21], and LERF [22] train NeRFs with CLIP image-language features, enabling open-vocabulary object segmentation and scene-editing. Similarly, subsequent work has enabled distillation of semantic features into GSplats [23, 24], with similar open-vocabulary object segmentation quality, albeit at much faster rendering rates [25]. Moreover, prior work has leveraged semantic radiance fields to enable GSplat-based world models [26] and open-vocabulary robotic manipulation in NeRFs [2, 5] and GSplat environments [6, 27]. In this work, we leverage semantic radiance fields for registration of 3D maps, which has not been explored in prior work, to the best of our knowledge.

Registration of Radiance Fields. Training large-scale radiance fields is often infeasible, due to computational resource constraints. Consequently, Nerf2nerf [28] aligns individually-trained NeRFs with different frames into a shared reference frame, by extracting the geometry of the scene from the NeRF as a surface field. Nerf2nerf requires human annotation of keypoints within each

NeRF for registration, posing a practical challenge. Similarly, the NeRF registration methods in [29, 30] computes spatial features of NeRFs using learned feature descriptors 3D primitives and subsequently estimates the transformation between the source and target NeRFs using the Kabsch-Umeyama algorithm [31] or RANSAC. More recent work has explored the registration of GSplats. LoopSplat [32] and PhotoReg [9] compute the optimal transformation between GSplat maps by minimizing the rendering loss but require access to the set of camera poses (keyframes) of each GSplat. In contrast, GaussReg [8] computes a coarse transformation between two GS-maps using a geometric transformer [33], which is refined with a 2D convolutional neural network augmented with a geometric transformer, without access to the camera poses. In contrast to these existing methods, SIREN leverages the semantics inherent to GSplat maps to identify regions of overlap and to coarsely align GSplat maps, eliminating the need for access to camera poses or images. Moreover, unlike GaussReg, SIREN does not require a separate training procedure for the learned CNN and geometric transformer models. We povide additional related work in Appendix A.

3 Robust Multi-Robot GSplat Map Registration

At its core, SIREN leverages open-vocabulary semantics within a principled optimization-based framework to enable the robust registration of multi-robot GSplat maps, via: (i) semantic feature extraction and matching, (ii) coarse Gaussian-to-Gaussian geometric registration, and (iii) Fine photometric registration, illustrated in Figure 2. Here, for simplicity, we discuss the registration pipeline in the problem setting with two multi-robot maps, where we seek to register a source GSplat map to a target GSplat map. However, the discussion applies to the registration of multiple local multi-robot maps.

First, SIREN identifies corresponding pairs of Gaussians in a pair of GSplat maps by examining the similarity between the semantic features of the ellipsoids. Subsequently, given the set of corresponding Gaussians, SIREN solves a Gaussian-to-Gaussian optimization problem to compute the optimal transformation aligning the pair of multi-robot maps with a robust objective function, which leverages the semantic similarity between each pair of ellipsoids to guard against the impacts of outliers. Lastly, SIREN harnesses the novel-view synthesis capabilities of Gaussian Splatting to render candidate images for image-to-image registration, enabling fine registration of both maps via a structure-from-motion-based approach. In this stage, SIREN utilizes image-level semantic features to identify pairs of corresponding images, critical to the robust matching of local features such as corners, edges, and blobs between the images.

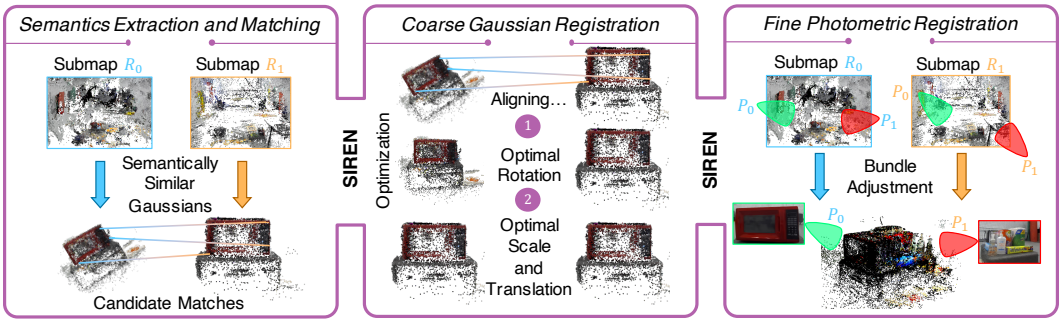

Figure 2: SIREN consists of three steps: (i) semantic feature extraction and matching of Gaussians across the local maps, (ii) coarse Gaussian-to-Gaussian registration for coarsely aligning the local maps, (iii) fine photometric registration for high-accuracy fusion of the local maps, through image-to-image registration and bundle adjustment. Here, SIREN identifies a microwave in a kitchen scene in both maps, extracts the Gaussians corresponding to the microwave, aligns these Gaussians, and subsequently refines the transformation.

3.1 Semantic Feature Extraction and Matching

Noting that semantics underpin SIREN, we begin with a discussion of the semantic distillation procedure utilized by SIREN in grounding 2D semantic information from the vision-language model in GSplat maps, where we associate semantic embeddings with each ellipsoid in the GSplat map.

Semantic Gaussian Splatting. Existing methods for training semantic GSplats generally train auxiliary models, e.g., autoencoders or CNNs, for dimensionality reduction of the semantic features from vision-language models to compute lower-dimensional semantic features, which are distilled into the GSplat [23, 24]. These methods require relatively significant computation time and GPU memory, which we seek to avoid in SIREN (see [25] for a discussion of these limitation). Consequently, we take a different approach to semantic distillation. In SIREN, alongside the GSplat, we simultaneously train a semantic field $\psi : \mathbb{R}^3 \mapsto \mathbb{R}^d$, which maps 3D points to d-dimensional semantic features, where d is determined by the vision-language foundation model, e.g., $d = 512$ or 1024 in CLIP. We parameterize ψ with a multi-resolution neural hashgrid, using the loss function:

$$\mathcal{L} = \mathcal{L}_{\text{gs}} + \gamma \sum_{\mathcal{I} \in \mathcal{D}} \left\| \mathcal{I}_f - \hat{\mathcal{I}}_f \right\|_F^2 - \beta \sum_{\mathcal{I} \in \mathcal{D}} \phi(\mathcal{I}_f, \hat{\mathcal{I}}_f), \tag{1}$$

where $\mathcal{I}_f \in \mathbb{R}^{W \times H \times d}$ and $\hat{\mathcal{I}}_f \in \mathbb{R}^{W \times H \times d}$ represent the ground-truth and predicted semantic feature maps associated with each image in the training dataset $\mathcal{D}$, γ and β represent relative weight terms, and ϕ represents the cosine-similarity function between each semantic feature in $\mathcal{I}_f$ and $\hat{\mathcal{I}}_f$. We provide additional details on the SIREN's semantic Gaussian Splatting in Appendix C.

Feature Extraction. In the feature extraction and matching step, SIREN identifies feature-rich areas of the scene via semantic localization, to improve the robustness of the subsequent optimization-based registration steps, as the feasibility and convergence of the optimization problems significantly depend on the presence of informative features. Given a trained semantic GSplat, we augment each Gaussian with a semantic attribute, computed by querying the semantic field ψ at the mean μ of each Gaussian. Subsequently, from a set of open-vocabulary queries (e.g., generated by an image-tagging model [34]), we compute the semantic relevancy score between each Gaussian and the natural-language query by taking the pairwise softmax over the cosine-similarity between the semantic feature of each Gaussian and the semantic embedding associated with the text query and the cosine-similarity between the semantic feature of each Gaussian and the semantic embedding associated with a generic or null text query (i.e., a text query for a generic object or an object a user does not want to localize) [22]. Depending on the quality of the local multi-robot maps, SIREN post-processes the resulting set of Gaussians to either inflate the set by incorporating neighboring Gaussians geometrically or semantically, or deflate the set by removing (statistical) outliers.

Feature Matching. Given the set of extracted Gaussians for each map, we match Gaussians from the source GSplat map to the target GSplat map, resulting in a set of correspondences $\mathcal{E}$ where $(i, j) \in \mathcal{E}$ indicates that Gaussian i in the source GSplat map corresponds to Gaussian j in target GSplat map. To identify the candidate matches in $\mathcal{E}$, we compute the cosine-similarity between the semantic embeddings of the Gaussians in both maps, matching the Gaussians in the submaps to a random set of M Gaussians in the other map, with uniform sampling or informed sampling with probability proportional to the cosine-similarity values. For computational reasons, this operation can be performed using efficient data structures such as KD-trees. Moreover, the matching process can be augmented with geometric information, by selecting candidate matches using geometric descriptors, such as the Fast Point Feature Histograms (FPFH) descriptors [35]. We denote the set of Gaussians in the source map present in $\mathcal{E}$ by $\mathcal{P}$ and the set of Gaussians in the target map present in $\mathcal{E}$ by $\mathcal{Q}$.

3.2 Coarse Gaussian-to-Gaussian Registration

From the matched Gaussians, SIREN computes an initial non-rigid transformation, consisting of a scale $s_c \in \mathbb{R}$, a rotation matrix $R \in \text{SO}(3)$, and a translation vector $t \in \mathbb{R}^3$, aligning the Gaussians in the source and target GSplat maps. Since computing this transformation using all the Gaussians is

intractable in general, we solve for this transformation using the Gaussians in $\mathcal{P}$ and $\mathcal{Q}$, a feature-dense, much smaller set of Gaussians, a design choice that not only reduces the computational cost, but also improves the feasibility and convergence properties of the resulting optimization problem. Specifically, we formulate the coarse Gaussian-to-Gaussian registration problem as an optimization problem over the transformation parameters, given by:

$$\underset{s_c\in\mathbb{R}_{++},R\in\mathrm{SO}(3),t\in\mathbb{R}^3}{\text{minimize}} \ \frac{1}{2}\sum_{(i,j)\in\mathcal{E}} w_{ij}\left(\|s_cRp_i+t-q_j\|_2^2+\left\|s_c^2R\Sigma_{p_i}R^\mathsf{T}-\Sigma_{q_j}\right\|_F^2\right), \tag{2}$$

where $p_i \in \mathbb{R}^3$ denotes the mean of Gaussian i in the source map, $q_j \in \mathbb{R}^3$ denotes the mean of Gaussian j in the target map, and Σ_{p_i} and Σ_{q_j} denote the covariance of Gaussian i in the source map and Gaussian j in the target map, respectively. We solve this problem in closed-form (described in Appendix D), with:

$$R_c^\star = U_c\Theta_cV_c^\top, \ \ s_c^\star = \frac{\mathrm{trace}(\Theta_c\Sigma)}{\mathrm{trace}\left(W\check{P}^\top\check{P}+\sum_{(i,j)\in\mathcal{E}} w_{ij}\check{H}_{p_i}^\top\check{H}_{p_i}\right)}, \ \ t_c^\star = \tilde{\mu}_\mathcal{Q}-s_c^\star R^\star\tilde{\mu}_\mathcal{P}. \tag{3}$$

Although the resulting solution is optimal for the problem in (4), the solution of (4) might not be optimal for the registration of the two sets of Gaussians, i.e., $\mathcal{P}$ and $\mathcal{Q}$, given that $\mathcal{C}$ might contain spurious correspondences. To improve the robustness of SIREN to spurious correspondences, we utilize RANSAC [36] when solving the optimization problem in (4). With RANSAC, we iteratively update the correspondences in $\mathcal{C}$ to remove false correspondences and compute an optimal transformation associated with the resulting set of correspondences.

3.3 Fine Photometric Registration

Coarse registration fails to leverage the highly-informative visual features inherent in the GSplat maps, effectively limiting the accuracy of the estimated transformation. To overcome this limitation, SIREN generates photorealistic images from each submap and optimizes over the resulting set of rendered images to compute a transformation consistent with the rendered images from both submaps. The fine photometric registration procedure employs a lightweight structure-from-motion framework to improve registration fidelity without expensive computation costs. This procedure consists of the following steps: (i) image generation, where we exploit novel-view synthesis in Gaussian Splatting to render images and compute corresponding poses in both maps with semantic image filtering; (ii) image registration and triangulation, using learned feature extractors; and (iii) bundle adjustment, which we solve approximately in closed-form. We describe this procedure further in Appendix E.

4 Experiments

We examine the performance of SIREN in comparison to existing registration methods for Gaussian Splatting and point clouds. Specifically, we compare two variants of SIREN—i.e., SIREN-NR, which solves the optimization problem (4) in closed-form without RANSAC, and SIREN-R, which utilizes RANSAC for coarse registration—to state-of-the-art GSplat registration methods GaussReg [8] and PhotoReg [9], in addition to RANSAC-based global registration (RANSAC-GR) [36, 37], Fast Global Registration (FGR) [38], and variants of the Iterative Closest Point (ICP) [39, 40]. We do not compare against Nerf2Nerf [28] because it requires a volumetric density field, which is unavailable in GSplats, and further requires human annotation of keypoints, violating the assumptions of our work. Likewise, LoopSplat [32] solves the full SLAM problem which is outside the scope of our paper. We evaluate each method not only on standard benchmark datasets for radiance fields, but also on real-world data collected by heterogeneous robot platforms, including a quadruped, drone, and manipulator (in the case of SIREN). In all our experiments, we only require the trained GSplat models as input; however, some of the baselines require access to the set of camera poses, which we provide when evaluating these methods. We ablate the different components of SIREN, quantifying the relative improvements in performance, and examine the gains in visual fidelity afforded by finetuning the fused model. Further, we demonstrate SIREN in collaborative multi-robot mapping, where the mapping task cannot

be accomplished by a single robot, necessitating mapping with multiple robots for task success. We provide additional results and discussion in Appendix F.

4.1 Mip-NeRF360 Dataset

We utilize the *Playroom*, *Truck*, and *Room* scenes in the Mip-NeRF360 Dataset, which were all collected in realistic settings with natural lighting effects. While the *Playroom* and *Room* scenes were captured indoors, the *Truck* scene was captured outdoors. We split the datasets into two subsets with varying overlap. Specifically, the first subset of the *Truck* scene captures the left side of the truck, while the second subset captures the right side of the truck. The only overlap between both subsets occurs at the front and rear of the truck. We split the *Room* scene into two subsets following the same procedure. In the *Playroom* scene, we allow for greater overlap, with the density of images per subregion of the scene varying between both subsets. We train independent GSplat maps for each scene-subset pair.

Geometric Evaluation. In Table 1, we report the geometric errors of each registration method across the three scenes. SIREN-R, our method, achieves the lowest rotation and translation errors in two of the three scenes (*Playroom* and *Truck*): with about 1.14x to 8.89x lower rotation errors and about 6x to 46x lower translation errors compared to the baseline methods. Meanwhile, in the *Room* scene, SIREN-NR achieves the lowest translation and scale error, with SIREN-R achieving the second-best performance on these metrics. In contrast to the baselines, SIREN leverages the semantic features in the submaps to identify accurate matches and optimize the relative transformation between the submaps, even in feature-sparse scenes. For example, whereas the baselines suffer a significant degradation in performance of the baselines in the *Truck* scene, which contains the smallest number of informative features, SIREN still computes a high-fidelity fused map. Moreover, estimating accurate scale and translation transformation parameters is particularly challenging in the presence of outliers. SIREN mitigates this challenge by identifying and removing outliers.

Table 1: Geometric performance of the registration algorithms on the Mip-NeRF360 dataset (see Section F for a description of the metrics).

	Playroom				*Truck*				*Room*			
Methods	RE ↓	TE ↓	SE ↓	CT ↓	RE ↓	TE ↓	SE ↓	CT ↓	RE ↓	TE ↓	SE ↓	CT ↓
PhotoReg [9]	6.036	18806	841.3	2177	177.3	2856	444.0	1814	0.161	4983	452.7	1409
GaussReg [8]	0.766	55.50	0.364	15.06	21.10	316.3	16.76	5.174	7.464	628.3	91.97	6.932
RANSAC-GR [36, 37]	4.835	56.22	17.85	0.996	46.72	2642	13.64	2.569	8.139	194.7	152.5	0.517
FGR [38]	2.988	18.83	14.37	0.887	3.778	2231	79.45	3.480	4.869	265.6	219.6	0.511
ICP [39]	2.362	19.11	14.37	2.127	3.672	2232	79.45	3.805	5.154	266.1	219.6	1.579
Colored-ICP [40]	0.194	12.28	14.37	3.951	4.043	2250	79.45	6.392	2.256	232.7	219.6	3.815
SIREN-NR [Ours]	0.348	4.860	0.282	41.16	0.511	8.07	9.581	53.42	0.381	2.648	1.016	40.24
SIREN-R [Ours]	0.170	1.933	0.170	39.73	0.413	6.845	2.548	52.47	0.237	3.289	2.673	39.71

Photometric Evaluation. Now, we examine the photometric performance of the GSplat registration methods reported in Table 2. SIREN-R achieves the best photometric performance in the *Playroom* scene. Similarly, in the *Room* scene, SIREN-NR achieves the best photometric performance across all metrics, followed by SIREN-R. These findings are consistent with the results in Table 1. Note that the coarse and fine registration procedures enable SIREN to construct highly-accurate fused maps that capture photorealistic details. In Figure 3, we show the rendered images from the fused GSplat maps generated by the registration methods from different viewpoints compared to the ground-truth images. We visualize a pair of images from the *Playroom*, *Truck*, and *Room* scenes, restricting our visualizations to PhotoReg, GaussReg, Colored-ICP, and SIREN-R due to space considerations. In the *Playroom* scene, PhotoReg fails to sufficiently register the individual maps to obtain photorealistic renderings, while other methods generate high-fidelity renderings. However, unlike SIREN, the fused maps in GaussReg and Colored-ICP are not accurately aligned, highlighted by the inset images.

Table 2: Photometric performance of registration algorithms for GSplat maps from the Mip-NeRF360 dataset.

Methods	*Playroom* PSNR ↑	*Playroom* SSIM ↑	*Playroom* LPIPS ↓	*Truck* PSNR ↑	*Truck* SSIM ↑	*Truck* LPIPS ↓	*Room* PSNR ↑	*Room* SSIM ↑	*Room* LPIPS ↓
PhotoReg [9]	11.5 ± 2.3	0.68 ± 0.11	0.67 ± 0.12	10.6 ± 0.9	0.39 ± 0.07	0.72 ± 0.08	10.2 ± 1.2	0.46 ± 0.07	0.78 ± 0.04
GaussReg [8]	23.7 ± 3.3	0.86 ± 0.06	0.22 ± 0.08	13.4 ± 1.9	0.54 ± 0.12	0.53 ± 0.13	13.4 ± 3.6	0.61 ± 0.11	0.55 ± 0.15
RANSAC-GR [36, 37]	17.9 ± 3.2	0.77 ± 0.09	0.37 ± 0.09	18.9 ± 7.0	0.66 ± 0.24	0.32 ± 0.22	14.2 ± 2.4	0.66 ± 0.09	0.46 ± 0.11
FGR [38]	22.2 ± 3.2	0.85 ± 0.06	0.24 ± 0.08	13.0 ± 2.6	0.57 ± 0.19	0.43 ± 0.20	17.2 ± 2.3	0.77 ± 0.08	0.33 ± 0.11
ICP [39]	22.7 ± 3.3	0.85 ± 0.06	0.24 ± 0.08	12.9 ± 2.6	0.57 ± 0.19	0.43 ± 0.20	16.8 ± 2.2	0.76 ± 0.09	0.35 ± 0.11
Colored-ICP [40]	26.2 ± 3.1	0.89 ± 0.04	0.17 ± 0.06	13.4 ± 2.4	0.58 ± 0.19	0.40 ± 0.18	15.4 ± 2.1	0.71 ± 0.10	0.41 ± 0.13
SIREN-NR [Ours]	26.3 ± 3.1	0.87 ± 0.05	0.17 ± 0.06	15.4 ± 1.7	0.52 ± 0.12	0.35 ± 0.05	24.8 ± 3.3	0.83 ± 0.04	0.22 ± 0.06
SIREN-R [Ours]	28.3 ± 2.9	0.90 ± 0.04	0.15 ± 0.06	16.4 ± 2.4	0.57 ± 0.13	0.31 ± 0.07	24.1 ± 3.1	0.82 ± 0.05	0.23 ± 0.06

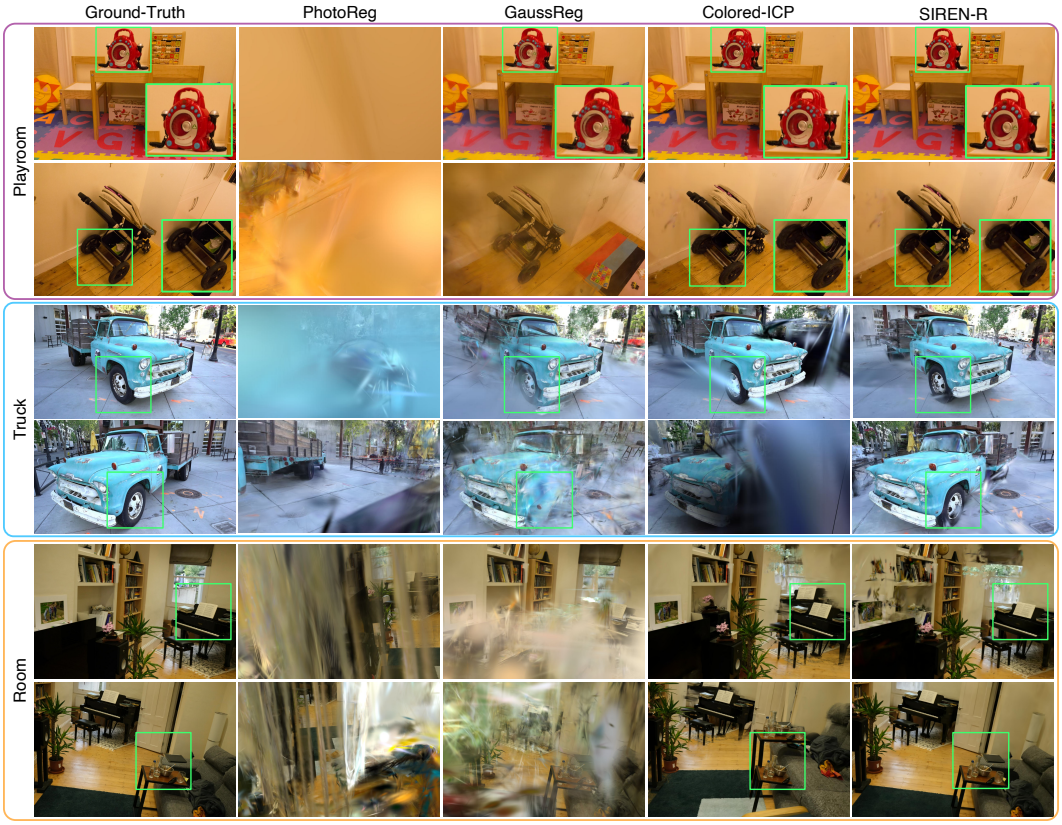

Figure 3: Rendered images from the fused GSplat maps of the *Playroom*, *Truck*, and *Room* scenes. Unlike the baselines, SIREN generates high-fidelity fused GSplat maps, evidenced by the precise geometric detail in the images, visible in the regions indicated by the green squares. Inaccurate registration of GSplat maps generally result in artifacts in the rendered images.

4.2 Mobile-Robot Mapping

Here, we utilize a quadruped and a drone to map three environments, depicted in Figure 4. The quadruped maps the *Kitchen* and *Workshop* environments, while the drone maps an *Apartment* scene, with multiple partitioned room-like areas. The robots create submaps in each environment individually, containing different regions of the scene. The submaps in the *Kitchen* and *Workshop* scenes have minimal overlap, while the submaps in the *Apartment* scene have greater overlap. Since each submap is trained independently in different reference frames, fusing the submaps requires registration of the maps. Here, we examine the performance of GaussReg, PhotoReg, and two variants of SIREN: SIREN-NR and SIREN-R, in registering the submaps in each scene to obtain a composite map of the entire scene.

Figure 4: Stillshots of a quadruped mapping different areas of a kitchen and workshop and a drone mapping an apartment-like scene. Each robot trains independent GSplat submaps of the areas it mapped. The submaps of each scene are registered to obtain a composite map covering the entirety of the scene.

Geometric Performance. Table 3 summarizes the geometric errors of each algorithm, showing that SIREN achieves the best geometric performance across all scenes, with the top-two-performing methods being the variants of SIREN. Specifically, in the *Kitchen* scene, SIREN-NR achieves the lowest rotation, translation, and scale errors by a factor of about 160x, 465x, and 488x, respectively, compared to the best-performing baseline. The performance of SIREN-R closely follows that of SIREN-NR. Moreover, in the *Workshop* and *Apartment* scenes, SIREN achieves the lowest rotation, translation, and scale errors. These results demonstrate the superior performance of SIREN in real-world robot mapping applications. Further, GaussReg requires the least computation time across all scenes, while PhotoReg requires the greatest computation time. Although slower than GaussReg, SIREN requires much lower computation times compared to PhotoReg.

Table 3: Geometric performance of GSplat registration algorithms in mobile-robot mapping.

Methods	*Kitchen*				*Workshop*				*Apartment*			
	RE ↓	TE ↓	SE ↓	CT ↓	RE ↓	TE ↓	SE ↓	CT ↓	RE ↓	TE ↓	SE ↓	CT ↓
PhotoReg [9]	40.49	2350	413.37	1042	140.5	10052	4310	934.2	24.09	4433	260.2	801.0
GaussReg [8]	40.89	1477	171.8	11.33	55.66	9531	4305	5.491	3.114	102.6	13.59	5.4983
SIREN-NR [**Ours**]	0.253	3.173	0.352	59.22	0.518	11.77	1.453	67.98	0.148	1.758	0.605	35.91
SIREN-R [**Ours**]	0.430	4.795	3.849	56.14	0.134	7.400	10.88	55.16	0.119	1.495	0.102	34.22

Photometric Performance. In line with the geometric results, SIREN outperforms all the baseline methods, as reported in Table 4. While SIREN-R achieves the best photometric scores in the *Workshop* scene, SIREN-NR attains the best performance in the *Kitchen* scene, followed by SIREN-R. In the *Apartment* scene, SIREN-R and SIREN-NR achieve the best photometric results. Further, GaussReg outperforms PhotoReg in all scenes.

Table 4: Photometric performance of GSplat registration algorithms for mobile-robot mapping.

Methods	*Kitchen*			*Workshop*			*Apartment*		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
PhotoReg [9]	0.75 ± 0.05	11.6 ± 1.3	0.53 ± 0.06	0.78 ± 0.08	11.3 ± 1.3	0.48 ± 0.05	13.4 ± 1.5	0.61 ± 0.05	0.75 ± 0.04
GaussReg [8]	14.1 ± 1.9	0.62 ± 0.06	0.60 ± 0.11	17.0 ± 2.8	0.61 ± 0.05	0.53 ± 0.11	14.3 ± 1.6	0.62 ± 0.05	0.62 ± 0.04
SIREN-NR [**Ours**]	19.3 ± 3.2	0.63 ± 0.05	0.40 ± 0.09	19.9 ± 2.5	0.60 ± 0.04	0.40 ± 0.08	15.3 ± 1.5	0.64 ± 0.04	0.55 ± 0.03
SIREN-R [**Ours**]	18.8 ± 2.8	0.62 ± 0.05	0.41 ± 0.08	20.3 ± 2.7	0.62 ± 0.04	0.38 ± 0.09	15.3 ± 1.4	0.63 ± 0.04	0.55 ± 0.03

5 Conclusion

We present SIREN, a semantics-grounded registration algorithm for multi-robot GSplat maps that neither requires access to camera poses/images nor initialization of inter-map relative transforms. SIREN harnesses the robustness of semantics to: (i) identify candidate matches between Gaussians across the input maps robustly, (ii) compute a coarse transformation between the submaps from a Gaussian-to-Gaussian registration problem with outlier rejection, and (iii) refine the coarse registration result for high-accuracy fusion of local submaps into a high-fidelity global map. We demonstrate the versatility of SIREN across maps constructed by robots of different embodiments, including a quadruped, drone, and manipulator, highlighting the superior performance of SIREN compared to GSplat registration algorithms and classical point-cloud registration methods.

6 Limitations and Future Work

Semantic GSplats. Our registration algorithm relies on semantics for robust, initialization-free registration, and thus requires that the input maps have embedded semantic codes. A GSplat map may lack semantic information if the map was not trained with semantics or if the scene lacks any semantically-relevant features, which would be a tail event in practical situations. We can post-train GSplats to embed semantics in 3D into the map or leverage 2D vision foundation models to directly extract semantic information from RGB images rendered from the GSplat by back-projecting 2D pixels into the 3D world.

Symmetric or Feature-sparse Scenes. Moreover, the feature extraction step might fail in areas of the map with particularly few semantic features for the matching process, e.g., a featureless wall. Likewise, in scenarios where each map contains many similar features, e.g., many copies of the same object, the feature matching procedure might generate many spurious (false) matches. Although the coarse and fine registration procedures are robust to outliers, these components would fail with too many outliers. The exact number of outliers depends on the application and the influence of the outliers on the estimated transformation. We emphasize that these limitations are also faced by other existing map-registration methods. To minimize these risks, our method utilizes both semantic and geometric features in the feature extraction and matching procedures, when applicable.

Floaters. Radiance fields are prone to generate floaters in areas of the scene with little to no supervision, which can degrade the fidelity of the map. The resulting floaters are retained in the fused map, which could ultimately reduce the accuracy of the map. However, by finetuning the fused map with synthetic data, i.e., images rendered from the map as opposed to real-world images, floaters in the map can be removed for high-fidelity mapping.

Acknowledgments

This work was supported in part by NSF grant 2342246, NSF CAREER Award 2044149, Office of Naval Research N00014-23-1-2148, and Princeton SEAS Innovation Award from The Addy Fund for Excellence in Engineering. Toyota Research Institute provided funds to support this work. Jiankai Sun is partially supported by Stanford Interdisciplinary Graduate Fellowship.

References

[1] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

[2] W. Shen, G. Yang, A. Yu, J. Wong, L. P. Kaelbling, and P. Isola. Distilled feature fields enable few-shot language-guided manipulation. *arXiv preprint arXiv:2308.07931*, 2023.

[3] T. Chen, O. Shorinwa, J. Bruno, J. Yu, W. Zeng, K. Nagami, P. Dames, and M. Schwager. Splat-nav: Safe real-time robot navigation in gaussian splatting maps. *arXiv preprint arXiv:2403.02751*, 2024.

[4] R.-Z. Qiu, Y. Hu, G. Yang, Y. Song, Y. Fu, J. Ye, J. Mu, R. Yang, N. Atanasov, S. Scherer, et al. Learning generalizable feature fields for mobile manipulation. *arXiv preprint arXiv:2403.07563*, 2024.

[5] A. Rashid, S. Sharma, C. M. Kim, J. Kerr, L. Y. Chen, A. Kanazawa, and K. Goldberg. Language embedded radiance fields for zero-shot task-oriented grasping. In *7th Annual Conference on Robot Learning*, 2023.

[6] O. Shorinwa, J. Tucker, A. Smith, A. Swann, T. Chen, R. Firoozi, M. D. Kennedy, and M. Schwager. Splat-mover: Multi-stage, open-vocabulary robotic manipulation via editable gaussian splatting. In *8th Annual Conference on Robot Learning*, 2024.

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

[8] J. Chang, Y. Xu, Y. Li, Y. Chen, W. Feng, and X. Han. Gaussreg: Fast 3d registration with gaussian splatting. In *European Conference on Computer Vision*, pages 407–423. Springer, 2025.

[9] Z. Yuan, T. Zhang, M. Johnson-Roberson, and W. Zhi. Photoreg: Photometrically registering 3d gaussian splatting models. *arXiv preprint arXiv:2410.05044*, 2024.

[10] G. K. Tam, Z.-Q. Cheng, Y.-K. Lai, F. C. Langbein, Y. Liu, D. Marshall, R. R. Martin, X.-F. Sun, and P. L. Rosin. Registration of 3d point clouds and meshes: A survey from rigid to nonrigid. *IEEE transactions on visualization and computer graphics*, 19(7):1199–1217, 2012.

[11] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[12] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[13] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[14] X. Gu, T.-Y. Lin, W. Kuo, and Y. Cui. Open-vocabulary object detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921*, 2021.

[15] M. Minderer, A. Gritsenko, A. Stone, M. Neumann, D. Weissenborn, A. Dosovitskiy, A. Mahendran, A. Arnab, M. Dehghani, Z. Shen, et al. Simple open-vocabulary object detection. In *European Conference on Computer Vision*, pages 728–755. Springer, 2022.

[16] Z. Wang, Y. Lu, Q. Li, X. Tao, Y. Guo, M. Gong, and T. Liu. Cris: Clip-driven referring image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11686–11695, 2022.

[17] T. Lüddecke and A. Ecker. Image segmentation using text and image prompts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7086–7096, 2022.

[18] R. Mokady, A. Hertz, and A. H. Bermano. Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*, 2021.

[19] H. Luo, L. Ji, M. Zhong, Y. Chen, W. Lei, N. Duan, and T. Li. Clip4clip: An empirical study of clip for end to end video clip retrieval and captioning. *Neurocomputing*, 508:293–304, 2022.

[20] C. Wang, M. Chai, M. He, D. Chen, and J. Liao. Clip-nerf: Text-and-image driven manipulation of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3835–3844, 2022.

[21] S. Kobayashi, E. Matsumoto, and V. Sitzmann. Decomposing nerf for editing via feature field distillation. *Advances in Neural Information Processing Systems*, 35:23311–23330, 2022.

[22] J. Kerr, C. M. Kim, K. Goldberg, A. Kanazawa, and M. Tancik. Lerf: Language embedded radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19729–19739, 2023.

[23] M. Qin, W. Li, J. Zhou, H. Wang, and H. Pfister. Langsplat: 3d language gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20051–20060, 2024.

[24] S. Zhou, H. Chang, S. Jiang, Z. Fan, Z. Zhu, D. Xu, P. Chari, S. You, Z. Wang, and A. Kadambi. Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21676–21685, 2024.

[25] O. Shorinwa, J. Sun, and M. Schwager. Fast-splat: Fast, ambiguity-free semantics transfer in gaussian splatting. *arXiv preprint arXiv:2411.13753*, 2024.

[26] G. Lu, S. Zhang, Z. Wang, C. Liu, J. Lu, and Y. Tang. Manigaussian: Dynamic gaussian splatting for multi-task robotic manipulation. In *European Conference on Computer Vision*, pages 349–366. Springer, 2025.

[27] M. Ji, R.-Z. Qiu, X. Zou, and X. Wang. Graspsplats: Efficient manipulation with 3d feature splatting. *arXiv preprint arXiv:2409.02084*, 2024.

[28] L. Goli, D. Rebain, S. Sabour, A. Garg, and A. Tagliasacchi. nerf2nerf: Pairwise registration of neural radiance fields. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9354–9361. IEEE, 2023.

[29] H. Jiang, R. Li, H. Sun, Y.-W. Tai, and C.-K. Tang. Registering neural radiance fields as 3d density images. *arXiv preprint arXiv:2305.12843*, 2023.

[30] Y. Chen and G. H. Lee. Dreg-nerf: Deep registration for neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22703–22713, 2023.

[31] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 13(04):376–380, 1991.

[32] L. Zhu, Y. Li, E. Sandström, S. Huang, K. Schindler, and I. Armeni. Loopsplat: Loop closure by registering 3d gaussian splats. *arXiv preprint arXiv:2408.10154*, 2024.

[33] Z. Qin, H. Yu, C. Wang, Y. Guo, Y. Peng, S. Ilic, D. Hu, and K. Xu. Geotransformer: Fast and robust point cloud registration with geometric transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(8):9806–9821, 2023.

[34] Y. Zhang, X. Huang, J. Ma, Z. Li, Z. Luo, Y. Xie, Y. Qin, T. Luo, Y. Li, S. Liu, et al. Recognize anything: A strong image tagging model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1724–1732, 2024.

[35] R. B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (fpfh) for 3d registration. In *2009 IEEE international conference on robotics and automation*, pages 3212–3217. IEEE, 2009.

[36] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6): 381–395, 1981.

[37] D. Holz, A. E. Ichim, F. Tombari, R. B. Rusu, and S. Behnke. Registration with the point cloud library: A modular framework for aligning in 3-d. *IEEE Robotics & Automation Magazine*, 22 (4):110–124, 2015.

[38] Q.-Y. Zhou, J. Park, and V. Koltun. Fast global registration. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 766–782. Springer, 2016.

[39] S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. In *Proceedings third international conference on 3-D digital imaging and modeling*, pages 145–152. IEEE, 2001.

[40] J. Park, Q.-Y. Zhou, and V. Koltun. Colored point cloud registration revisited. In *Proceedings of the IEEE international conference on computer vision*, pages 143–152, 2017.

[41] K. Zhang, G. Riegler, N. Snavely, and V. Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020.

[42] A. Yu, V. Ye, M. Tancik, and A. Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4578–4587, 2021.

[43] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5470–5479, 2022.

[44] A. Guédon and V. Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5354–5363, 2024.

[45] B. Huang, Z. Yu, A. Chen, A. Geiger, and S. Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024.

[46] Z. Yu, A. Chen, B. Huang, T. Sattler, and A. Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19447–19456, 2024.

[47] B. Lee, H. Lee, X. Sun, U. Ali, and E. Park. Deblurring 3d gaussian splatting. In *European Conference on Computer Vision*, pages 127–143. Springer, 2025.

[48] P. J. Besl and N. D. McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. Spie, 1992.

[49] Y. Chen and G. Medioni. Object modelling by registration of multiple range images. *Image and vision computing*, 10(3):145–155, 1992.

[50] S. Bouaziz, A. Tagliasacchi, and M. Pauly. Sparse iterative closest point. In *Computer graphics forum*, volume 32, pages 113–123. Wiley Online Library, 2013.

[51] Y. Wang and J. M. Solomon. Deep closest point: Learning representations for point cloud registration. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3523–3532, 2019.

[52] K. Fu, S. Liu, X. Luo, and M. Wang. Robust point cloud registration framework based on deep graph matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8893–8902, 2021.

[53] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5297–5307, 2016.

[54] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020.

[55] E. Karami, S. Prasad, and M. Shehata. Image matching using sift, surf, brief and orb: performance comparison for distorted images. *arXiv preprint arXiv:1710.02726*, 2017.

[56] J. L. Schonberger and J.-M. Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016.

[57] M. Tancik, E. Weber, E. Ng, R. Li, B. Yi, T. Wang, A. Kristoffersen, J. Austin, K. Salahi, A. Ahuja, et al. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–12, 2023.

Appendix A Additional Related Work

Radiance Fields. Neural radiance fields (NeRFs) [7] significantly outperform traditional 3D scene reconstruction methods, such as those based on point clouds and voxels, generating photorealistic renderings, which capture intricate levels of geometric and visual details. NeRFs represent a scene using volumetric density and color fields over a 5D input space, comprising a 3D location and a 2D viewing direction. NeRFs parameterize each field using multi-layer perceptrons (MLPs) trained through gradient descent. Although NeRFs achieve remarkable high-fidelity reconstructions, NeRFs are limited by significant training time and slow rendering speeds [41, 42, 43]. Gaussian Splatting [1] was introduced to address these limitations. GSplats represent the scene using ellipsoidal primitives, each with a mean and covariance (spatial and geometric parameters) and opacity and spherical harmonic parameters (visual-related parameters). GSplats generate high-fidelity scene renderings at real-time speeds with generally faster training times compared to NeRFs. Recent work has improved the geometric accuracy of GSplats [44, 45], in addition to eliminating high-frequency artifacts [46, 47].

Point Cloud Registration. The Iterative Closest Point (ICP) algorithm [48] has proven to be notably effective for point cloud registration, despite its simplicity. However, ICP generally requires a good initial solution, which is often computed using global registration techniques, e.g., RANSAC [36, 37] and FGR [38]. Many variants of ICP have been introduced to improve its robustness [49, 39, 50, 40], leveraging the local color and geometry of the constituent points for faster convergence. More recently, learning-based methods [51, 52, 33] have emerged for point cloud registration, utilizing convolutional neural networks (CNNs) and transformers for feature extraction and feature matching to compute the correspondences between points.

Appendix B Background

We provide a brief introduction to Gaussian Splatting. Gaussian Splatting represents non-empty space in a scene using a set of ellipsoidal primitives, each parameterized by a mean $\mu \in \mathbb{R}^3$, a covariance $\Sigma \in \mathbb{R}^{3\times3}$ defined by a rotation matrix $H \in \mathrm{SO}(3)$ and a diagonal scaling matrix $\Lambda \in \mathbb{R}^{3\times3}$, an opacity parameter $\alpha \in [0, 1]$, and spherical harmonic parameters. These attributes are optimized via gradient descent on the loss function: $\mathcal{L}_{\text{gs}} = (1-\lambda)\sum_{\mathcal{I}\in\mathcal{D}} \left\|\mathcal{I} - \hat{\mathcal{I}}\right\|_1 + \lambda\mathcal{L}_{\text{D-SSIM}}$, over the training dataset $\mathcal{D}$, where $\lambda \in (0, 1)$ represents the relative weight term and $\mathcal{L}_{\text{D-SSIM}}$ represents the differentiable structural similarity loss index measure. The first term in the rendering loss represents the photometric loss between the ground-truth image and the rendered image, generated via a tile-based rasterization procedure, given a camera pose.

Appendix C Semantic Gaussian Splatting

As demonstrated by prior work [25], existing semantic Splatting methods [23, 24] are inefficient to train, requiring significant training time and memory. To address this challenge, we design a neural-based semantic distillation procedure, described in the main paper. To predict the semantic feature map associated with each training image, we leverage a key insight of Gaussian Splatting: Gaussian Splatting provides highly-accurate depth estimates, even without any depth supervision [45]. This key insight enables us to avoid training proposal networks (as required in NeRFs) that generate samples of the termination points of rays associated with each pixel in the rendered image of a camera, ultimately enabling SIREN to avoid significant compute and memory overhead associated with training proposal networks. As such, given a camera pose, we back-project points from the image plane to the 3D world and pass these points into ψ to predict the semantic feature associated with these points. We augment each pixel in the image plane with its semantic features to obtain the semantic feature map. Training the GSplat with the semantic component does not adversely impact the photometric performance of the GSplat, enabling us to utilize the same hyperparameters and adaptive densification procedure used in the original GSplat work [1].

Appendix D Coarse Gaussian-to-Gaussian Registration

The coarse Gaussian-to-Gaussian registration problem is given by:

$$\underset{s_c\in\mathbb{R}_{++},R\in\mathrm{SO}(3),t\in\mathbb{R}^3}{\text{minimize}} \ \frac{1}{2}\sum_{(i,j)\in\mathcal{E}} w_{ij}\left(\|s_cRp_i+t-q_j\|_2^2 + \left\|s_cRH_{p_i}\Lambda_{p_i}-H_{q_j}\Lambda_{q_j}\right\|_F^2\right), \tag{4}$$

which can be solved efficiently in closed-form, which we show in Appendix D, with:

$$R_c^\star = U_c\Theta_cV_c^\top, \tag{5}$$

$$s_c^\star = \frac{\mathrm{trace}(\Theta_c\Sigma)}{\mathrm{trace}\left(W\check{P}^\top\check{P}+\sum_{(i,j)\in\mathcal{E}} w_{ij}\check{H}_{p_i}^\top\check{H}_{p_i}\right)}, \tag{6}$$

$$t_c^\star = \tilde{\mu}_\mathcal{Q} - s_c^\star R^\star\tilde{\mu}_\mathcal{P}, \tag{7}$$

where $U_c\Sigma_cV_c^\top = \check{Q}W\check{P}^\top + \sum_{(i,j)\in\mathcal{E}} w_{ij}\check{H}_{q_j}\check{H}_{p_i}^\top$, computed via the singular value decomposition (SVD), and $\Theta_c = \mathbf{diag}(1,1,\det(U_cV_c^\top))$. We define $\tilde{\mu}_\mathcal{P}$ and $\tilde{\mu}_\mathcal{Q}$ as the weighted average of the means of the Gaussians in $\mathcal{P}$ and $\mathcal{Q}$, with weights w_{ij} for Gaussian i in $\mathcal{P}$ and Gaussian j in $\mathcal{Q}$. Further, $\check{P}\in\mathbb{R}^{3\times N}$ and $\check{Q}\in\mathbb{R}^{3\times N}$ represent the *zero-centered* Gaussians in $\mathcal{P}$ and $\mathcal{Q}$, respectively, with the ith column of $\check{P}$ given by $\check{P}_i = p_i - \tilde{\mu}_\mathcal{P}$ and similarly for the jth column of $\check{Q}$. We introduce the terms $\check{H}_{p_i}\in\mathbb{R}^{3\times3}$ and $\check{H}_{q_j}\in\mathbb{R}^{3\times3}$ to simplify notation, with: $\check{H}_{p_i} = H_{p_i}\Lambda_{p_i}$ and $\check{H}_{q_j} = H_{q_j}\Lambda_{q_j}$. In addition, $W\in\mathbb{R}^{N\times N}$ denotes the diagonal weight matrix, $W_{kk} = w_k$, with $w_k = w_{ij}$, $\forall k = (i,j)\in\mathcal{E}$. We discuss the derivation of the closed-form solution to (4). Let the objective function of (4) be denoted by J. From the first-order optimality conditions:

$$\nabla_t J = \sum_{(i,j)\in\mathcal{E}} (w_{ij}s_cRp_i + t - q_j) = 0, \tag{8}$$

yielding the optimal translation:

$$t_c^\star = \tilde{\mu}_\mathcal{Q} - s_c^\star R^\star\tilde{\mu}_\mathcal{P}, \tag{9}$$

where $\tilde{\mu}_\mathcal{P}$ and $\tilde{\mu}_\mathcal{Q}$ denote the weighted average of the means of the Gaussians in $\mathcal{P}$ and $\mathcal{Q}$, with weights w_{ij} for Gaussian i in $\mathcal{P}$ and Gaussian j in $\mathcal{Q}$. By substituting the optimal value of t (9) in (4), we obtain the following optimization problem over s_c and R:

$$\underset{s_c\in\mathbb{R}_{++},R\in\mathrm{SO}(3)}{\text{minimize}} \ \frac{1}{2}\left\|(s_cR\check{P}-\check{Q})W^{\frac{1}{2}}\right\|_F^2 + \frac{1}{2}\sum_{(i,j)\in\mathcal{E}} w_{ij}\left\|s_cR\check{H}_{p_i}-\check{H}_{q_j}\right\|_F^2, \tag{10}$$

where $\check{P}\in\mathbb{R}^{3\times N}$ and $\check{Q}\in\mathbb{R}^{3\times N}$ represent the *zero-centered* Gaussians in $\mathcal{P}$ and $\mathcal{Q}$, respectively, with the ith column of $\check{P}$ given by $\check{P}_i = p_i - \tilde{\mu}_\mathcal{P}$ and similarly for the jth column of $\check{Q}$, and $\check{H}_{p_i} = H_{p_i}\Lambda_{p_i}$ and $\check{H}_{q_j} = H_{q_j}\Lambda_{q_j}$. Lastly, $W\in\mathbb{R}^{N\times N}$ denotes the diagonal weight matrix, $W_{kk} = w_k$, with $w_k = w_{ij}$, $\forall k = (i,j)\in\mathcal{E}$. Now, we can reformulate (10) as a trace-minimization problem, by leveraging the relation: $\|A\|_F^2 = \mathrm{trace}(A^\top A)$ for any real-valued matrix $A\in\mathbb{R}^{m\times n}$. Reformulating the problem as a trace-minimization problem enables us to decompose the norm-minimization problem (10) into a nested pair of subproblems: an outer subproblem over s_c and an inner subproblem over R. We can simplify the inner subproblem into the form:

$$\underset{R\in\mathrm{SO}(3)}{\text{minimize}} \ -\mathrm{trace}\left(R^\top\left(\check{Q}W\check{P}^\top + \sum_{(i,j)\in\mathcal{E}} w_{ij}\check{H}_{q_j}\check{H}_{p_i}^\top\right)\right), \tag{11}$$

which affords a closed-form optimal solution, with

$$R_c^\star = U_c\Theta_cV_c^\top, \tag{12}$$

where $U_c\Sigma_cV_c^\top = \check{Q}W\check{P}^\top + \sum_{(i,j)\in\mathcal{E}} w_{ij}\check{H}_{q_j}\check{H}_{p_i}^\top$, computed via the singular value decomposition (SVD), and $\Theta_c = \mathbf{diag}(1,1,\det(U_cV_c^\top))$. Using the first-order optimality condition, we can

compute the optimal scale after computing the optimal rotation from the outer subproblem given by:

$$\begin{aligned}\underset{s_c\in\mathbb{R}_{++}}{\text{minimize}}\ \frac{1}{2}s_c^2\text{trace}\left(W\check{P}^\top\check{P}+\sum_{(i,j)\in\mathcal{E}}w_{ij}\check{H}_{p_i}^\top\check{H}_{p_i}\right)\\ -s_c\text{trace}\left(R^\top\left(\check{Q}W\check{P}^\top+\sum_{(i,j)\in\mathcal{E}}w_{ij}\check{H}_{q_j}\check{H}_{p_i}^\top\right)\right),\end{aligned}\tag{13}$$

yielding the optimal solution:

$$s_c^\star=\frac{\text{trace}(\Theta_c\Sigma)}{\text{trace}\left(W\check{P}^\top\check{P}+\sum_{(i,j)\in\mathcal{E}}w_{ij}\check{H}_{p_i}^\top\check{H}_{p_i}\right)}.\tag{14}$$

For brevity, we omit further analysis of the optimality of the solution and refer interested readers to [31] for the proof of a related problem, which applies to the problem considered in this work.

Appendix E Fine Photometric Registration

Image Generation and Matching. The fine registration procedure begins with the identification of a set of images with common features across the source and target maps, constituting arguably the most important step of the fine registration procedure. In particular, the feasibility of the fine registration procedure hinges on matching corresponding features across all images in the set. In general, identifying good candidate images for the matching process is challenging, especially without any prior knowledge of the region of overlap between the source and target maps. To address this challenge, we leverage the semantic submap extracted in the first stage of SIREN to identify a region of overlap between the source and target maps. Subsequently, we exploit novel-view synthesis in Gaussian Splatting to render images at corresponding poses in both maps, by transforming the camera pose in one map to the associated camera pose in the other map, utilizing the coarse registration result to compute the corresponding pose. With this approach, not only do the resulting images contain common features from the overlapping region, the images also contain a dense set of features, associated with the semantic submap. However, the pair of rendered images may not contain sufficient matches, which could degrade the accuracy of the fine registration procedure. To mitigate this risk, we harness image semantics in vision foundation models to evaluate the similarity between each pair of rendered images, retaining only sufficiently similar images. In this work, we use CLIP along with the cosine-similarity metric, given that the image embeddings of CLIP were trained with a cosine-similarity loss function; however, other vision foundation can also be used, e.g., [12].

Image Registration and Triangulation. Following the generation of corresponding images, we extract features from all images using the learned feature extractors NetVLad [53] for global image-level descriptors and SuperPoint [54] for local features, which we found to be more robust compared to classical feature extractors, e.g., SIFT [55]. Subsequently, we match features across all images using [54]. From corresponding features, we estimate the relative pose of the camera and the estimated 3D locations of the feature points via image registration and triangulation, yielding an initial estimate of the camera pose associated with each image in a common reference frame.

Bundle Adjustment. The image registration step does not always provide high-accuracy camera pose estimates. Hence, we refine the estimated camera poses via bundle adjustment, i.e., we optimize over the camera pose and the 3D locations of the feature points jointly through non-linear optimization. For brevity, we do not discuss the bundle adjustment problem in greater detail, noting its extensive discussion in prior work, e.g., [56]. Although non-convex, the optimization problem can be solved efficiently via iterative methods, such as the Levenberg-Marquardth method, which we employ in this work. From the bundle adjustment optimization problem, we compute the camera poses associated with each image in an arbitrary common frame $\mathcal{B}$. Given the camera poses expressed in $\mathcal{A}$ and the corresponding poses in the source and target maps, we can compute an optimal transformation for

registering $\mathcal{A}$ to either the source frame (frame $\mathcal{B}_s$) or the target frame (frame $\mathcal{B}_t$) from the following registration problem in $\mathrm{SE}(3)$:

$$\underset{s_f \in \mathbb{R}_{++}, R \in \mathrm{SO}(3), t \in \mathbb{R}^3}{\text{minimize}} \ \frac{1}{2} \sum_{(i,j) \in \mathcal{V}} \left(\| s_f R a_i + t - b_j \|_2^2 \ + \beta_{ij} \left\| R R_{c_i} - R_{d_j} \right\|_F^2 \right), \tag{15}$$

where s_f, R, and t denote the scale, rotation, and translation parameters, respectively, $\mathcal{V}$ denotes the set of edges between the camera poses expressed in $\mathcal{A}$ and the corresponding poses in either the source or target frame, with $a_i \in \mathbb{R}^3$ and $b_j \in \mathbb{R}^3$ denoting the origin of the camera in $\mathcal{A}$ and the origin of the camera in the frame $\mathcal{B}_s$ or $\mathcal{B}_t$, respectively, and R_{a_i} and R_{b_j} denoting the associated rotation matrices. We introduce the weight parameter $\beta_{ij} \in \mathbb{R}_{++}$, which determines the contribution of the rotation-error component. In general, the optimization problem in (15) cannot be solved in closed-form. Solving (15) generally requires an iterative optimization method, e.g., sequential convex programming methods or Riemannian optimization methods. However, as β_{ij} approaches zero, $\forall (i,j) \in \mathcal{V}$, the optimal solution (15) approaches a limit point, with:

$$R_f^\star \to U_f \Theta_f V_f^\top, \ \ s_f^\star \to \frac{\mathrm{trace}(\Theta_f \Sigma_f)}{\mathrm{trace}\left(\check{A}^\top \check{A}\right)}, \ \ t_f^\star \to \mu_{\mathcal{B}} - s_f^\star R_f^\star \mu_{\mathcal{A}}, \tag{16}$$

where $U_f \Sigma_f V_f^\top = \check{B} \check{A}^\top$, $\Theta_f = \mathbf{diag}(1, 1, \det(U_f V_f^\top))$, $\mu_{\mathcal{A}}$ and $\mu_{\mathcal{B}}$ denote the mean of the camera origins in frames $\mathcal{A}$ and $\mathcal{B}$, respectively, and the ith column of $\check{A} \in \mathbb{R}^{3 \times N}$ and the jth column of $\check{B} \in \mathbb{R}^{3 \times N}$ are given by $a_i - \mu_{\mathcal{A}}$ and $b_j - \mu_{\mathcal{B}}$, respectively. The limit point follows from the derivation in Section 3.2 and [31]. We can compose the pairwise transformations between frame $\mathcal{A}$ and frames $\mathcal{B}_s$ and $\mathcal{B}_t$ to compute a transformation from $\mathcal{B}_s$ and $\mathcal{B}_t$. We apply the resulting transformation to the source map to express the source and target maps in a common frame and subsequently merge the resulting maps to obtain a composite GSplat map. Following the registration procedures, the composite map can be finetuned with new or existing data, which we explore in our experiments in Appendix F.5. We summarize the procedures in SIREN in Algorithm 1.

Algorithm 1: SIREN: Multi-Robot Map Registration

Input: Local GSplat Maps $\mathcal{G}_1, \mathcal{G}_2$;
Output: Fused GSplat Map $\mathcal{G}_f$;
// Semantic Feature Extraction and Matching
Correspondence Set $\mathcal{C} \leftarrow \mathrm{GetCorrespondence}(\mathcal{G}_1, \mathcal{G}_2)$;
// Coarse Registration
// Compute the Optimal Rotation
$R_c^\star \leftarrow$ Procedure (5);
// Compute the Optimal Scale
$s_c^\star \leftarrow$ Procedure (6);
// Compute the Optimal Translation
$t_c^\star \leftarrow$ Procedure (7);
// Fine Registration
// Get Images
$\mathcal{D}_s \leftarrow \mathrm{Render}(\mathcal{G}_1, \mathcal{G}_2, R_c^\star, s_c^\star, t_c^\star)$;
// Refine Transformation
$(R_f^\star, s_f^\star, t_f^\star) \leftarrow$ Procedure (16);
// Fuse Local Maps
$\mathcal{G}_f \leftarrow \mathrm{Fuse}(\mathcal{G}_1, \mathcal{G}_2, R_f^\star, s_f^\star, t_f^\star)$;

Appendix F Experiments

We report the photometric performance of each registration method with the Mip-NeRF360 dataset and the data collected by the robots in our experiments and provide further discussion of the experimental results. In addition, we present ablations, examining the different components of SIREN. Lastly, we explore finetuning the resulting composite maps for higher visual fidelity.

Experimental Setup and Metrics. For the real-world robot data, we utilize the Unitree Go1 Quadruped and a Modal AI drone with an onboard camera and the Franka Panda manipulator with a wrist camera to collect RGB images. In addition, we evaluate all methods on the real-world scenes in the Mip-NeRF360 dataset [43], a state-of-the-art benchmark dataset for neural rendering. We train the GSplat models using the original implementation provided by the authors of [1] for baselines which require this pipeline and utilize Nerfstudio [57] for SIREN. We execute SIREN on a desktop computer with a 24GB NVIDIA GeForce RTX 3090 GPU and the baselines on an H20 GPU after training the GSplat maps for 30000 iterations. We note that in robotics, the geometric fidelity of robot's map is of significant importance for effective localization and collision avoidance. Hence, we compare all methods in terms of the rotation error (RE) [deg.], translation error (TE), and scale error (SE) [in non-metric units] attained by each method, in addition to the computation time (CT) [sec.]. We can then compute the mean for each error metric in each dataset. RE represents the absolute relative rotation angle between the estimated and ground-truth rotation matrices, i.e., the geodesic distance between the estimated and ground-truth rotation matrices on the $\mathrm{SO}(3)$ manifold computed via the trace function. SE and TE represent the ℓ_2-distance between the estimated scale and translation and the ground-truth scale and translation. Moreover, we examine the photometric quality of the fused maps generated by each method, computing the peak signal-to-noise ratio (PSNR), the structural similarity index measure (SSIM), and the learned perceptual image patch similarity (LPIPS), standard metrics in the computer vision community for assessing visual fidelity. We provide color-coded results for each metric with the red shade denoting the top-performing statistic, the yellow shade denoting the second-best, and the green shade denoting the third-best. In all the registration methods, we do not pre-process the individual submaps to remove floaters (i.e., non-existing geometry). Consequently, floaters present in these submaps are retained in the fused map.

Specifying the Ground-Truth Transformation. In our evaluations, e.g., in the benchmark datasets, we compute the camera poses over all the images in a common (global) frame using structure-from-motion, for each dataset. Next, we partition the dataset into subsets with different level of overlap and apply a random transformation (i.e., rotation, translation, and scaling) to the subset of the data to create a local frame, capturing the broad range of differences in rotation, scaling, and translation that robots might encounter in multi-robot mapping. These transformation parameters represent the ground-truth transformation from the local frame to the global frame. By composing the ground-truth transformations between the local and global frames, we compute the ground-truth source-to-target transformation, i.e., from the local frame of the source submap to the local frame of the target submap.

F.1 Mip-NeRF360 Dataset

Geometric Evaluation. In the *Room* scene, PhotoReg achieves the lowest rotation error by a factor of about 1.47x but also achieves the largest translation and scale errors. Based on the results across all scenes, SIREN almost always consistently outperforms competing methods. From Table 1, RANSAC-GR and FGR achieve the fastest computation times; however, RANSAC-GR and FGR do not generally achieve consistently low geometric errors. Although SIREN is slower than the classical point-cloud registration algorithms and GaussReg, SIREN generally outperforms these methods in accuracy by significant margins. Moreover, about 40% to 50% of the total computation time of SIREN is spent on the semantics extraction procedure. Hence, the total computation time can be significantly improved by utilizing faster semantics distillation methods, e.g., [25].

In our geometric evaluations, we note that the scale errors of the baseline methods are quite high, which we discuss here. The baseline methods essentially rely exclusively on geometric and visual details in each map to compute a transformation between the maps. As a result, the baseline methods are more likely to achieve higher scale errors when the maps lack a sufficiently high number of distinct geometric and visual features. For example, the baseline methods achieve notably lower scale errors when evaluated in the *Playroom* dataset, which contains many interesting visual features, shown in Fig. 4. In contrast, the baseline methods have notably larger scale errors in the *Truck* dataset, which contain objects with fewer colors and other interesting geometric detail. Further, the translation parameter estimated by each method accounts for the scale and relative rotation between both maps.

Photometric Evaluation. In the *Truck* scene, RANSAC-GR achieves the best mean PSNR and SSIM scores. Although this finding may appear inconsistent with the geometric results presented in Table 1, the high standard deviation of each of the scores achieved by RANSAC-GR (about 2x to 3x larger than that of SIREN) suggests that the geometric and photometric performance metrics for this scene might actually be consistent, indicating that the fused map generated by RANSAC-GR warrants further examination. We provide rendered images from the fused map generated by RANSAC-GR compared to the ground-truth images in Figure 5 to examine the registration results of RANSAC-GR. From Figure 5, we note that RANSAC-GR fails to accurately register the left and right sides of the truck. In fact, the left side of the truck is missing in the bottom panel associated with RANSAC-GR in Figure 5. However, this failure mode is not fully captured by the mean score of the photometric performance metrics, since the rendered images of the right side of the truck (shown in the top panel in Figure 5) look quite similar to the corresponding ground-truth images. In conclusion, RANSAC-GR does not accurately register the individual GSplat maps, despite achieving the highest mean PSNR and SSIM scores in the *Truck* scene.

From Figure 3, the fused map in GaussReg and Colored-ICP contain duplicate objects due to inaccurate registration of the individual maps. In contrast, SIREN-R provides greater accuracy. Likewise, SIREN-R achieves the highest-fidelity rendering in the *Truck* scene with consistent geometry, whereas Colored-ICP fails to register the left and right sides of the truck. Although GaussReg fuses both sides of the truck, GaussReg fails to compute a high-accuracy transform, resulting in the artifacts visible in Figure 3. Although PhotoReg registers the cargo bed of the truck in both maps, PhotoReg fails to align the truck accurately in terms of the rotation transform, with the front end of the truck in one map registered to the rear end of the truck in the other map. Finally, in the *Room* scene, whereas SIREN-R generates high-fidelity rendered images, other methods fail to accurately register the individual maps. In particular, Colored-ICP generates a fused map with duplicate objects, e.g., the piano and the table, indicated by the green squares, while PhotoReg and GaussReg generate fused maps with notable artifacts.

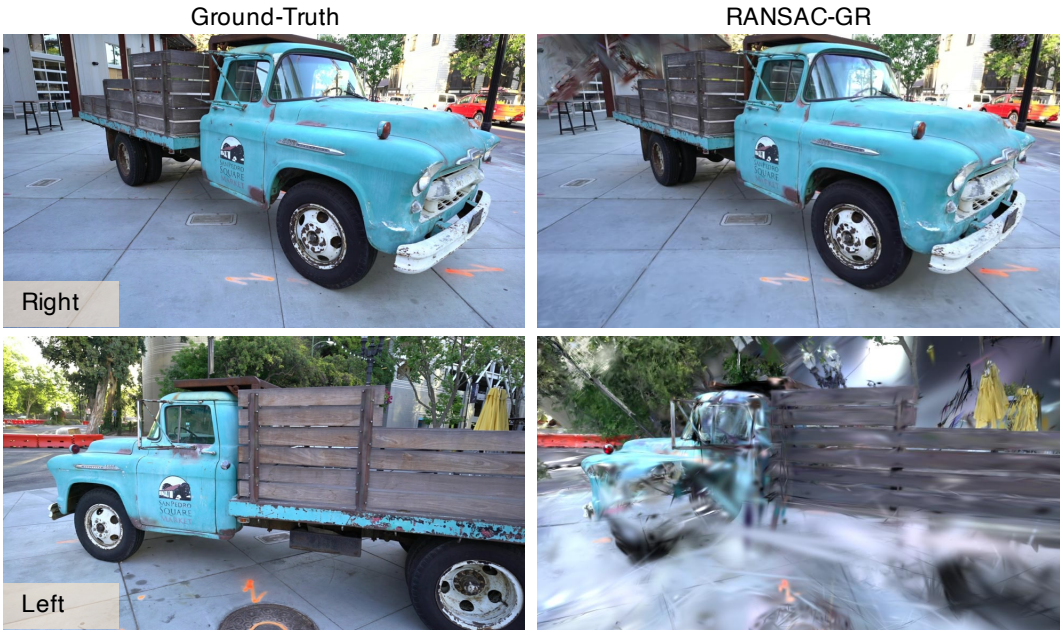

Figure 5: Although RANSAC-GR achieves the highest mean PSNR and SSIM scores and the lowest LPIPS score in the *Truck* scene, RANSAC-GR does not accurately register the individual GSplat maps. While the right side of the truck in the RANSAC-GR fused map looks similar to the ground-truth image (shown in the top panel), the left side of the truck is missing (shown in the bottom panel). The standard deviation of the PSNR, SSIM, and LPIPS scores achieved by RANSAC-GR reflects the actual registration performance of the method.

F.2 Mobile-Robot Mapping

Photometric Evaluation. From Figure 6, as highlighted by the green squares, SIREN-R generates composite maps that are consistent with the ground-truth, unlike GaussReg and PhotoReg. The fused maps generated by GaussReg and PhotoReg contain conspicuous artifacts due to inaccurate registration of the individual maps created by the robots, especially in the *Kitchen* scene. PhotoReg fails to sufficiently register the individual maps, resulting in blurry renderings, with few recognizable features, e.g., in the *Workshop* scene. In the *Apartment* scene, the rendered images from GaussReg contain duplicate objects, unlike those of SIREN-R, which have accurate geometric detail.

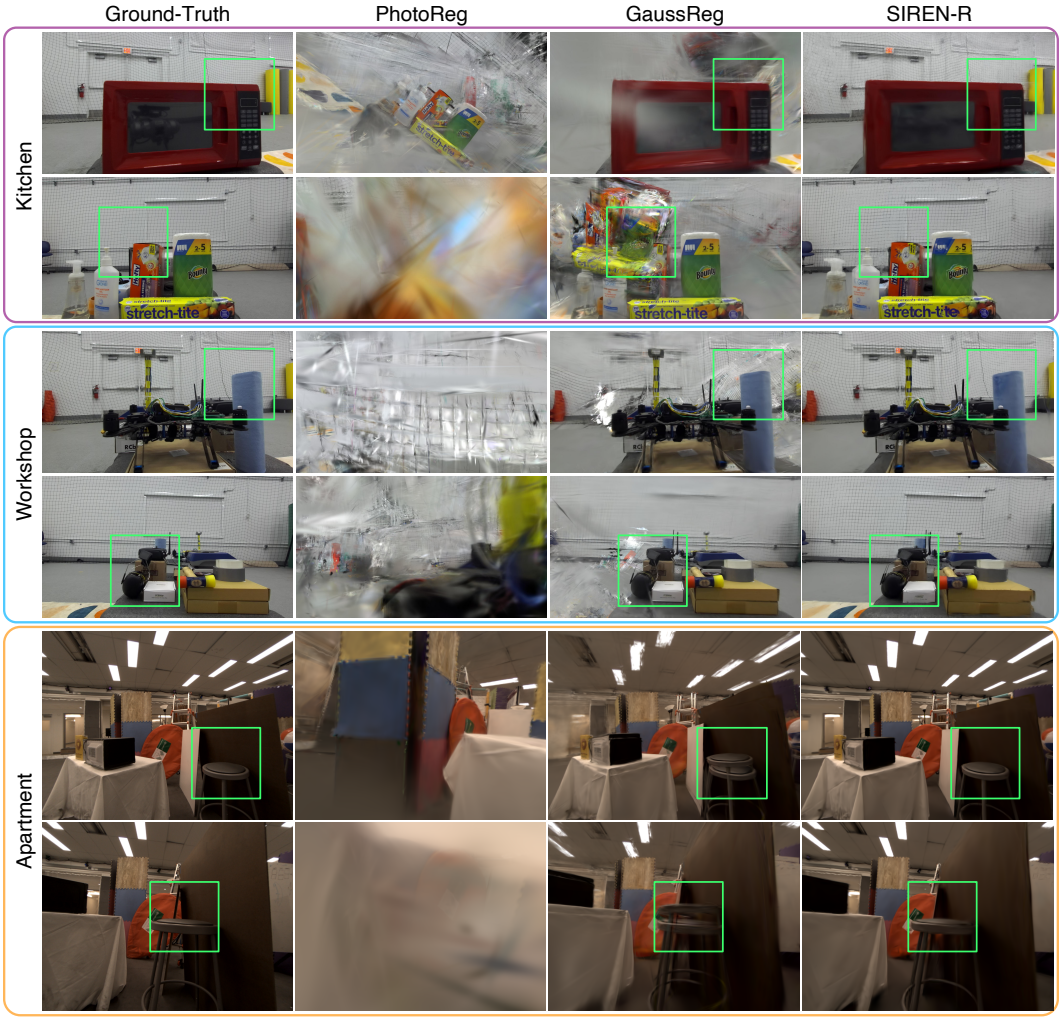

Figure 6: Rendered images from the fused GSplat maps of the *Kitchen*, *Workshop*, and *Apartment* scenes mapped by a quadruped and drone. Unlike other competing methods, SIREN generates fused GSplat maps of high visual fidelity, e.g., in the regions indicated by the green squares.

Real-time Applicability. We emphasize that the current Python implementation of SIREN is not optimized for speed unlike the baselines (explaining its slower runtimes), e.g., ICP, FGR, and RANSAC are all written in highly-optimized C++ code. Through code optimizations (e.g., code/model compilation), we can achieve orders of magnitude speedup. Nevertheless, in multi-robot mapping, the map fusion process is often performed at a much lower frequency than the local mapping process executed by a robot. Hence, our method can be employed in real-time multi-robot mapping problems.

F.3 Tabletop Mapping with Multiple Manipulators

We demonstrate the effectiveness of SIREN in tabletop robotics tasks with fixed-base manipulators, which often require the robots to map the scene prior to the task, e.g., in manipulation [2, 6]. In Figure 7, we provide an example with two Franka robots, each with a wrist camera. Due to the limited workspace of each robot, visualized in Figure 7, mapping often requires the assistance of a human-operator [6] or ad-hoc solutions such as hardware improvisation, e.g., using selfie sticks [2]. By enabling the fusion of GSplat maps trained individually by each robot, SIREN effectively eliminates these limitations. In other words, with SIREN, each robot can train a submap within its reachable workspace and still recover the global map via registration with SIREN. In Figure 8, we show the submaps trained by each robot. As expected, each robot has a high-fidelity submap within the confines of its reachable workspace, evident in the first-two images in the left robot's map and the last-two images in the right's robot map in Figure 8. In areas outside of its reachable workspace, the robot's map fails to represent the real world accurately, visible in the last-two images in the left robot's map and the first-two images in the right's robot map. With SIREN, each robot obtains a higher-fidelity map over a much broader region of the environment. However, floaters present in the submaps can degrade the quality of the fused map in certain regions. To address this challenge, we finetune the fused map for about 70.98 secs using images generated entirely from the GSplat maps, i.e., we do not require any real-world data. We provide rendered images from the finetuned fused map in Figure 8, showing near-perfect reconstruction of the global scene. We explore the finetuning procedure in Appendix F.5.

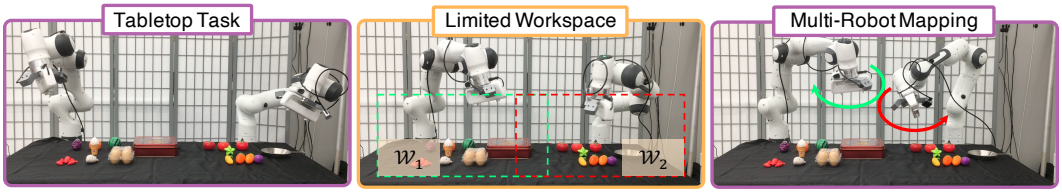

Figure 7: Tabletop robotics tasks, e.g., manipulation, generally require robots to map the scene prior to completing the task. However, the limited workspace of each robot often demands assistance from a human-operator or improvised hardware, e.g., selfie sticks. SIREN eliminates these challenges, via registration of the local maps trained by each robot to construct a global map consistent with the real-world.

F.4 Ablations

We examine the constituent registration steps in SIREN, namely: the coarse Gaussian-to-Gaussian and fine photometric registration procedures, assessing the accuracy of the registration result generated by each procedure. We denote the variant of SIREN with coarse registration performed without RANSAC and fine registration by SIREN-CNR. Likewise, we denote the variant of SIREN with coarse registration performed using RANSAC but without fine registration by SIREN-CR. We compute the geometric and photometric performance metrics for each of these variants and report the results in Table 5 and Table 6, respectively. We also report the performance of SIREN-NR and SIREN-R from Table 1 and Table 2 for easy reference. From Table 5, we note that the fine registration step in SIREN notably improves the rotation error to sub-degree errors, achieving about 2x smaller translation errors and in some cases, 100x smaller translation errors. Likewise, the fine registration step generally results in much smaller scale errors, although not necessarily in all cases, as reflected in the *Truck* scene. Similarly, the variants of SIREN with fine registration (i.e., SIREN-NR and SIREN-R) achieve notably higher photometric performance, especially in the *Playroom* and *Room* scenes, reported in Table 6. In general, the coarse registration step brings corresponding objects in both GSplat maps into close proximity in the fused map. However, the resulting fused map lacks precise geometric detail, degrading its visual fidelity. After the coarse registration step, the fine registration procedure refines the transformation parameters for precise alignment of the individual maps, ultimately generating a photorealistic fused map.

Figure 8: Rendered images of the local maps of a tabletop scene trained by two manipulators. The maps provide high-fidelity reconstructions within the workspace of each robot, but fail to represent the real-world in regions outside the workspace. SIREN fuses the local maps to generate a high-fidelity global map consistent with the entirety of the scene, especially after finetuning on data rendered directly from the GSplat to remove floaters, without any interaction with the real-world, as indicated by the green squares.

Although SIREN-CR does not always outperform RANSAC-GR in Table 1, we observed empirically that the performance of RANSAC-GR has a high variance, posing a challenge for the fine registration step, which requires a sufficient number of corresponding features between rendered frames across the individual maps to compute a solution. Moreover, ICP and its variants tend to converge to a local optimum, close to the solution used for initialization. As a result, these methods generally fail to provide a sufficiently good initialization for the fine registration procedure. The coarse registration step in SIREN relies significantly on the semantics extracted from the map to overcome these limitations, leveraging the inherent semantics to register corresponding objects at a sufficient level of accuracy for fine registration.

Table 5: Geometric Performance: Ablation of the Coarse Gaussian-to-Gaussian and Fine Photometric Registration in SIREN.

Methods	*Playroom*				*Truck*				*Room*			
	RE ↓	TE ↓	SE ↓	CT ↓	RE ↓	TE ↓	SE ↓	CT ↓	RE ↓	TE ↓	SE ↓	CT ↓
SIREN-CNR	22.72	454.2	482.1	20.20	49.98	355.4	55.06	24.17	20.50	474.0	371.9	17.27
SIREN-CR	21.15	324.2	51.94	20.47	0.804	7.691	7.744	26.32	24.07	381.8	155.1	17.58
SIREN-NR	0.348	4.860	0.282	41.16	0.511	8.07	9.581	53.42	0.381	2.648	1.016	40.24
SIREN-R	0.170	1.933	0.170	39.73	0.413	6.845	2.548	52.47	0.237	3.289	2.673	39.71

To avoid any confusion, we emphasize that the ablations without a fine registration step are not designed to outperform the baselines in this work. In particular, we emphasize that the baselines in this paper are *fine* registration methods, e.g., Colored-ICP and ICP. Colored-ICP and ICP are

Table 6: Photometric Performance: Ablation of the Coarse Gaussian-to-Gaussian and Fine Photometric Registration in SIREN.

Methods	*Playroom* PSNR ↑	*Playroom* SSIM ↑	*Playroom* LPIPS ↓	*Truck* PSNR ↑	*Truck* SSIM ↑	*Truck* LPIPS ↓	*Room* PSNR ↑	*Room* SSIM ↑	*Room* LPIPS ↓
SIREN-CNR	12.0 ± 3.2	0.67 ± 0.11	0.65 ± 0.17	11.4 ± 2.4	0.52 ± 0.11	0.63 ± 0.22	6.9 ± 5.9	0.72 ± 0.11	0.39 ± 0.15
SIREN-CR	13.7 ± 3.4	0.65 ± 0.14	0.58 ± 0.19	16.2 ± 2.2	0.56 ± 0.14	0.31 ± 0.07	11.7 ± 1.9	0.48 ± 0.12	0.64 ± 0.10
SIREN-NR	26.3 ± 3.1	0.87 ± 0.05	0.17 ± 0.06	15.4 ± 1.7	0.52 ± 0.12	0.35 ± 0.05	24.8 ± 3.3	0.83 ± 0.04	0.22 ± 0.06
SIREN-R	28.3 ± 2.9	0.90 ± 0.04	0.15 ± 0.06	16.4 ± 2.4	0.57 ± 0.13	0.31 ± 0.07	24.1 ± 3.1	0.82 ± 0.05	0.23 ± 0.06

first initialized using a RANSAC-based coarse registration method. Hence, Colored-ICP and ICP generally outperform our coarse registration method, when they converge. Moreover, our implementation of the coarse registration procedure is very basic compared to the fully-featured open-source implementations of Colored-ICP and ICP used in our work. For example, Colored-ICP and ICP utilize a more robust implementation of RANSAC and more informative geometric features, such as FPFH descriptors. Although we could integrate these robust code implementations into our coarse registration method, we note that our fine registration method already provides robustness to inaccurate transformations computed by our coarse registration method. We highlight that a more appropriate comparison would be between Colored-ICP, ICP, and our fine registration method. In fact, our experiments demonstrate that our fine registration method generally outperforms Colored-ICP and ICP.

F.5 Finetuning

SIREN does not pre-process the local GSplat maps before registration of the maps, resulting in the retention of floaters in the fused map whenever floaters exist in the local maps. Here, we examine finetuning the fused map with rendered images from the local maps to remove visual artifacts, without requiring access to the data used in the training the local GSplat maps, i.e., we do not require access to the real-world camera images and poses. To finetune the fused map without access to the original dataset, we select camera poses expressed in the local frames of the local GSplat maps (e.g., randomly or via an informed approach) and render images from these maps at these camera poses. Subsequently, we transform the set of camera poses from their associated local frames to the frame of the fused map using the transformation parameters computed by SIREN. We construct a finetuning dataset from the set of images and associated camera poses, which we use in finetuning the fused map.

In Table 7, we provide the photometric scores of the fused GSplat map from SIREN-R before and after finetuning and the ground-truth GSplat map. We train the ground-truth GSplat map using the combined training datasets used in training the local GSplat maps (i.e., the real-world camera images and poses, not the set of rendered images generated from the local GSplat maps), representing the ideal composite GSplat model. The computation time in Table 7 represents the total training time for the ground-truth map and the total time used in finetuning the fused map. Table 7 indicates that finetuning the fused map improves the PSNR, SSIM, and LPIPS scores compared to that of the pre-finetuned map. Specifically, in less than 90 seconds, finetuning reduces the gap between the photometric scores of the ground-truth map and the photometric scores of the fused map by about 20% to 40%. The relative improvements provided by finetuning the fused map depend on the finetuning data used, an area for future research. We provide rendered images from the fused GSplat map computed by SIREN-R, before and after finetuning, and the corresponding images in the ground-truth fused map in Figure 9. Across all three scenes, finetuning the fused map removes floaters and other artifacts, e.g., in the regions indicated by the green squares, ultimately resulting in higher PSNR and SSIM scores, as reported in Table 7.

Table 7: Photometric performance after finetuning SIREN-R.

Methods	*Playroom* PSNR ↑	*Playroom* SSIM ↑	*Playroom* LPIPS ↓	*Playroom* CT ↓	*Truck* PSNR ↑	*Truck* SSIM ↑	*Truck* LPIPS ↓	*Truck* CT ↓	*Room* PSNR ↑	*Room* SSIM ↑	*Room* LPIPS ↓	*Room* CT ↓
Ground-Truth	36.3 ± 3.5	0.96 ± 0.03	0.09 ± 0.05	721.1	26.4 ± 1.4	0.89 ± 0.02	0.10 ± 0.01	601.7	34.1 ± 1.7	0.94 ± 0.02	0.12 ± 0.04	840.1
Pre-Finetuning	29.1 ± 3.3	0.91 ± 0.04	0.15 ± 0.06	N/A	16.8 ± 2.5	0.61 ± 0.10	0.30 ± 0.07	N/A	22.5 ± 2.5	0.79 ± 0.05	0.26 ± 0.06	N/A
Post-Finetuning	30.8 ± 2.6	0.92 ± 0.04	0.14 ± 0.06	72.69	21.1 ± 1.8	0.69 ± 0.1	0.23 ± 0.04	86.26	26.0 ± 3.6	0.83 ± 0.09	0.22 ± 0.08	79.78

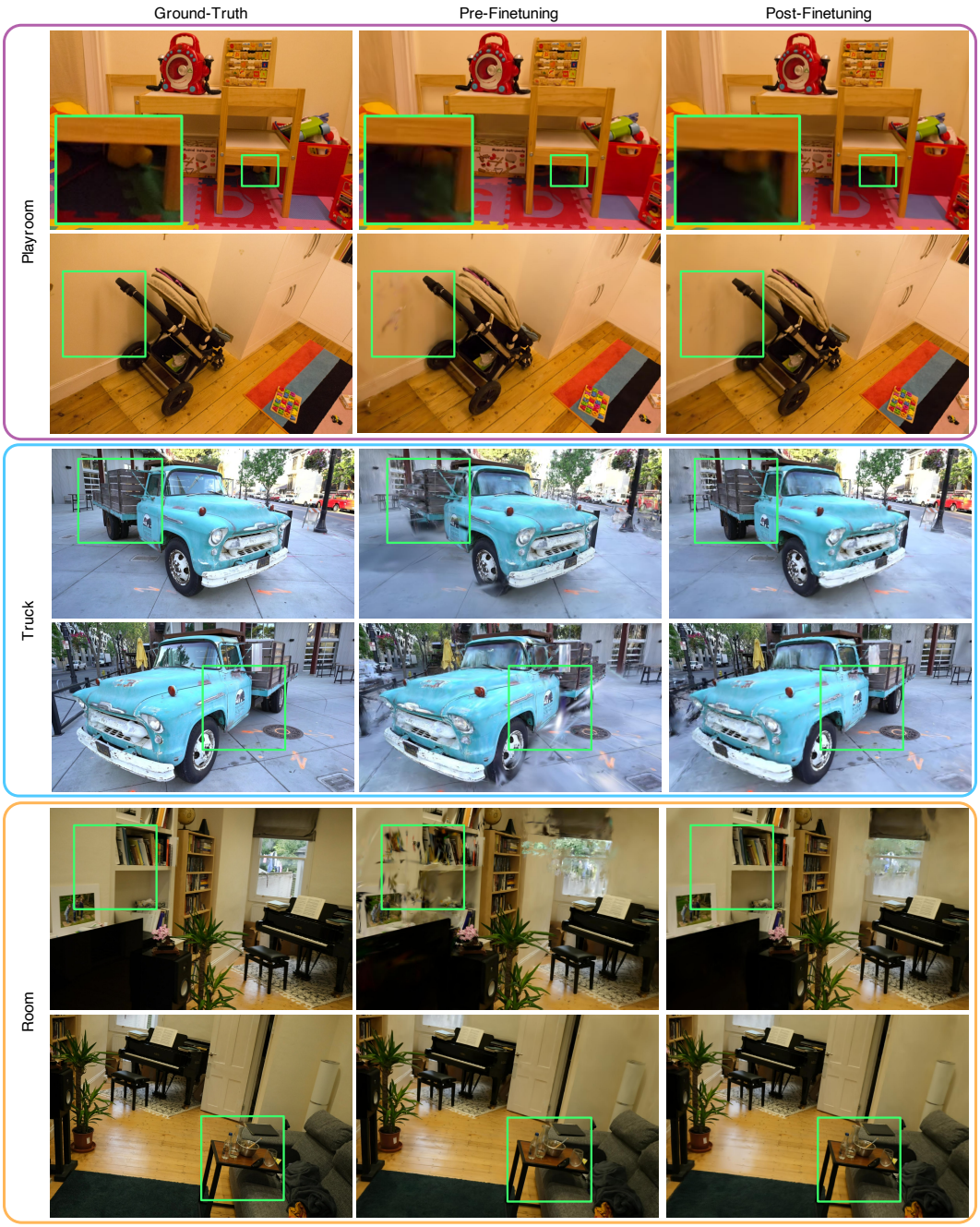

Figure 9: Rendered images from the fused GSplat maps generated by SIREN-R before and after finetuning, in the *Playroom*, *Truck*, and *Room* scenes. Finetuning improves the visual fidelity of the fused map, removing floaters and other artifacts.

For clarity, we reiterate that in our comparisons with the baseline methods, we *do not* fine-tune any method after the map fusion procedure. We explore fine-tuning only in this subsection.