arXiv:2503.01582v3 [cs.CV] 29 Jul 2025

# Category-level Meta-learned NeRF Priors for Efficient Object Mapping

Saad Ejaz[1], Hriday Bavle[1], Laura Ribeiro[1], Holger Voos[1], and Jose Luis Sanchez-Lopez[1]

***Abstract*— In 3D object mapping, category-level priors enable efficient object reconstruction and canonical pose estimation, requiring only a single prior per semantic category (e.g., chair, book, laptop, etc.). DeepSDF has been used predominantly as a category-level shape prior, but it struggles to reconstruct sharp geometry and is computationally expensive. In contrast, NeRFs capture fine details but have yet to be effectively integrated with category-level priors in a real-time multi-object mapping framework. To bridge this gap, we introduce PRENOM, a Prior-based Efficient Neural Object Mapper that integrates category-level priors with object-level NeRFs to enhance reconstruction efficiency and enable canonical object pose estimation. PRENOM gets to know objects on a first-name basis by meta-learning on synthetic reconstruction tasks generated from open-source shape datasets. To account for object category variations, it employs a multi-objective genetic algorithm to optimize the NeRF architecture for each category, balancing reconstruction quality and training time. Additionally, prior-based probabilistic ray sampling directs sampling toward expected object regions, accelerating convergence and improving reconstruction quality under constrained resources. Experimental results highlight the ability of PRENOM to achieve high-quality reconstructions while maintaining computational feasibility. Specifically, comparisons with prior-free NeRF-based approaches on a synthetic dataset show a 21% lower Chamfer distance. Furthermore, evaluations against other approaches using shape priors on a noisy real-world dataset indicate a 13% improvement averaged across all reconstruction metrics, and comparable pose and size estimation accuracy, while being trained for 5× less time.**

**Code available at: https://github.com/snt-arg/PRENOM**

## I. Introduction

Object-level mapping has gained significant attention for its ability to separate objects from cluttered environments for high-level robotic tasks. Recent works use compact primitives such as cuboids [1] and quadrics [2] to approximate object bounds, while reconstructing dense geometry using Signed Distance Fields (SDFs) [3], [4], Neural Radiance Fields (NeRFs) [5], [6], and Gaussian splats [7]. Typical methods operate without explicit priors and treat all objects uniformly, overlooking object semantics. In contrast, 3D priors capture common geometric properties, providing valuable cues for reconstruction while also enabling canonical pose estimation of objects.

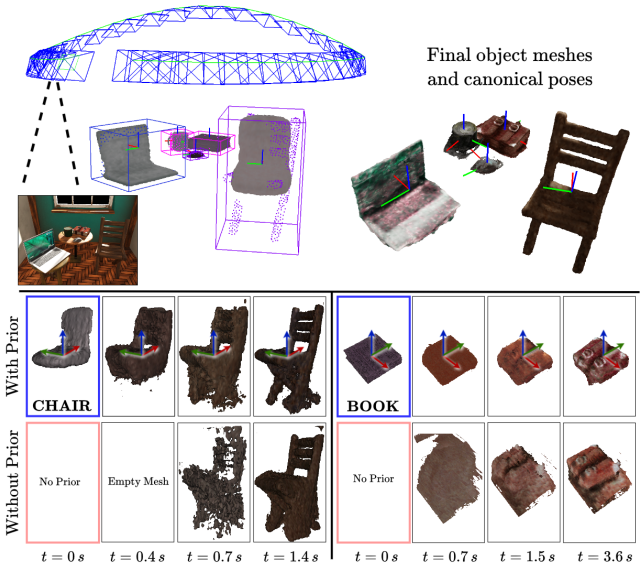


Fig. 1: PRENOM uses category-level priors to accelerate object-level NeRF training and localize objects in a canonical frame. Top left shows priors loaded during mapping (displayed all at once for illustration). PRENOM operates online, training objects in parallel while initializing them sequentially as they are observed. The bottom rows demonstrates how priors provide a reasonable initial estimate, enabling faster and more accurate reconstruction. Time indicates dedicated GPU training time.

These priors can be broadly classified into three categories. Instance-level priors [8], [9] offer high accuracy, but require knowledge of the environment, through precise CAD models of each object instance, limiting their applicability to specialized use cases. On the other end, generalized priors [10] use a single prior for all objects but rely on bulky diffusion models, making them computationally intensive even for a single object. Category-level priors are a suitable trade-off as they rely solely on semantic labels, and thus can work with unseen objects within predefined categories of interest. Recent works represent such priors using variational autoencoders [11], shape embeddings [12], and DeepSDFs [13]–[15]. While these methods offer reasonable global shape and pose estimation, they are computationally demanding, requiring high-end GPUs; use identical prior designs for all

[1]Authors are with the Automation and Robotics Research Group, Interdisciplinary Centre for Security, Reliability, and Trust (SnT), University of Luxembourg, Luxembourg. Holger Voos is also associated with the Faculty of Science, Technology, and Medicine, University of Luxembourg, Luxembourg. {saad.ejaz, hriday.bavle, laura.ribeiro, holger.voos, joseluis.sanchezlopez}@uni.lu

This research was funded, in whole or in part, by the Luxembourg National Research Fund (FNR) under the DEUS Project (Ref. C22/IS/17387634/DEUS) and the MR-Cobot Project (Ref. 18883697/MR-Cobot).

*For the purpose of open access, and in fulfillment of the obligations arising from the grant agreement, the author has applied a Creative Commons Attribution 4.0 International (CC BY 4.0) license to any Author Accepted Manuscript version arising from this submission.

categories, ignoring differences in shape complexity; and in most scenarios struggle with reconstructing sharp geometry.

To address these limitations, we present PRENOM, a Prior-based *Efficient* Neural Object Mapper that represents objects by lightweight NeRF models, with meta-learned [16] parameter initializations as category-level priors, within a real-time RGB-D multi-object mapping framework. Having been trained on large shape datasets, these priors can localize objects within a canonical frame while also benefiting from the high-fidelity reconstructions typically associated with NeRFs. Further integrating them with neural architectures optimized per-category and prior-based probabilistic ray sampling, we achieve faster training convergence (see Figure 1), enabling accurate reconstructions with limited GPU resources.

The primary contributions of this paper are as follows:

- An online object mapping framework that uses lightweight object-level NeRFs with category-level meta-learned priors to enable accurate object reconstruction on resource-constrained systems.
- A pipeline for optimizing NeRF architectures per object category to achieve a balance between reconstruction performance and training speed.
- Integration of prior-based probabilistic ray sampling in object-level NeRFs for improved convergence leading to better reconstruction quality.

## II. Related Works

### A. 3D Object Mapping

Object-level mapping has been extensively explored in the context of SLAM, where camera poses and object states are jointly estimated by enforcing multi-view consistency [1], [2], [11], [13]. Alternatively, it has also been studied independently, taking advantage of known camera poses to focus purely on object representation and reconstruction [3], [5], [6], [10], [12], [15], [17]. While the former approach prioritizes simple, optimizable object representations to support joint estimation, the latter decouples localization and mapping, shifting the focus on improving reconstruction fidelity. To achieve this, implicit neural object representations, such as object-level SDFs [3], [4] and object-level NeRFs [5], [6], [17], [18], have been widely adopted to generate high-quality meshes. Our approach falls in this latter category.

Considering only object representation, RO-MAP [6] and [17] are the closest to our approach; however, their reliance on random weight initialization and uniform ray sampling results in slow training, requiring a high-end GPU for real-time multi-object mapping. Moreover, scale ambiguity in the monocular RO-MAP results in noisy object pose estimation, which is why we opt for an RGB-D pipeline.

### B. Category-level Priors for Object Mapping

NodeSLAM [11] employs variational autoencoders as category-level shape priors to predict occupancy grids, jointly estimate the shape and pose of objects. However, it struggles with reconstructing complex shapes. Frodo [12] represents shape priors using encoder-decoder networks to model objects as point clouds and SDFs. Unlike our sequential method, it operates in batch mode.

Several works [13]–[15] adopt the DeepSDF architecture [19] as a shape prior, learning latent embeddings as shape priors, and training decoder networks to predict object SDFs. These approaches require long training times of several seconds per object on a low-end GPU while having limited capability to reconstruct fine geometry. Moreover, these methods also jointly estimating the object pose with shape reconstruction. While this can refine object localization, it increases training time and makes reconstruction accuracy dependent on state estimation quality. In contrast, our approach follows [11] by estimating the object pose, before initializing reconstruction. However, unlike the CNNs used by [11], we rely solely on our priors to estimate and refine the pose.

## III. Methodology

Our proposed methodology is composed of two key modules as can be seen in Figure 2. The offline prior trainer (Section III-A) learns a prior only once per category, after which it can be used across all subsequent runs of the online object mapping (Section III-B). Each mapped object is represented using a compact NeRF model based on the *tiny-cuda-nn* framework [20], with architecture $\alpha$ and initial parameters $\theta$.

### A. Offline Prior Training

We first define a set of categories of interest $\mathcal{I}$. Then, for each object category $\mathbf{k} \in \mathcal{I}$, a canonical object frame is defined, and a prior is trained to optimize $\hat{\alpha}_{\mathbf{k}}$ and $\hat{\theta}_{\mathbf{k}}$ while generating a mesh $\overline{\mathcal{M}}_{\mathbf{k}}$ and a prior density grid $\overline{\mathcal{F}}_{\mathbf{k}}$ in the Normalized Object Coordinate Space (NOCS) [21], which is bounded within a unit cube. Priors are trained via meta-learning across several object reconstruction *tasks*.

**Task Generation** Each task $\mathcal{T}$ consists of the data required to reconstruct a single object provided its pose and size. This includes a random number of RGB-D frames, instance masks, and camera poses (see Figure 2 for example), generated using publicly available 3D models primarily from the Shapenet [22] and Sapiens [23] datasets. 3D models are divided into training and test set, with $\mathcal{T}^{train}$ used to optimize $\theta$ and $\mathcal{T}^{test}$ used to optimize $\alpha$.

**Meta-learning** We use the Reptile [24] meta-learning algorithm to learn optimized initial parameters $\hat{\theta}$ such that they can be quickly adapted to unseen object instances within the same category during online mapping. Formally, given a distribution of tasks $\mathcal{T}^{train}_{\mathbf{k}}$ for an object category $\mathbf{k} \in \mathcal{I}$, each task $\mathcal{T}^{train}_{\mathbf{k},\mathbf{i}}$, for the meta-learning step $\mathbf{i}$ of $\mathbf{N}$, corresponds to learning optimal parameters $\theta_{\mathbf{k},\mathbf{i}}$ to reconstruct the associated object instance, while maintaining a fixed architecture, by performing gradient descent on the task-specific loss:

$$\theta^{(t+1)}_{\mathbf{k},\mathbf{i}} = \theta^{(t)}_{\mathbf{k},\mathbf{i}} - \eta \nabla_{\theta} \mathcal{L}_{\mathcal{T}}(\theta^{(t)}_{\mathbf{k},\mathbf{i}}), \tag{1}$$

where $\eta$ is the task learning rate. The loss function $\mathcal{L}_{\mathcal{T}}$ is based on differentiable volume rendering described later

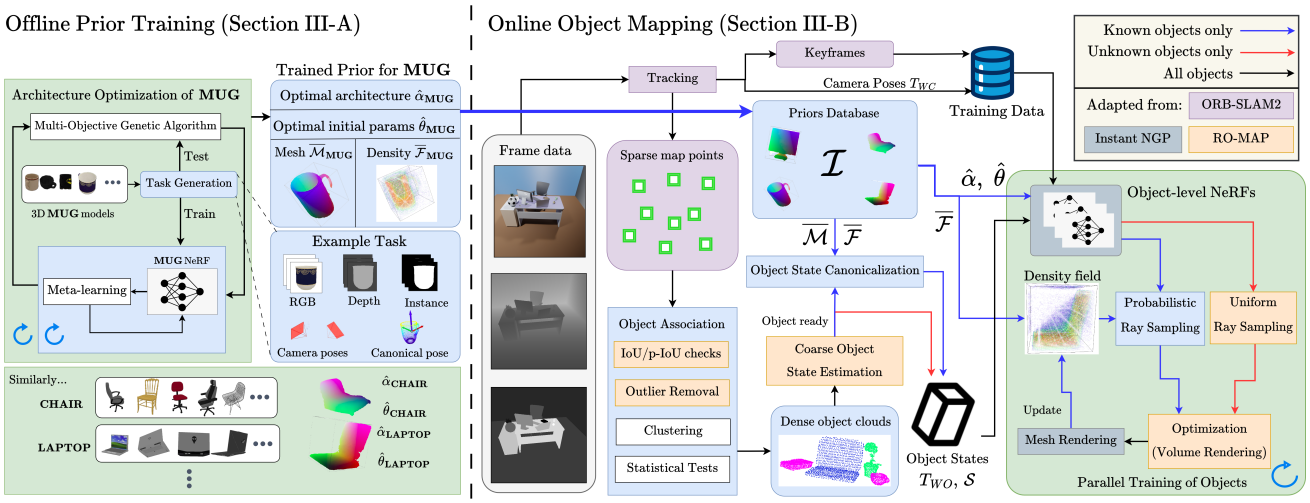


Fig. 2: Overview of the two key components of PRENOM.

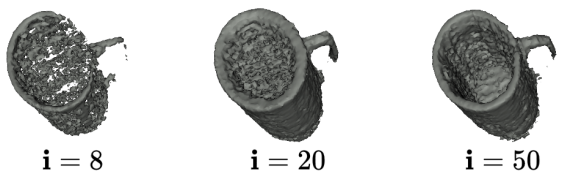


Fig. 3: Evolution of the prior normalized mesh $\overline{\mathcal{M}}_{\text{MUG}}$ during meta-learning (uniform color for illustration). The process first captures a rough shape, then refines the handle and brim, and finally learns the distinctive cavity of mugs.

in Section III-B. After a set number of inner iterations $q$, the updated parameters $\theta_{\mathbf{i}}^{(q)}$ are used to update the meta-parameters via a simple interpolation:

$$\theta_{\mathbf{k},\mathbf{i}+1} = \theta_{\mathbf{k},\mathbf{i}} + \beta(\theta_{\mathbf{k},\mathbf{i}}^{(q)} - \theta_{\mathbf{k},\mathbf{i}}), \tag{2}$$

where $\beta$ is the meta-learning step size. By meta-learning on several reconstruction tasks, the prior learns common geometric features within a category, as seen in Figure 3.

**Architecture Optimization** To choose $\hat{\alpha}$, we perform a multi-objective optimization, using a mixed-variable genetic algorithm [25] with the objectives of minimizing the training time and the object reconstruction error (measured using Chamfer distance). At each optimization iteration, meta-learning is used to obtain the optimized parameters $\theta_{\mathbf{k},\mathbf{N}}$, as described in Equations (1) and (2). These learned parameters are then evaluated on $\mathcal{T}_{\mathbf{k}}^{test}$ to guide the genetic algorithm to generate an improved architecture for the next iteration. Starting with an initial population of architectures with size $\mathbf{P}$, this iterative process is repeated for $\mathbf{M}$ generations, producing a set of Pareto-optimal solutions. From these solutions, the final architecture $\hat{\alpha}_{\mathbf{k}}$ is chosen as the knee point of the convex decreasing curve that best fits the solution set (see Figure 4). Consequently, $\hat{\theta}_{\mathbf{k}}$ are the corresponding meta-learned parameters for the selected architecture.

**Normalized mesh generation** $\overline{\mathcal{M}}_{\mathbf{k}}$ is generated using marching cubes on the density output of the NeRF with

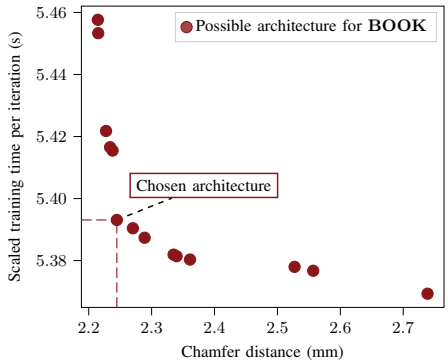


Fig. 4: Set of optimal architectures for category **BOOK**. The final architecture (dashed lines) is chosen as the knee point of the convex decreasing curve.

the optimized initial parameters $\hat{\theta}_{\mathbf{k}}$. The density output, computed over a cubic grid of resolution $R$, is stored as $\overline{\mathcal{F}}_{\mathbf{k}}$. During online object mapping, the mesh $\overline{\mathcal{M}}$ and density grid $\overline{\mathcal{F}}$ are used to canonicalize object poses.

### *B. Online Object Mapping*

Our mapping framework builds on top of RO-MAP [6], which extends ORB-SLAM2 [26], maintaining real-time capability. Camera poses are directly sourced from ORB-SLAM2's tracking module.

**Object Association** Our association strategy is adapted from [1], [2]. For consecutive frames, objects are considered the same if their bounding box Intersection over Union (IoU) exceeds a predefined threshold, while for non-consecutive frames, projected IoU (p-IoU) is used. If these criteria are met, we refine the association by using object point clouds from depth frames and instance masks. Each incoming object cloud is downsampled and clustered, with small clusters treated as outliers. The remaining clusters undergo statistical testing. Since object clouds do not follow a Gaussian distribution, we apply the Wilcoxon Rank-Sum Test [27] to assess

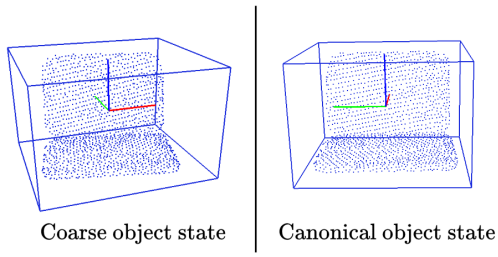


Fig. 5: Before and after pose canonicalization. Our approach aligns the object frame with the canonical frame established during offline prior training ($x$-axis, in red, points into the laptop screen), which also refines the fit of the bounding box.

distribution similarity. Meanwhile, object cloud centroids are assumed to follow a Gaussian distribution, verified using a one-sample t-test. If both tests pass, the incoming object cloud is merged with the existing one.

**Coarse Object State Estimation** The 9-DoF state comprises a 6-DoF pose and a 3-DoF size. Consistent with related works [1], [6], we assume that objects are parallel to the ground, reducing the pose estimation problem to position and yaw estimation. The optimal yaw is determined by aligning line features extracted from images with the edges of the bounding box. Once yaw is estimated, the object size is computed from the lengths of its spatial extents, while its position is defined as the center of these extents.

**Object State Canonicalization** An object is considered ready for reconstruction after it has been observed from a predefined sufficiently wide range of viewpoints. If the object has a prior (semantic class $\mathbf{k} \in \mathcal{I}$), its pose is transformed into the canonical frame relative to which the prior was learned. For this, we sample yaw angles between 0 and $2\pi$, and for each sample, apply the corresponding yaw rotation to the object point cloud $P$ to obtain $P_\gamma$. The object's size $\mathcal{S}_\gamma$ is estimated as the spatial extents of $P_\gamma$ and is used to normalize the point cloud, producing $\overline{P}_\gamma = P_\gamma/\mathcal{S}_\gamma + 1/2$ leading to $\overline{P}_\gamma \in [0, 1]$. The optimal yaw angle $\hat{\gamma}$ is then selected by maximizing the accumulated density of the normalized points in the prior density grid:

$$\hat{\gamma} = \underset{\gamma}{\operatorname{argmax}} \sum_{j=1}^{|P|} \overline{\mathcal{F}}_{\mathbf{k}}(\overline{P}_{\gamma,j}) \tag{3}$$

where $\overline{\mathcal{F}}_{\mathbf{k}}(x)$ represents the density of a normalized point $x$, computed via trilinear interpolation over the fixed-resolution $\overline{\mathcal{F}}_{\mathbf{k}}$. Here, $\overline{P}_{\gamma,j}$ denotes the $j^{\text{th}}$ point of the normalized object cloud $\overline{P}_\gamma$, which contains $|P|$ points. Note that this optimization is unnecessary for objects with rotational symmetry. Moreover, as a final step, we transform and scale $\overline{\mathcal{M}}_{\mathbf{k}}$ using $\hat{\gamma}$ and $\mathcal{S}\hat{\gamma}$ to obtain a reference mesh, which is then aligned using ICP.

This process not only ensures that the prior is loaded correctly but also refines the object state, resulting in a compact and well-aligned bounding box, as shown in Figure 5.

**Object-level NeRFs** Once the object state is estimated, it is implicitly reconstructed using a lightweight NeRF model, which incorporates a multi-resolution hash encoding [28] alongside a shallow MLP. Supervision is achieved through differentiable volume rendering, which samples and accumulates color and opacity along camera rays to reconstruct the object. For a known object $\mathbf{k} \in \mathcal{I}$, the optimized initial parameters $\hat{\theta}_{\mathbf{k}}$ are loaded into a NeRF model with architecture $\hat{\alpha}_{\mathbf{k}}$. Otherwise, a default architecture (adapted from [6]) and random parameter initialization are used. Refer to Figure 2 for color-coded flow.

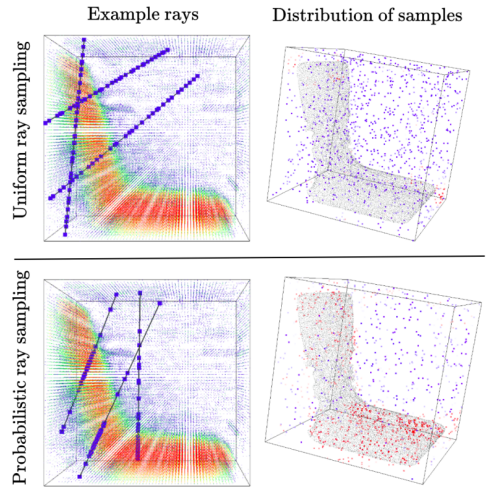


Fig. 6: Comparison of ray sampling strategies: Uniform vs. Probabilistic (Ours). *Left*: Three random rays with sampled points overlaid on $\overline{\mathcal{F}}_{\mathbf{LAPTOP}}$. *Right*: Distribution of sampled points with red denoting high density. Probabilistic sampling focuses more points where the mesh is expected.

**Training Data** RGB-D images, instance masks, and camera poses at every keyframe are used to train the NeRFs.

**Probabilistic Ray Sampling** We employ a ray sampling strategy based on a Cumulative Distribution Function (CDF) derived from $\overline{\mathcal{F}}$, which steers sampling towards regions with high occupancy probability. All camera rays are first transformed into the object frame using the estimated object pose, and $N_c$ samples are drawn at equal distance intervals along the ray, bounded within the object's bounding box. These serve as the discrete intervals to estimate the occupancy probability given by,

$$\rho_{\text{occ}}(\mathbf{x}_i) = 1 - \exp(-\sigma_i \delta_i) \tag{4}$$

where $\delta_i = \mathbf{d}_{i+1} - \mathbf{d}_i$, is the distance between adjacent sampled points $\mathbf{x}_{i+1}$ and $\mathbf{x}_i$. For ray sampling, we set $\sigma_i = \overline{\mathcal{F}}(\mathbf{x}_i)$ to both provide an initial estimate and avoid costly inference. The termination probability then becomes,

$$\rho_{\text{term}}(\mathbf{x}_i) = \rho_{\text{occ}}(\mathbf{x}_i) \prod_{j=1}^{i-1} (1 - \rho_{\text{occ}}(\mathbf{x}_j)) \tag{5}$$

using which a discrete CDF is constructed as follows:

$$\text{CDF}(i) = \frac{\sum_{j=1}^{i} \rho_{\text{term}}(\mathbf{x}_j)}{\sum_{j=1}^{N_c} \rho_{\text{term}}(\mathbf{x}_j)} \tag{6}$$

This monotonically increasing function represents the probability that the ray terminates at or before each sampled point. Finally, we use inverse transform sampling on the discrete CDF distribution to generate the $N_r$ sampled points $\mathbf{p}_k$ for ray $\mathbf{r}_k$. Doing so ensures that more samples are concentrated in regions where the mesh is expected, as can be seen in Figure 6. For rays that *escape* ($\sum_{i=1}^{N} \rho_{\text{term}}(\mathbf{x}_i) < \epsilon$, where $\epsilon$ is a small number), we revert to a uniform sampling strategy, maintaining coverage in low-density regions. Moreover, as training progresses, $\overline{\mathcal{F}}$ is updated every $m$ iterations through the mesh rendering process. This ensures that sampling progressively converges on the actual object, even if the initial object pose detection was imprecise.

**Volume Rendering** An overview of the differentiable volume rendering [5], [6], [17] is provided for completeness. For each object-centric ray $\mathbf{r}$, the positions of the sampled points $\mathbf{p}$ are normalized and encoded, then provided as input to an MLP to obtain the density $\sigma$ and the RGB color $\mathbf{c}$ for all points. The predicted color and depth of the ray are then accumulated using Equation (5) as follows:

$$\hat{\mathbf{C}}(\mathbf{r}) = \sum_{j=1}^{N_r} \rho_{\text{term}}(\mathbf{p}_j)\mathbf{c}_j, \quad \hat{\mathbf{D}}(\mathbf{r}) = \sum_{j=1}^{N_r} \rho_{\text{term}}(\mathbf{p}_j)\mathbf{d}_j \tag{7}$$

To compute the loss in the forward pass, rays $\mathcal{R}$ are categorized into those hitting the object and those that are part of the background as $\mathcal{R}_o$ and $\mathcal{R}_b$ respectively. Then the photometric loss $\mathcal{L}_{\mathbf{c}}$, depth loss $\mathcal{L}_{\mathbf{d}}$, and density loss $\mathcal{L}_\sigma$ are defined as:

$$\mathcal{L}_{\mathbf{c}} = \sum_{k \in \mathcal{R}_o} \left\| \hat{\mathbf{C}}(\mathbf{r}_k) - \mathbf{C}(\mathbf{r}_k) \right\|_2 + \sum_{k \in \mathcal{R}_b} \left\| \hat{\mathbf{C}}(\mathbf{r}_k) - \mathbf{C}_{\text{rand}} \right\|_2$$

$$\mathcal{L}_{\mathbf{d}} = \sum_{k \in \mathcal{R}_o} \left| \hat{\mathbf{D}}(\mathbf{r}_k) - \mathbf{D}(\mathbf{r}_k) \right|, \quad \mathcal{L}_\sigma = \sum_{k \in \mathcal{R}_b} \sum_{j=1}^{N_r} |\sigma_{k,j}| \tag{8}$$

where $\sigma_{k,j}$ is the predicted density for the $j^{\text{th}}$ sampled point of the $k^{\text{th}}$ ray. Here, $\mathbf{C}_{\text{rand}}$ is random color supervision that, together with $L_\sigma$, guide the network to learn zero densities for points sampled in the background of the object, as done in [6]. The final loss $L$ is a weighted sum,

$$\mathcal{L} = \mathcal{L}_{\mathbf{c}} + \lambda_{\mathbf{d}}\mathcal{L}_{\mathbf{d}} + \lambda_\sigma \mathcal{L}_\sigma \tag{9}$$

which is back-propogated to optimize the NeRF parameters using the Adam optimizer.

## IV. Evaluation

### A. Experimental Setup and Implementation Details

To train piors, we run the genetic algorithm, to optimize the architectures $\{\alpha_{\mathbf{k}} \,|\, \mathbf{k} \in \mathcal{I}\}$, with initial population size $\mathbf{P} = 25$ and for $\mathbf{M} = 100$ generations. Each $\alpha$ consists of all tunable hyperparameters of the NeRF representation (refer to [28] for a full list), the meta-learning ($\mathbf{N}$, $q$, $\eta$, and $\beta$), and the volume rendering loss ($\lambda_{\mathbf{d}}$ and $\lambda_\sigma$). For object reconstruction during online mapping, 4096 rays are generated at each iteration with $N_c = N_r = 32$ samples per ray. Moreover, we set $R = 64$, $\epsilon = 10^{-4}$ and $m = 50$. The probabilistic ray sampling is implemented using CUDA kernels ensuring minimal computational overhead. Moreover, a new thread is created for the training of each object-level NeRF.

Fig. 7: Example scenes from the generated synthetic dataset.

We run the experiments on a system equipped with an Intel Core i9-12900H and a low-power, low-memory (4GB) NVIDIA T600 Laptop GPU. With this setup, only a limited number of training iterations per object is possible to maintain sequential real-time processing of multi-object scenarios.

### B. Comparison with prior-free methods

We assess effectiveness by comparing against prior-free methods that also use NeRFs as the object representation.

**Baselines** We compare against three approaches: vMap [5], [17], and RO-MAP [6]. For vMAP, we provide its open-source implementation with poses from ORB-SLAM2 and disable the scene background reconstruction. For RO-MAP, we extend their monocular framework by incorporating depth data, which fixes scale ambiguity and enables dense depth supervision. We refer to this modified version as RO-MAP*. For [17], since it is not open-source, we adapt it from RO-MAP* by using the architecture and hyperparameters mentioned in their paper, naming it [17]*. For all comparisons, we adjust the training iterations to maintain sequential real-time multi-object mapping. However, since training is performed in batches, we allow the last batch in the queue to complete. Moreover, if a framework produces an empty object mesh, we increase the training iterations accordingly. Therefore for fairness, we also report the average processing time per sequence. Notably, all frameworks operate in a multi-object setting, training objects in parallel.

**Dataset** We evaluate on a synthetic dataset with available ground-truth meshes by extending the Cube Diorama dataset [17]. Originally comprising a single scene with four objects, we expand it to six scenes featuring 32 unique objects across nine diverse categories: *display, laptop, chair, mug, ball, keyboard, mouse, book*, and *plant* (see Figure 7 for examples). Notably, none of the object instances in this dataset were encountered by PRENOM during the offline prior training phase. Moreover, instance masks are provided offline to ensure consistent input data across all frameworks.

**Metrics** We use Chamfer Distance (CD) and Completion Ratio (CR) to quantitatively evaluate reconstruction quality. We do not evaluate object state recovery since prior-free methods cannot localize objects in a canonical frame.

| Method | S0 (4 objects) | | S1 (5 objects) | | S2 (5 objects) | | S3 (5 objects) | | S4 (6 objects) | | S5 (7 objects) | | **Average/obj.** | | Time |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CD↓ | CR↑ | CD↓ | CR↑ | CD↓ | CR↑ | CD↓ | CR↑ | CD↓ | CR↑ | CD↓ | CR↑ | CD↓ | CR↑ | (s)↓ |
| vMap | 0.538 | 0.396 | 0.718 | 0.283 | 0.985 | 0.263 | 0.615 | 0.481 | 1.404 | 0.233 | 1.122 | 0.337 | 0.870 | 0.334 | 19.9 |
| [17]* | 0.724 | 0.518 | 0.920 | 0.405 | 1.434 | 0.308 | 1.062 | 0.387 | 1.944 | 0.197 | 1.933 | 0.260 | 1.282 | 0.358 | 24.0 |
| RO-MAP* | 0.434 | **0.705** | 0.657 | 0.488 | 0.917 | **0.378** | 0.850 | 0.656 | 1.523 | 0.255 | 1.167 | **0.482** | 0.886 | 0.504 | 19.7 |
| *Ours* | **0.375** | 0.669 | **0.541** | **0.527** | **0.811** | 0.351 | **0.608** | **0.724** | **1.127** | **0.279** | **0.785** | 0.477 | **0.688** | **0.512** | 17.9 |

TABLE I: Evaluation results for six sequences with increasing number of objects. CD represents the Chamfer Distance (in cm) and CR represents the Completion Ratio ($< 0.4$ cm). All experiments were repeated thrice and the mean reported.

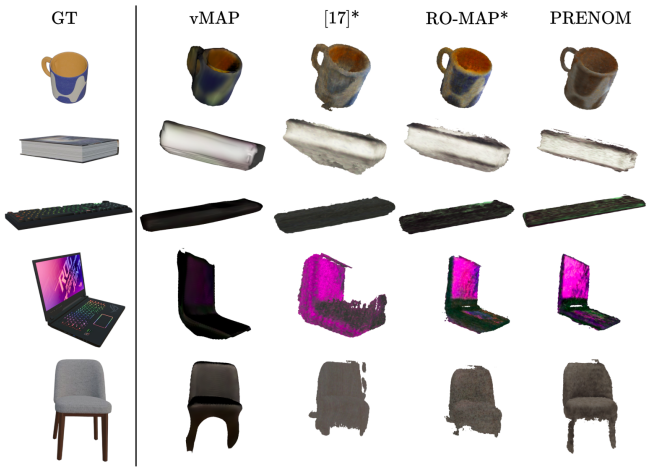


Fig. 8: PRENOM generates more refined and detailed geometry, such as the brim and cavity of the mug, pages of the book, flat surfaces of the keyboard and laptop, and thin legs of the chair. vMap produces over-smoothed geometry, while [17]* and RO-MAP* exhibit excess geometry.

**Results and discussion** Quantitative analysis in Table I shows that PRENOM consistently achieves a significantly lower CD across all scenes, outperforming the next-best method by 21% on average, while also achieving a slight improvement in average CR. Qualitative results on example objects in Figure 8 further illustrate PRENOM's ability to reconstruct sharper geometry, such as the brim and cavity of a mug, the flat surfaces of a book, keyboard, and laptop, and the thin legs of a chair. In contrast, excess geometry generated by RO-MAP* contributes to its improved completion ratio in some scenes, as evidenced by its significantly lower CD. It is also worth noting that PRENOM has been optimized to prioritize accurate shape reconstruction over photometric quality, which can result in minor losses in visual fidelity (e.g., the blue in the mug). This is a reasonable trade-off for object mapping scenarios, where novel-view rendering is typically unimportant. However, if needed, our prior generation pipeline can accommodate additional optimization objectives to improve photometric consistency.

### *C. Comparison with category-level shape prior approaches*

Most open-source approaches rely on DeepSDF as a shape prior which is computationally expensive, particularly in multi-object scenarios on a low-end GPU. Therefore, we run these evaluations in an offline, single-object setting.

**Baselines** We compare against three approaches: NodeSLAM [11], DSP-SLAM [13], and UncertainShape-Pose (USP) [15]. To ensure a fair object-mapping-only comparison, we adapt all approaches from the open-source implementation of USP. NodeSLAM does not originally use DeepSDF as a prior but since is not open-source, we use the variant of USP with the uncertainty component and 3D loss disabled, incorporating the sampling-based rendering from NodeSLAM while retaining the energy score from USP, denoted as NodeSLAM*. For DSP-SLAM, we use the variant of USP with the uncertainty component disabled and deterministic 3D and 2D loss functions, denoted as DSP-SLAM*. For the original USP, we retain the uncertainty component as it is integrated with the optimization process, despite the increased computational cost. Additionally, we also evaluate against a prior-free variant of PRENOM. This variant, while using the same initial object state and architecture as PRENOM, reconstructs an object-level NeRF from scratch without using meta-learned parameters.

For efficient execution of all frameworks, we analyze the loss curves of each method to determine the average number of iterations required for convergence. This iteration count is then used as the stopping criterion to ensure near-optimal results while maintaining computational efficiency. Therefore for reference, we also record the average runtime per object provided for each framework, though it does not necessarily represent the minimum time required.

**Dataset** We evaluate on the real-world ScanNet [29] dataset, focusing on the *chair* category. The dataset provides noisy depth and instance masks, while reference meshes are sourced from Scan2CAD [30] annotations. Ground-truth camera poses are used for all methods. We select 10 scenes containing 43 chair instances to assess both reconstruction quality and object state recovery. For all methods, we use the same 25 views, with identical segmentation masks and initial estimates of the object state.

**Metrics** To evaluate reconstruction quality, we use CD and CR as before. Moreover, we also incorporate Earth Mover's Distance (EMD) to provide a robust assessment of global shape, since NeRF-based approaches tend to produce sharper local features than DeepSDF-based approaches, and CD alone can be biased toward these details.

For object state estimation, we compute translation, rotation, and size errors. Morover, similar to PRENOM, the

| Method | CD↓ (cm) | EMD↓ (cm) | CR↑ (<1 cm) | Size↓ (MB) | Time↓ (s) |
|---|---|---|---|---|---|
| NodeSLAM* | 6.927 | 8.995 | 0.157 | 25.6 | 45.3 |
| DSP-SLAM* | 6.136 | 9.050 | 0.179 | 25.6 | 46.3 |
| USP [15] | 6.301 | 8.859 | 0.189 | 25.6 | 109.9 |
| *Ours* (w/o prior) | 6.273 | 10.024 | 0.254 | - | 9.4 |
| *Ours* | **5.189** | **8.017** | **0.294** | 17.9 | 9.1 |

TABLE II: Quantitative comparison of object reconstruction performance of shape prior approaches. Size represents the disk space required to store the prior, including $\overline{\mathcal{F}}_{\mathbf{CHAIR}}$ and $\overline{\mathcal{M}}_{\mathbf{CHAIR}}$ for PRENOM. Time indicates the reconstruction time provided per object (not minimum).

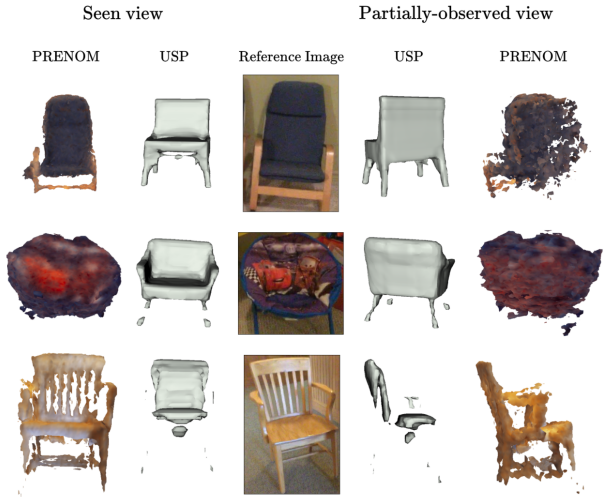


Fig. 9: PRENOM generates sharper geometry than the longest trained DeepSDF-based method (USP) for in-view regions but struggles with out-of-view features as scene information eventually dominates the weak prior. Both methods struggle with legs because of very noisy instance segmentation. Reference image is subjectively the clearest one of 25 input views.

baseline methods also assume that chairs lie parallel to the ground to compute the initial pose. However, since these methods update the pose during reconstruction, we align their final poses' vertical rotation axes with the ground truth before evaluation to ensure fairness. In contrast, PRENOM only estimates the pose using $\overline{\mathcal{F}}_{\mathbf{CHAIR}}$ and $\overline{\mathcal{M}}_{\mathbf{CHAIR}}$ (Equation (3) for reference) once before reconstructing, so it's vertical rotation axis is already aligned.

**Results and discussion** PRENOM achieves an improvement in reconstruction quality of approximately 13% averaged across all metrics compared to the second-best approach, as shown in Table II. Notably, it outperforms other methods in reconstruction quality while only being trained for 5× less time and with a 30% smaller prior size. Additionally, comparisons with our prior-free version re-emphasize that our prior-based approach not only enables canonical object state estimation but also significantly enhance reconstruction quality.

| Method | Trans. (cm)↓ | Rot. (°)↓ | Size (%)↓ |
|---|---|---|---|
| NodeSLAM* | **14.38** | 52.58 | 36.27 |
| DSP-SLAM* | 14.74 | 52.48 | 24.48 |
| USP [15] | 14.95 | 50.88 | **23.41** |
| *Ours* | 18.60 | 19.46 | 31.60 |

TABLE III: Quantitative comparison with other shape prior approaches for object pose and size estimation.

| Mode | AO | ML | PS | CD↓ (cm) | CR↑ (<0.4 cm) | Size↓ (MB) |
|---|---|---|---|---|---|---|
| RGB | ✗ | ✗ | ✗ | 0.495 | 0.545 | 17.2 |
| | ✓ | ✗ | ✗ | 0.422 (↓ 15%) | 0.600 (↑ 10%) | 5.5 |
| | ✓ | ✓ | ✗ | 0.369 (↓ 25%) | 0.660 (↑ 21%) | 5.5 |
| | ✓ | ✓ | ✓ | **0.330 (↓ 33%)** | **0.732 (↑ 34%)** | 5.5 |
| RGB-D | ✗ | ✗ | ✗ | 0.455 | 0.569 | 17.2 |
| | ✓ | ✗ | ✗ | 0.368 (↓ 19%) | 0.630 (↑ 11%) | 5.5 |
| | ✓ | ✓ | ✗ | 0.349 (↓ 23%) | 0.643 (↑ 13%) | 5.5 |
| | ✓ | ✓ | ✓ | **0.327 (↓ 28%)** | **0.711 (↑ 25%)** | 5.5 |

TABLE IV: Ablation study with two seconds of dedicated training time per object on the *Cube Diorama* dataset. AO denotes Architecture Optimization, ML denotes Meta-learning, and PS denotes Probabilistic ray Sampling. Size represents the disk space serialized NeRF parameters occupy.

Furthermore, example qualitative results in Figure 9 demonstrate PRENOM's ability to reconstruct sharp geometrical features within the observed regions. However, we do not impose geometric constraints with the initial prior, and during training, scene information gradually dominates, leading to noisier reconstructions in partially-observed/unobserved regions and regions with occlusions. In contrast, DeepSDF-based methods struggle to reconstruct complex and thin geometries typical of chairs but can predict the unseen regions in a way that remains structurally similar to a chair, even if it is not consistent with the scene.

Moreover, despite not jointly optimizing object state with reconstruction, our method achieves comparable object state recovery results as can be seen in Table III. We do not consider the improvement in rotation accuracy to be significant, as PRENOM does not adjust pose during reconstruction and thus benefits from the initial assumption that objects are parallel to the ground. While this initial assumption is used by all methods, the baseline approaches are more susceptible to noise as they alter the pose during the optimization process, which can negatively impact the rotation accuracy.

### *D. Ablation Study*

We ablate the impact of the three main components of PRENOM by evaluating the upper bound of reconstruction quality using ground-truth object states and camera poses on the *Cube Diorama* dataset. As shown in Table IV, our per-category architecture optimization achieves a 68% reduction in model size, averaged over $\mathcal{I}$, compared to the default tuned category-agnostic architecture adapted from [6], while simultaneously improving reconstruction quality. Additionally,

meta-learning the parameters and incorporating probabilistic sampling independently contribute further improvements.

We also conduct this ablation with monocular input to highlight the potential benefits of incorporating our priors into monocular reconstruction. Given object states, our method enables the transfer of depth-based knowledge acquired during offline training to the online monocular reconstruction process, resulting in even greater improvements than those observed with RGB-D. However, accurately estimating object states in monocular setups remains challenging due to scale ambiguity and noisy object association.

## V. Conclusion

We presented PRENOM, an efficient prior-based neural object mapper that leverages category-level priors to improve 3D object reconstruction and enables canonical pose estimation. By integrating meta-learned priors and probabilistic ray sampling, optimized per-category using a multi-objective genetic algorithm, PRENOM enhances reconstruction accuracy and efficiency, making it well-suited for multi-object mapping in resource-constrained setups. Experimental results demonstrate its effectiveness, achieving a 21% lower Chamfer distance compared to prior-free obejct-level NeRF-based approaches. Further comparisons with DeepSDF-based approaches show a 13% improvement in reconstruction metrics, with comparable pose and size estimation capabilities, highlighting the efficiency of our approach, as it was trained for $5\times$ less time and with a 30% smaller prior size. However, despite its advantages, PRENOM is inherently limited by its NeRF-based representation, which struggles with out-of-view feature reconstruction. Although our priors help mitigate these issues, the absence of hard constraints to enforce consistency eventually leads to noisier reconstructions, especially if trained for extended periods. Additionally, object state is not refined during training, resulting in slightly worse state estimation than some existing shape prior methods. Future work will focus in these directions.

## References


[1] Y. Wu, Y. Zhang, D. Zhu, Z. Deng, W. Sun, X. Chen, and J. Zhang, "An object slam framework for association, mapping, and high-level tasks," IEEE Transactions on Robotics, vol. 39, no. 4, pp. 2912–2932, 2023.

[2] X. Han and L. Yang, "Sq-slam: Monocular semantic slam based on superquadric object representation," Journal of Intelligent & Robotic Systems, vol. 109, no. 2, p. 29, 2023.

[3] Q. Wu, X. Liu, Y. Chen, K. Li, C. Zheng, J. Cai, and J. Zheng, "Object-compositional neural implicit surfaces," in European Conference on Computer Vision. Springer, 2022, pp. 197–213.

[4] Q. Wu, K. Wang, K. Li, J. Zheng, and J. Cai, "Objectsdf++: Improved object-compositional neural implicit surfaces," in The IEEE/CVF International Conference on Computer Vision, 2023, pp. 21 764–21 774.

[5] X. Kong, S. Liu, M. Taher, and A. J. Davison, "vmap: Vectorised object mapping for neural field slam," in The IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 952–961.

[6] X. Han, H. Liu, Y. Ding, and L. Yang, "Ro-map: Real-time multi-object mapping with neural radiance fields," IEEE Robotics and Automation Letters, 2023.

[7] C. Yang, S. Li, J. Fang, R. Liang, L. Xie, X. Zhang, W. Shen, and Q. Tian, "Gaussianobject: High-quality 3d object reconstruction from four views with gaussian splatting," ACM Transactions on Graphics, 2024.

[8] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. Kelly, and A. J. Davison, "Slam++: Simultaneous localisation and mapping at the level of objects," in The IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2013, pp. 1352–1359.

[9] N. Merrill, Y. Guo, X. Zuo, X. Huang, S. Leutenegger, X. Peng, L. Ren, and G. Huang, "Symmetry and uncertainty-aware object slam for 6dof object pose estimation," in The IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 14 901–14 910.

[10] Z. Liao, B. Xu, and S. L. Waslander, "Toward general object-level mapping from sparse views with 3d diffusion priors," in 8th Annual Conference on Robot Learning, 2024.

[11] E. Sucar, K. Wada, and A. Davison, "Nodeslam: Neural object descriptors for multi-view shape reconstruction," in The International Conference on 3D Vision (3DV). IEEE, 2020, pp. 949–958.

[12] M. Rünz, K. Li, M. Tang, L. Ma, C. Kong, T. Schmidt, I. Reid, L. Agapito, J. Straub, S. Lovegrove, and R. Newcombe, "Frodo: From detections to 3d objects," in The IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 14 708–14 717.

[13] J. Wang, M. Rünz, and L. Agapito, "Dsp-slam: Object oriented slam with deep shape priors," in The International Conference on 3D Vision (3DV). IEEE, 2021, pp. 1362–1371.

[14] Z.-X. Zou, S.-S. Huang, T.-J. Mu, and Y.-P. Wang, "Objectfusion: Accurate object-level slam with neural object priors," Graphical Models, vol. 123, p. 101165, 2022.

[15] Z. Liao, J. Yang, J. Qian, A. P. Schoellig, and S. L. Waslander, "Uncertainty-aware 3d object-level mapping with deep shape priors," in The IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2024, pp. 4082–4089.

[16] M. Tancik, B. Mildenhall, T. Wang, D. Schmidt, P. P. Srinivasan, J. T. Barron, and R. Ng, "Learned initializations for optimizing coordinate-based neural representations," in The IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 2846–2855.

[17] J. Abou-Chakra, F. Dayoub, and N. Sünderhauf, "Implicit object mapping with noisy data," arXiv preprint arXiv:2204.10516, 2022.

[18] B. Yang, Y. Zhang, Y. Xu, Y. Li, H. Zhou, H. Bao, G. Zhang, and Z. Cui, "Learning object-compositional neural radiance field for editable scene rendering," in The IEEE/CVF International Conference on Computer Vision, 2021, pp. 13 779–13 788.

[19] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.

[20] T. Müller, "tiny-cuda-nn," 4 2021. [Online]. Available: https://github.com/NVlabs/tiny-cuda-nn

[21] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, and L. J. Guibas, "Normalized object coordinate space for category-level 6d object pose and size estimation," in The IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 2642–2651.

[22] A. X. Chang et al., "Shapenet: An information-rich 3d model repository," arXiv preprint arXiv:1512.03012, 2015.

[23] F. Xiang et al., "SAPIEN: A simulated part-based interactive environment," in The IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 2020.

[24] A. Nichol and J. Schulman, "Reptile: a scalable metalearning algorithm," arXiv preprint arXiv:1803.02999, vol. 2, no. 3, p. 4, 2018.

[25] J. Blank and K. Deb, "pymoo: Multi-objective optimization in python," IEEE Access, vol. 8, pp. 89 497–89 509, 2020.

[26] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," IEEE Transactions on Robotics, vol. 33, no. 5, pp. 1255–1262, 2017.

[27] F. Wilcoxon, "Individual comparisons by ranking methods," in Breakthroughs in statistics: Methodology and distribution. Springer, 1992, pp. 196–202.

[28] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," ACM Transactions on Graphics, vol. 41, no. 4, pp. 102:1–102:15, Jul. 2022. [Online]. Available: https://doi.org/10.1145/3528223.3530127

[29] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in The IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2017, pp. 5828–5839.

[30] A. Avetisyan, M. Dahnert, A. Dai, M. Savva, A. X. Chang, and M. Niessner, "Scan2cad: Learning cad model alignment in rgb-d scans," in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.