Scene Coordinate Reconstruction Priors

Wenjing Bian[2,*] Axel Barroso-Laguna[1] Tommaso Cavallari[1]
Victor Adrian Prisacariu[1,2] Eric Brachmann[1]
[1]Niantic Spatial [2]University of Oxford

https://nianticspatial.github.io/scr-priors/

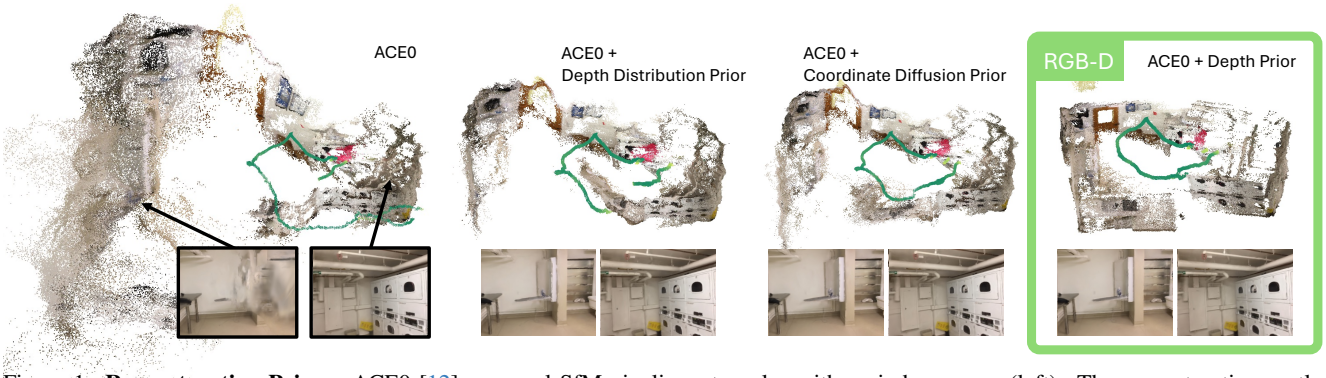

Figure 1. **Reconstruction Priors.** ACE0 [12], a neural SfM pipeline, struggles with an indoor scene (left). The reconstruction partly degenerates, as is evident by the dispersed point cloud, camera poses (green) penetrating a wall, and artifacts in one of the views synthesized with Splatfacto [90] (bottom). We incorporate various priors into ACE0 which lead to a more consistent scene layout, better pose estimates, and fewer artifacts in synthesized views (center). We also show a version of ACE0 taking RGB-D rather than RGB images as input (right).

Abstract

Scene coordinate regression (SCR) models have proven to be powerful implicit scene representations for 3D vision, enabling visual relocalization and structure-from-motion. SCR models are trained specifically for one scene. If training images imply insufficient multi-view constraints SCR models degenerate. We present a probabilistic reinterpretation of training SCR models, which allows us to infuse high-level reconstruction priors. We investigate multiple such priors, ranging from simple priors over the distribution of reconstructed depth values to learned priors over plausible scene coordinate configurations. For the latter, we train a 3D point cloud diffusion model on a large corpus of indoor scans. Our priors push predicted 3D scene points towards plausible geometry at each training step to increase their likelihood. On three indoor datasets our priors help learning better scene representations, resulting in more coherent scene point clouds, higher registration rates and better camera poses, with a positive effect on down-stream tasks such as novel view synthesis and camera relocalization.

1. Introduction

With the recent advent of learning-based structure-from-motion (SfM) pipelines, there are three avenues to further push their capabilities: 1) priors, 2) priors, and 3) priors.

In this work, we consider a particular class of neural SfM models that hitherto only incorporated very weak priors: Scene Coordinate Regression (SCR) [82]. SCR models are implicit scene representations. They are trained on images of a particular scene with known ground truth camera poses [4, 7, 8, 11, 82]. Once trained, SCR models allow to estimate the camera poses of unseen query images relative to the training scene, *i.e.* they enable visual relocalization. Recently, ACE0 [12] showed self-supervised training of SCR, turning it into a fully differentiable, neural SfM approach.

SCR reinterprets classical 3D reconstruction using a machine learning methodology. However, they still follow classical 3D vision principles when learning the 3D geometry of a scene: where feature-based methods triangulate sparse key points explicitly, SCR methods rely on dense, implicit triangulation of image patches [4]. Whether one triangulates implicitly or explicitly, without sufficient multi-view constrains triangulation fails [34]. This happens *e.g.* for texture-poor areas, repetitive structures, reflections, *etc.* where multi-view observations cannot be resolved to the

* Work done during an internship at Niantic.

same 3D scene point. In Fig. 1, featureless walls and floor cause ACE0 to degenerate with an adverse effect on novel view synthesis on top of the ACE0 reconstruction.

Our aim is to overcome the ambiguity and potential incompleteness of scene-specific training data, by leveraging high-level reconstruction priors. It is easy to see that the degenerate scene representation of Fig. 1 (left) is unlikely to correspond to real scene geometry as the left half of the room is dispersed into space, and the estimated camera trajectory penetrates a wall. Following this intuition, we reformulate the training objective of SCR to easily incorporate reconstruction priors in a maximum likelihood framework. These priors can be hand-crafted [2, 63, 73, 100] or learned from data [99], and should lead to coherent and plausible reconstructions as the one in Fig. 1 (right).

For hand-crafted priors, we model plausible distributions of depth values and force reconstructed scene coordinates to follow those distributions. For a learned prior, we pretrain a 3D diffusion model on point clouds of indoor scenes, to encode plausible scene geometries. Training generative models, like diffusion models, on 3D representations is a difficult endeavor. 3D datasets are orders of magnitude smaller than their 2D counterparts. The largest 3D datasets are comprised of simplistic 3D assets [19, 25, 26] wheres datasets with entire 3D scenes are smaller still, containing a few hundred scenes at most [1, 23, 48, 103]. We also still lack efficient but expressive architectures to generate entire 3D scenes with high fidelity. Both data and architecture limitations contribute to the fact that, so far, 3D point cloud diffusion was only shown for individual, isolated objects [55–57, 105]. However, we show that for mere use as a *prior*, a lean 3D point cloud diffusion model trained on relatively little data (around 700 indoor scenes) suffices to provide useful guidance when reconstructing indoor scenes.

We summarize our **contributions**:

- We reformulate SCR training as maximum likelihood learning to enable incorporation of reconstruction priors.
- We propose hand-crafted priors that push depth values of reconstructed scene coordinates to follow a reasonable distribution, and a learned prior in form of a 3D point cloud diffusion model that pushes reconstructed scene coordinates towards representing plausible scene layouts.
- We plug our priors into state-of-the-art SCR frameworks ACE [11], GLACE [93] and ACE0 [12]. They lead to learning better scene representations signified by more coherent and accurate point clouds, higher registration rates in SfM, and better pose estimates. Our approach does not significantly increase the training time of SCR models and it does not affect efficiency at query time.
- As a byproduct, we show how measured depth maps can be incorporated as a prior, leading to effective RGB-D versions of ACE and ACE0.

2. Related Work

Visual Reconstruction and Relocalization Structure-from-motion refers to the problem of estimating scene geometry and camera poses from a set of images [14, 65, 80, 83, 97]. Visual relocalization is a related task where the camera pose of a query image should be estimated relative to mapping images with known camera poses [39, 74, 76, 78]. Both tasks have been addressed with classical feature-matching where scenes are represented as sparse 3D point clouds. The current generation of learning-based feature matchers [30, 49, 75, 89] is very reliable [40], with methods like MicKey [3] and MASt3R [46, 96] demonstrating successful feature matching even for opposing shots. However, despite the robustness of feature matchers, classical triangulation relies on the availability of sufficient multi-view constraints to reconstruct a scene [34]. In texture-poor areas or with small camera baselines, point triangulation might still degenerate. Since feature-based pipelines consist of various stages with complex control flow [35, 80, 83, 94, 97], it is difficult to infuse high-level priors for regularization.

Feed-forward reconstruction methods, like DUSt3R [96] and MASt3R [46], do suffer less from potential degeneracies of point triangulation. Trained on large datasets, these methods incorporate strong reconstruction priors to enable reconstruction in sparse-view scenarios with little to no visual overlap. However, these methods are inherently binocular, and are difficult to scale to large image collections. In a spirit similar to DUSt3R/MASt3R, our work aims at enabling strong priors for SCR methods.

As another alternative to feature matching, pose regression methods aim to predict camera poses using a feed-forward network [13, 20, 41, 42, 81]. These networks represent the scene implicitly, and do not offer introspection on whether their scene representations are more or less prone to degeneracies. In any case, pose regression methods either lack accuracy [79], or depend on massive amounts of scene-specific synthesized training data [21, 60].

Scene Coordinate Regression (SCR) SCR models fuse the key point extraction and matching stages of classical feature-based methods into a single regression step, performed by a scene-specific machine learning model. Originally proposed by Shotton *et al*. [82] for relocalization in small-scale indoor scenes with RGB-D images and random forests, the approach was later extended to on-the-fly adaptation [16–18], RGB inputs [4, 8], using neural networks [6, 7, 9] and to larger scenes [5, 47, 93, 95]. Recently, the ACE framework [11] has demonstrated training of extremely memory-efficient SCR models in a few minutes per scene. Subsequently, ACE0 [12] extended the applicability of ACE models from visual relocalization to structure-from-motion by demonstrating self-supervised training.

Most previous SCR methods rely entirely on scene-specific training, and make very little use of scene-agnostic data to distill reconstruction priors. ACE and ACE0 use a pre-trained feature encoder. However, as a low-level component, the feature encoder can do little to ensure coherence of the final scene representation. Marepo [22] pre-trains a scene-agnostic transformer to replace RANSAC [31] and PnP [33] in the ACE pipeline. In a similar spirit, SACReg [72] and SANet [101] pre-train scene-agnostic SCR predictors that learn to interpolate scene coordinate annotations of mapping images. Marepo, SACReg and SANet all concern test-time components of SCR, and assume that a valid scene representation has already been built. To the best of our knowledge, we are the first to regularize SCR training by high-level priors, derived from scene-agnostic data. Some of our priors, regularizing the depth distribution of scene coordinates, bear conceptional similarity with depth regularization terms proposed for novel view synthesis [2, 73]. We show how to apply such priors in the context of SCR, based on a probabilistic re-formulation of training.

Denoising Diffusion Models (DDMs) One of the priors we propose is a DDM on 3D point clouds to assess the likelihood of scene coordinate configurations. DDMs are generative models that incrementally transform random noise to a target distribution, *e.g.* transforming high-dimensional Gaussian noise to the distribution of natural images [37, 84, 85]. The transformation is implemented using a model that is trained to remove a small amount of noise at each timestep. Generation can be unconditional, or conditional, *e.g.* generating images of a pre-defined class [27, 36]. Adjacent to our work, scene coordinates have been used as a conditioning signal for image diffusion [62].

Due to the limited size and diversity of 3D datasets, diffusion for 3D generation is difficult. 2D diffusion models have been utilized to regularize or guide 3D generation [32, 67, 98]. For autonomous driving, large-scale scene generation has been demonstrated for 2.5D LiDAR range images [64, 70, 106], and coarse voxel grids [45, 51].

Diffusion models are related to score matching methods [86, 87] where models estimate the gradient of the log-likelihood of the target distribution [15, 37, 54]. This observation was used by DiffusioNeRF [99] to utilize a diffusion model to represent the prior probability over RGB-D image patches when training a neural radiance field [58]. We take inspiration from DiffusioNeRF's formulation but avoid the significant slow-down of rendering 2.5D image patches throughout training, and regularize directly in 3D.

DDMs for Point Clouds Luo and Hu [55] first demonstrated diffusion on 3D point clouds using a PointNet [68] architecture. Subsequent works [56, 59, 104, 105] proposed various improvements, including better architectures like PointNet++ [69] and point-voxel CNN [53]. These works model simple, isolated objects like *e.g.* those of ShapeNet [19]. Melas-Kyriazi *et al.* [57] show image-conditional diffusion of realistic, but still isolated, objects on Co3D [71]. To the best of our knowledge, generation of scene-level 3D point clouds using diffusion has not been demonstrated. We use a point-voxel CNN [53] to model the distribution of scenes in ScanNet [23]. While our generative model also cannot generate scenes with high fidelity, we show that it still serves as an efficient prior for regularizing SCR.

3. Method

SCR models [82] encode a scene into a scene-specific neural network f. The network f maps an image patch $\mathbf{p}_i$ centered around pixel i of image $\mathcal{I}$ to a 3D scene point $\mathbf{y}_i$,

$$\mathbf{y}_i = f(\mathbf{p}_i; \mathbf{w}), \tag{1}$$

where $\mathbf{w}$ denotes the parameters that encode the scene.

An explicit 3D point cloud of the scene can be extracted from f by running it over scene images and collecting the 3D points $\mathbf{y}_i$ [7]. However, the main application of SCR models is camera pose estimation, where f is applied to query images with unknown poses. Since the prediction $\mathbf{y}_i$ represents a 2D-3D correspondence from pixel i to scene space, the outputs of f can be used for camera pose estimation by feeding them into RANSAC [31] and PnP [33].

An SCR model f can be trained using RGB-D or RGB images [7]. Although we will present an RGB-D version of our approach in Sec. 3.4, we are mainly interested in the general case and will assume that only RGB images are available if not stated otherwise. Training also requires known camera poses for the training images. With multiple mapping images $\mathcal{I}_\text{M}$ and their known camera poses, f can be optimized using a reprojection loss L_reproj. However, when multi-view constraints from the images are insufficient or ambiguous, f may (partly) degenerate, and estimate scene points $\mathbf{y}_i$ that are noisy, distorted or plain outliers, and degrade downstream performance, see Fig. 1 (left).

To mitigate this, we complement L_reproj with a regularization term L_reg. L_reg should enforce prior geometric constraints, *e.g.* that scene coordinates follow a plausible depth distribution, or that all scene coordinates represent a coherent scene layout. To this end, we reformulate the SCR training objective as maximum likelihood learning, and set the regularization term to the negative log-likelihood of scene coordinates: $L_\text{reg} = -\log p(\mathbf{y})$. Since we apply the prior during training, it does not affect test time efficiency. We base our work on ACE [11], due to its good accuracy and attractive training time. Our priors can be readily applied to any derivative of ACE as long as it keeps the general training framework. We show results using ACE0 [12], a self-supervised version of ACE for SfM, and GLACE [93] for relocalization. See Fig. 2 for a system overview.

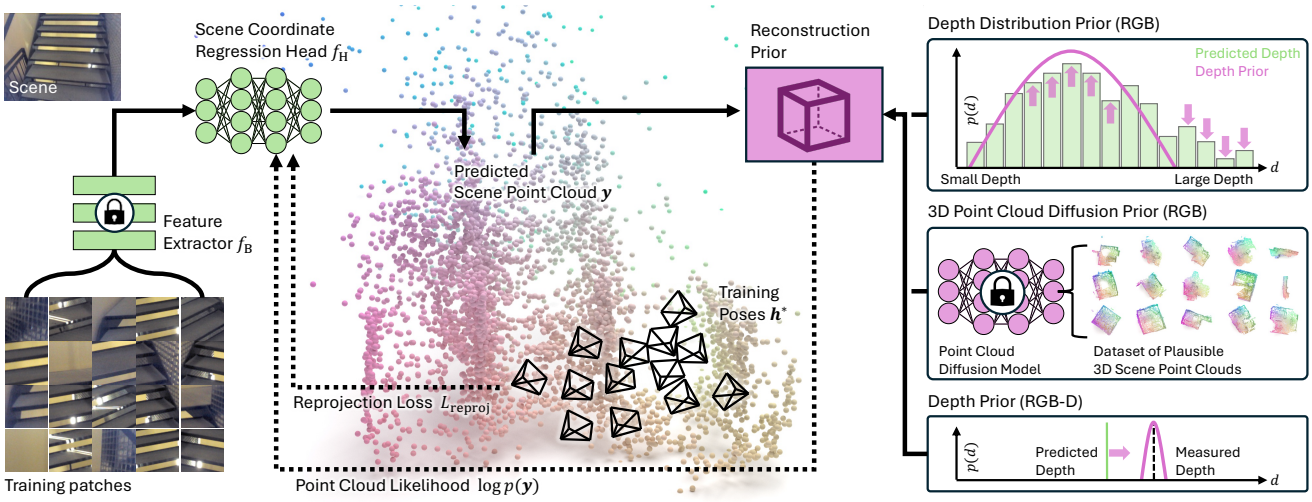

Figure 2. **System. Left:** We train a scene coordinate regression (SCR) model to represent a scene. Following ACE [11] and ACE0 [12], the SCR network predicts scene coordinates for a batch of random image patches in each training iteration. We supervise the SCR network with the reprojection error of scene coordinates w.r.t. ground truth camera poses. **Right:** Extending [11, 12], we introduce a reconstruction prior that outputs the gradient of the log-likelihood of the predicted point cloud. The prior guides the SCR model to learn a scene representation that is more likely to correspond to a real scene. We investigate various priors: A *depth distribution prior* encourages depth values of reconstructed scene coordinates to follow a plausible distribution. A *3D point cloud diffusion prior* is a network that was trained offline on a large corpus of scenes and encodes plausible scene layouts. We can also infuse a *depth prior* if inputs are RGB-D.

3.1. ACE Framework

ACE decomposes a SCR model f into a scene-agnostic feature extractor f_B and a scene-specific regression head f_H,

$$f(\mathbf{p}_i; \mathbf{w}) = f_\text{H}(\mathbf{f}_i; \mathbf{w}_\text{H}), \text{ with } \mathbf{f}_i = f_\text{B}(\mathbf{p}_i; \mathbf{w}_\text{B}). \quad (2)$$

The feature extractor f_B maps an image patch to a high-dimensional feature vector $\mathbf{f}_i$, and the regression head f_H maps the feature to a scene coordinate $\mathbf{y}_i$. Following ACE, we assume f_B was pre-trained and remains fixed. We focus on the mapping stage which optimizes f_H, see Fig. 2 (left).

ACE first passes all mapping images $\mathcal{I}_\text{M}$ to the feature extractor f_B to obtain a training buffer with features for a large number of patches randomly sampled from mapping images. When training f_H, to achieve faster and more stable convergence, the buffered features are shuffled at each epoch. In each training iteration, a batch of N features $\mathbf{f} = \{\mathbf{f}_i \mid i = 1, ..., N\}$ is sampled from the feature buffer to estimate the 3D scene points $\mathbf{y} = \{\mathbf{y}_i \mid i = 1, ..., N\}$ for the corresponding 2D pixels from $\mathcal{I}_\text{M}$.

The ACE training loss $L_\text{reproj}(\mathcal{I}_\text{M}, \mathbf{y}, \mathbf{h}^*)$ is a pixel-wise projection loss applied to the 3D scene predictions $\mathbf{y}$ in each training iteration. Here, $\mathbf{h}^* = \{\mathbf{h}^*_i \mid i = 1, ..., N\}$ denotes the ground truth poses for each image from which the pixels are sampled. To regularize training in the first few iterations, ACE applies an additional loss, $L_\text{init}(\mathcal{I}_\text{M}, \mathbf{y}, \mathbf{h}^*)$, that mainly ensures that predicted coordinates are in front of their associated camera. For further details, please refer to ACE [11].

3.2. Probabilistic Training Objective

We recast the usual loss minimization objective of SCR as the maximization of a scene coordinate's probability. The posterior probability for the scene points $\mathbf{y}$, given the mapping images $\mathcal{I}_\text{M}$ and poses $\mathbf{h}^*$, is proportional to the product of the likelihood $p(\mathbf{h}^*, \mathcal{I}_\text{M} \mid \mathbf{y})$ and the prior $p(\mathbf{y})$:

$$p(\mathbf{y} \mid \mathbf{h}^*, \mathcal{I}_\text{M}) = \frac{p(\mathbf{h}^*, \mathcal{I}_\text{M} \mid \mathbf{y}) p(\mathbf{y})}{p(\mathbf{h}^*, \mathcal{I}_\text{M})}. \quad (3)$$

Taking the negative logarithm of the posterior and omitting the constant $p(\mathbf{h}^*, \mathcal{I}_\text{M})$ yields:

$$-\log p(\mathbf{y} \mid \mathbf{h}^*, \mathcal{I}_\text{M}) \propto -\log p(\mathbf{h}^*, \mathcal{I}_\text{M} \mid \mathbf{y}) - \log p(\mathbf{y}). \quad (4)$$

In ACE mapping, minimizing the reprojection error can be interpreted as maximizing the log-likelihood of the mapping views and poses given the predicted scene points, *i.e.*

$$-\log p(\mathbf{h}^*, \mathcal{I}_\text{M} \mid \mathbf{y}) \coloneqq L_\text{reproj}. \quad (5)$$

Similarly, ACE's regularization term can be interpreted as a form of prior on scene coordinates, *i.e.* $-\log p(\mathbf{y}) \coloneqq L_\text{init}$. However, ACE only optimizes either L_reproj or L_init per scene coordinate where the latter serves as an initialization target for the former. In contrast, our perspective suggests that reprojection error and prior should be optimized jointly to maximize the likelihood of a scene coordinate. We propose a set of alternative priors as stronger regularization terms L_reg instead of L_init, to be jointly optimized with the reprojection error: $L_\text{reproj} + L_\text{reg} = L_\text{reproj} - \log p(\mathbf{y})$.

3.3. Depth Distribution Prior (RGB)

Firstly, we present a prior that builds on, and extends, the intuition of ACE's original initialization loss. That loss ensures that the depth d_i of a predicted scene coordinate $\mathbf{y}_i$ is within sensible bounds, namely $0 < d_i < d_{\text{max}}$, where d_{max} is a user-defined upper bound. Between those bounds, the initialization loss provides no signal. Instead ACE switches to optimizing the reprojection error only.

We propose to model a continuous distribution of plausible depth values, and to encourage scene coordinate to adhere to that distribution. First, we have to choose a distribution family. We need our prior to be defined over the whole real line to be able to even assess the likelihood of *negative* depth values which can occur during SCR training. We empirically found a Laplacian distribution, $\text{Lap}(d \mid \mu, b)$, to be a good choice. Having chosen the distribution family, we fit its hyper-parameters, mean μ and bandwidth b, to a held out set of scenes with measured (*i.e.* ground truth) depth values.

We can directly use the negative log-likelihood of individual depth values as a prior per pixel i, *i.e.*

$$\log p(\mathbf{y_i}) \coloneqq \lambda_{\text{reg}} \ \log \text{Lap}(d_i \mid \mu, b), \tag{6}$$

where λ_{reg} is a hyper-parameter to balance the prior with the reprojection loss. This prior corresponds to an L_1 loss on the predicted scene coordinate's depth, pulling it to the empirical mean.

While Eq. 6 increases the likelihood of *individual* scene coordinates, it does not guarantee the set of *all* scene coordinates to actually follow the target distribution, in particular to have the correct variance. Therefore, as an alternative to Eq. 6, we can utilize a *distribution loss* between the predicted depth values and the prior. We use the Wasserstein distance between the set of predicted depth values of a mini-batch, $\{d_i\}$, and the target Laplace distribution:

$$\log p(\mathbf{y_i}) \coloneqq \lambda_{\text{reg}} \ W\left[\{d_i\}, \ \text{Lap}(\cdot \mid \mu, b)\right]. \tag{7}$$

Here, we take advantage of the ACE framework which constructs mini-batches associated with random scene points throughout optimization. Therefore, we can assume that the distribution of predicted depth values of a mini-batch roughly follows the depth distribution of the entire scene.

3.4. Depth Prior (RGB-D)

The log-likelihood prior of Eq. 6 lends itself to an extension that ingests measured depth values d_i^* if available, namely for RGB-D input images. In this case, we substitute the broad distribution over plausible depth values with a narrow distribution centered at the measured depth value for each pixel:

$$\log p(\mathbf{y_i}) \coloneqq \lambda_{\text{reg}} \ \log \text{Lap}(d_i \mid d_i^*, b'). \tag{8}$$

Here, the bandwidth b' is a user-defined value that controls the tolerance of the prior. We will use $b' = 10$cm.

3.5. Point Cloud Diffusion Prior (RGB)

Finally, we propose adding a 3D diffusion prior to SCR training. The prior is pre-trained on a held out set of scenes, and encodes knowledge about plausible scene point clouds. During SCR training, the diffusion model is kept fixed, and nudges scene coordinates towards coherent scene layouts.

In forward diffusion, Gaussian noise ϵ is progressively added to a signal $\mathbf{x}_0$ over T time steps until the signal becomes completely noisy. Similarly, SCR training incrementally recovers the scene point cloud from a random initialization. Therefore, SCR training aligns with reversed diffusion, and a 3D diffusion model can guide SCR optimization.

Diffusion models [37, 85] learn to recover the original signal $\mathbf{x}_0$ by estimating the noise added to the noisy signal $\mathbf{x}_\tau$ at time step $\tau \in [0, T-1]$, through a neural network ϵ_θ. It has been shown in [37, 91, 99] that the noise estimate is proportional to the score function of the input signal, *i.e.*

$$\nabla_{\mathbf{x}} \log p(\mathbf{x}) \coloneqq -\lambda_{\text{reg}} \epsilon_\theta(\mathbf{x}_\tau, \tau). \tag{9}$$

In our context, this allows us to utilize a diffusion denoising model ϵ_θ as prior over 3D coordinates $\mathbf{y}$, because the prediction of the model corresponds to the gradient of the log-likelihood of scene coordinates. This prior is inspired by DiffusioNeRF [99] with the main difference that we regularize directly in 3D.

Architecture Our noise estimator $\epsilon_\theta(\mathbf{x}_\tau, \tau)$ takes the noisy point cloud $\mathbf{x}_\tau \in \mathbb{R}^{N \times 3}$ and the diffusion timestep τ as inputs. We adopt PVCNN [52] with certain modifications made by [57] for diffusion timestep embedding.

Training We follow the training protocol in DDPM [37]. During each forward iteration, we first normalize the input point cloud $\mathbf{x}_0$ with a predefined scaling factor to ensure the points are within the range $[-1, 1]$. We then transform $\mathbf{x}_0$ to a noisy version $\mathbf{x}_\tau$ by adding noise to $\mathbf{x}_0$ according to the noise schedule at a randomly sampled time step τ. The training objective of the diffusion model is expressed as:

$$L_{\text{training}} = \mathbb{E}_{\tau, \epsilon \sim \mathcal{N}(0,1), \mathbf{x}_0} \left[\|(\epsilon - \epsilon_\theta(\mathbf{x}_\tau, \tau))\|_2^2 \right]. \tag{10}$$

Inference Once the diffusion model is trained, we integrate it into SCR mapping. We take the estimated scene points $\mathbf{y}$ as input for the diffusion model and estimate the noise. We use the estimate as regularization as specified by Eq. (9). The distributions of the Gaussian noise and the noise encountered during ACE mapping differ, particularly during the first mapping iterations. Therefore, we apply the diffusion prior only after iteration 5k of ACE mapping. We align the diffusion time $\frac{T}{20}$ with ACE iteration 5k, and linearly interpolate timestep τ down to 0 as training progresses. We do not apply the diffusion prior to points with a

Table 1. **Reconstruction of ScanNet [23] and Indoor6 [28].** We compare results of ACE0 without and with our priors added. We report the percentage of images registered to the reconstruction (*Reg. Rate*), absolute trajectory and relative pose errors (*ATE*/*RPE*) as well as median pose errors compared to pseudo ground truth (pGT), and PSNR of novel view synthesis using two separate evaluation splits: the usual split with one test frame / seven training frames, and a harder split alternating between 60 test frames and 60 training frames.

			Comparison to Pose pGT (BundleFusion)		Splatfacto PSNR (dB) ↑		Indoor6	
		Reg. Rate ↑	ATE / RPE (cm) ↓	Med. Err. (cm/°) ↓	1/7	60/60	Reg. Rate ↑	PSNR (dB) ↑
RGB	ACE0	98.1%	26.6 / 4.0	19.7 / 9.0	30.2	22.3	57.1%	13.5
	ACE0 + Laplace NLL (Ours)	**98.9%**	25.4 / **3.5**	17.5 / 8.8	30.2	22.2	58.0%	14.1
	ACE0 + Laplace WD (Ours)	98.7%	25.9 / 3.6	17.5 / 6.8	30.3	21.7	57.7%	14.1
	ACE0 + Diffusion (Ours)	98.6%	26.5 / 3.8	18.8 / 8.9	30.2	22.4	**61.8%**	**14.6**
RGB-D	ACE0 + DSAC* Loss	96.2%	29.2 / 6.0	20.9 / 5.9	30.0	21.9	N/A	N/A
	ACE0 + Laplace NLL (Ours)	**98.9%**	**18.3 / 3.5**	**12.8 / 4.4**	**30.6**	**22.9**	N/A	N/A

reprojection error smaller than 30 pixels, as we assume that sufficient multi-view constraints exist for those points.

4. Experiments

Implementation Details We build on the official PyTorch [66] implementation of ACE0 [12]. Unless otherwise specified, we use the same parameters as ACE0 [12] for reconstruction, and the same parameters as ACE [11] for relocalization. We use the ACE feature backbone which was pre-trained on 100 ScanNet training scenes. We also integrate the diffusion prior into GLACE [93]. In this variant, we maintain the regularization identical to its implementation in ACE, while keeping all other settings of GLACE.

Training the Priors We fit our priors to the training set of ScanNetV2 [23] which consists of 706 scenes, densely reconstructed from RGB-D images. For the depth distribution prior (Sec. 3.3), we fit a Laplace distribution to randomly sampled depth values of the training images, yielding a mean of $\mu = 1.73$m and a bandwidth of $b = 60$cm. For training the diffusion model (Sec. 3.5), we use the ground truth point cloud of each training scene from ScanNetV2 as target. In each iteration, we randomly sample 5,120 points from a point cloud in the training dataset and apply augmentations including random rotation, translation and scaling within a certain range. The total number of diffusion time steps is set to 200. We train the diffusion model with a batch size of 16 for 100,000 iterations on a single V100 GPU. We use AdamW [44], with a learning rate that decays linearly from 0.0002 to 0 throughout the training process. The diffusion model is frozen after training and applied during the mapping stage across all scenes and datasets.

4.1. Structure-from-Motion

We reconstruct a number of indoor scenes using ACE0 [12], with and without incorporating our priors.

Datasets We evaluate on the first 20 test scenes of ScanNetV2 [23], and on the mapping sequences of Indoor6 [28]. Where ScanNet consists of single room scans, Indoor6 features six larger indoor environments comprised of multiple rooms. Indoor6 does not come with RGB-D images.

Metrics We report the percentage of images successfully registered to the reconstruction (*Reg. Rate*), *i.e.* images with a final inlier count above 1,000 [12]. We confirm the quality of estimated camera poses using novel view synthesis [12, 92]. That is, we estimate camera poses using all images. Then, we divide images into training and validation images for Splatfacto [43, 90, 102] to check whether we can re-render the scene based on the estimated camera poses. We report *PSNR* numbers on the usual training/validation split of Splatfacto, taking one validation image for every seven training images. For ScanNet we also report results for a harder split, alternating between 60 validation images and 60 training images. Finally, for ScanNet, we compare the estimated poses to the pseudo ground truth (pGT, [10]) that comes with the dataset, estimated by BundleFusion [24], a RGB-D SLAM system that exploits the ordering of images. For reference, BundleFusion achieves a PSNR of 22.2dB on the 60/60 Splatfacto split, *lower* that some of our results. Still, SLAM poses serve as a suitable reference in terms of global consistency, and we report the absolute trajectory error (ATE) and relative pose errors (RPE) [88], as well as median rotation and translation errors after least-squares alignment to the pGT camera trajectory.

Discussion We report results in Table 1. We couple our Laplace depth distribution prior (Sec. 3.3) with log-likelihood optimization (*Laplace NLL*) as well as the Wasserstein distance (*Laplace WD*). Both variations of the prior increase the registration rates and pose quality on ScanNet. The novel view synthesis quality is largely on par with ACE0, except for a noticeable drop in PSNR for the 60/60 evaluation split when using the Wasserstein distance. Effects are stronger on Indoor6 where ACE0 struggles due to the size of the scenes. Both depth distribution priors lead to higher registration rates, and increase PSNR by 0.6 dB.

Table 2. **Relocalization Results on 7Scenes [82].** We report the percentage of test images below a 5cm/5° pose error (higher is better), mapping time and map size. Methods in "SCR w/ 3D" use depth or 3D point cloud supervision during mapping. Best results within the SCR groups are highlighted in **bold**. All methods use SfM mapping poses.

Type	Method	Mapping Time	Map Size	Chess	Fire	Heads	Office	Pumpkin	Redkitchen	Stairs	Avg
FM	AS (SIFT) [77]	∼1.5h	∼200MB	N/A	N/A	N/A	N/A	N/A	N/A	N/A	**98.5%**
	D.VLAD+R2D2 [38]	∼1.5h	∼1GB	N/A	N/A	N/A	N/A	N/A	N/A	N/A	95.7%
	hLoc (SP+SG) [74, 75]	∼1.5h	∼2GB	N/A	N/A	N/A	N/A	N/A	N/A	N/A	95.7%
SCR w/ 3D	DSAC* [7]	15h	28MB	99.8%	98.9%	99.8%	98.5%	98.9%	97.8%	93.8%	**98.2%**
	SRC [29]	2min	40MB	89.1%	79.6%	97.6%	84.1%	65.7%	87.3%	64.7%	81.1%
	FocusTune [61]	6min	4MB	99.7%	99.0%	87.1%	99.9%	**99.9%**	**100%**	**99.5%**	97.9%
	ACE [11, 12] + DSAC* Loss [7]	5.5min	4MB	**100%**	**99.5%**	**100%**	98.8%	96.2%	98.2%	82.3%	96.4%
	ACE + Laplace NLL (Ours)	5.5min	4MB	**100%**	99.2%	**100%**	**99.8%**	97.2%	98.4%	87.3%	97.4%
SCR	DSAC* [7]	15h	28MB	**100%**	99.1%	98.8%	99.3%	99.4%	96.9%	78.6%	96.0%
	GLACE [93]	6min	9MB	**100%**	**99.9%**	99.9%	99.8%	**100%**	98.2%	71.2%	95.6%
	GLACE + Diffusion (Ours)	9min	9MB	**100%**	99.8%	**100%**	99.9%	99.5%	98.8%	73.6%	95.9%
	ACE [11]	5min	4MB	**100%**	99.5%	99.7%	**100%**	99.9%	98.6%	81.9%	97.1%
	ACE + Laplace NLL (Ours)	4.5min	4MB	**100%**	98.9%	99.9%	99.9%	**100%**	98.5%	84.2%	97.3%
	ACE + Laplace WD (Ours)	4.5min	4MB	**100%**	99.6%	**100%**	99.9%	99.8%	98.2%	83.2%	97.2%
	ACE + Diffusion (Ours)	8min	4MB	**100%**	99.5%	**100%**	**100%**	99.0%	**99.1%**	**86.2%**	**97.7%**

The diffusion prior yields slight but consistent improvements across all metrics on ScanNet. On Indoor6, the diffusion prior causes an increase in registered images (+4.7%) leading to higher PSNR (+1.1 dB). The diffusion prior takes the global scene layout into account, providing a stronger regularization signal than our depth distribution priors.

On ScanNet, we also test a version of ACE0 with our RGB-D prior (Sec. 3.4). We compare to an ACE0 RGB-D baseline where we activate the ACE0 RGB-D loss, usually only applied when initializing the reconstruction, throughout the entire reconstruction. The ACE0 RGB-D loss is inspired by DSAC* [7] and optimizes the distance of predicted scene coordinates to ground truth derived from depth maps. The loss switches to the reprojection error, when predicted scene coordinates are within 10cm of the ground truth. This RGB-D baseline of ACE0 (denoted *ACE0 + DSAC* Loss*) performs worse than the default ACE0 on ScanNet, presumably due to the hard switch between RGB and RGB-D terms reducing optimization stability. In contrast, ACE0 coupled with our depth prior, derived from our probabilistic interpretation of SCR optimization, yields significant improvements across all metrics.

4.2. Relocalization Results

We evaluate performance of our priors on the relocalization task using the 7Scenes and Indoor6 datasets, comparing it against other Scene Coordinate Regression (SCR) methods and feature-matching (FM) approaches. We report the percentage of images localized within a 5cm/5° threshold [82].

7Scenes Results in Tab. 2 demonstrate that ACE coupled with our priors achieves higher relocalization accuracy than ACE alone. The diffusion prior has the strongest effect, improving results by 4.1% on the most difficult scene, *Stairs*. The supplement includes results using alternative ground truth poses [10] where the same trend holds. In Fig. 3, we show qualitatively how the diffusion prior leads to a more compact representation for this scene. To underline the versatility of our priors, we report results of GLACE [93], a recent ACE extension, coupling it with our diffusion prior. Again, relocalization results improve on average, particularly on the Stairs scene. In the supplement, we analyze the effect of our priors on the point clouds learned by ACE.

In terms of mapping time, our depth distribution priors lead to simplified optimization objectives compared to ACE, resulting in a slight reduction in mapping time. On the other hand, incorporating the diffusion prior incurs additional cost, as we need to execute a separate model many times throughout ACE mapping. However, the nominal increase in mapping time is still modest, with +3 minutes.

Table 3. **Relocalization Results on Indoor6 [28].** We report relocalization accuracy (5cm, 5°) and mapping time. Except for EGFS, we report all results as mean over 5 runs. Best results **bold** within each pair w/ and w/o our diffusion prior (denoted *-Diff*).

		Reloc. Acc.	Map. Time
N=5,120	EGFS [50]	56.1%	21 min
	GLACE [93]	44.2% ± 1.8%	11 min
	GLACE-Diff (Ours)	**46.4**% ± 1.9%	15 min
	ACE [11]	36.2% ± 1.5%	5 min
	ACE-Diff (Ours)	**37.5**% ± 1.8%	8 min
N=51,200	GLACE [93]	69.5% ± 1.4%	33 min
	GLACE-Diff (Ours)	**69.6%** ± 2.0%	40 min
	ACE [11]	57.2% ± 1.6%	10 min
	ACE-Diff (Ours)	**57.9**% ± 1.1%	13 min

Indoor6 In Tab. 3, we report the average relocalization accuracy on Indoor6 for ACE [11]; GLACE [93] with and

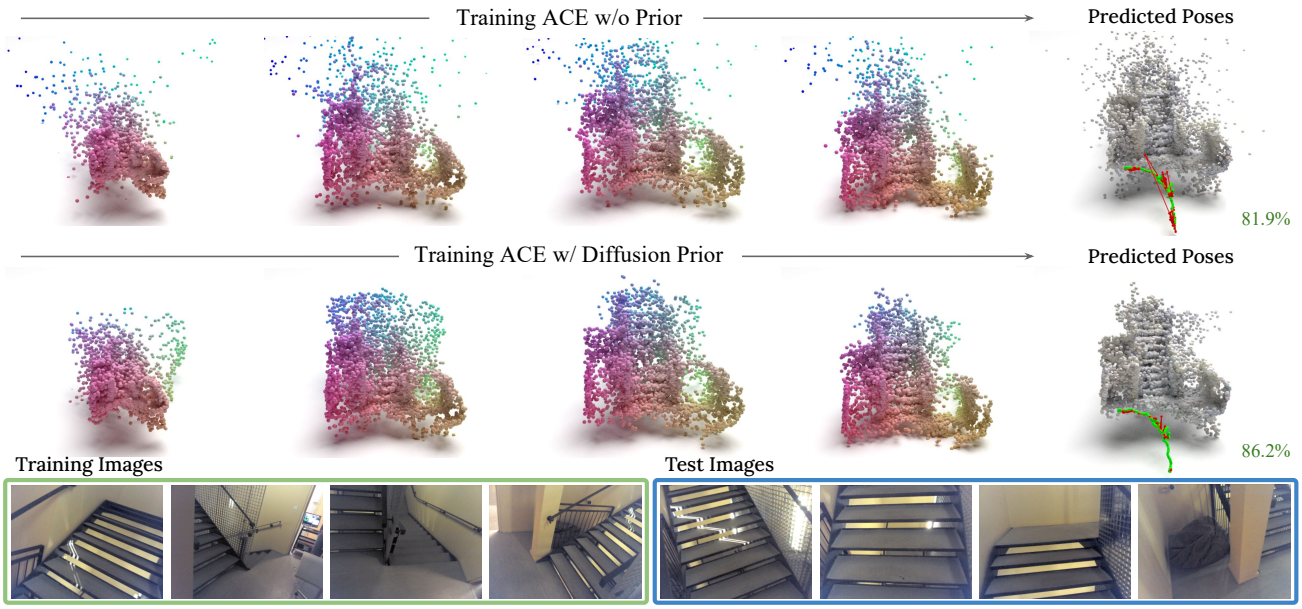

Figure 3. **Diffusion Prior on Stairs.** We guide the training of ACE [11] on the Stairs scene with our 3D diffusion model, leading to a more coherent scene geometry, and higher pose estimation accuracy on test images (5cm,5° threshold).

Figure 4. **Generated Point Clouds.** Point clouds generated by our diffusion model together with ScanNet point clouds used for training.

without our diffusion prior. Our experiments show that increasing the batch size significantly improves the localization accuracy, presumably due to the larger size of the scenes. Therefore, we evaluated two versions of each method, using batch sizes of 5,120 and 51,200. We observed noticeable variance in relocalization accuracy on this dataset, and report mean results over 5 runs. Adding our prior helps on average, although ultimate conclusions are difficult due to the variance on this dataset. For per-scene results, please see the supplement. For EGFS [50], another ACE derivative, the code is not publicly available yet. We report results with batch size 5,120 from their paper. EGFS could be coupled with our diffusion prior, as well.

4.3. Prior Introspection

Fig. 4 shows point clouds generated by the diffusion model, compared with ScanNet scenes, illustrating the prior learned by the model. Although lacking fine details, the prior represents sensible room layouts sufficient to regularize SCR training. The supplement includes further analysis, regarding point cloud encoder architecture, efficiency, the prior's effect on scarce data and hyper-parameters.

5. Conclusion

We have presented a probabilistic reformulation of scene coordinate regression training that allows for the easy incorporation of reconstruction priors. We presented multiple potential priors: priors regularizing the distribution of reconstructed depth values, a prior leveraging measured depth, as well as a high-level learned prior in the form of a 3D point cloud diffusion model. We found that the regularization terms can be integrated into ACE-based frameworks, such as ACE, ACE0 or GLACE, yielding performance gains while not affecting the test-time latency.

Limitations Our experiments have focused on indoor scenes, with results on a few outdoor scenes in the supplement. Properly extending our approach to outdoor scenes requires better models of depth distribution, and more diverse data for diffusion training to learn a comprehensive prior over larger areas. A point cloud encoding network with better expressiveness could result in generations with higher fidelity, but it would need to remain efficient to be practical. An additional conditional signal for the diffusion model can also be a promising direction for future research.

References

[1] Eduardo Arnold, Jamie Wynn, Sara Vicente, Guillermo Garcia-Hernando, Áron Monszpart, Victor Adrian Prisacariu, Daniyar Turmukhambetov, and Eric Brachmann. Map-free visual relocalization: Metric pose relative to a single image. In *ECCV*, 2022. 2

[2] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. In *CVPR*, 2022. 2, 3

[3] Axel Barroso-Laguna, Sowmya Munukutla, Victor Prisacariu, and Eric Brachmann. Matching 2D images in 3D: Metric relative pose from metric correspondences. In *CVPR*, 2024. 2

[4] Eric Brachmann and Carsten Rother. Learning less is more - 6d camera localization via 3D surface regression. In *CVPR*, 2018. 1, 2

[5] Eric Brachmann and Carsten Rother. Expert sample consensus applied to camera re-localization. In *ICCV*, 2019. 2

[6] Eric Brachmann and Carsten Rother. Neural-guided RANSAC: Learning where to sample model hypotheses. In *ICCV*, 2019. 2

[7] Eric Brachmann and Carsten Rother. Visual camera relocalization from RGB and RGB-D images using DSAC. *IEEE TPAMI*, 2021. 1, 2, 3, 7

[8] Eric Brachmann, Frank Michel, Alexander Krull, Michael Y. Yang, Stefan Gumhold, and Carsten Rother. Uncertainty-driven 6D pose estimation of objects and scenes from a single RGB image. In *CVPR*, 2016. 1, 2

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

[10] Eric Brachmann, Martin Humenberger, Carsten Rother, and Torsten Sattler. On the limits of pseudo ground truth in visual camera re-localisation. In *ICCV*, 2021. 6, 7

[11] Eric Brachmann, Tommaso Cavallari, and Victor Adrian Prisacariu. Accelerated coordinate encoding: Learning to relocalize in minutes using RGB and poses. In *CVPR*, 2023. 1, 2, 3, 4, 6, 7, 8

[12] Eric Brachmann, Jamie Wynn, Shuai Chen, Tommaso Cavallari, Áron Monszpart, Daniyar Turmukhambetov, and Victor Adrian Prisacariu. Scene coordinate reconstruction: Posing of image collections via incremental learning of a relocalizer. In *ECCV*, 2024. 1, 2, 3, 4, 6, 7

[13] S. Brahmbhatt, J. Gu, K. Kim, J. Hays, and J. Kautz. Geometry-Aware Learning of Maps for Camera Localization. In *CVPR*, 2018. 2

[14] Matthew Brown and David G. Lowe. Unsupervised 3D object recognition and reconstruction in unordered datasets. In *3DIM*, 2005. 2

[15] Ruojin Cai, Guandao Yang, Hadar Averbuch-Elor, Zekun Hao, Serge Belongie, Noah Snavely, and Bharath Hariharan. Learning gradient fields for shape generation. In *ECCV*, 2020. 3

[16] Tommaso Cavallari, Stuart Golodetz, Nicholas A Lord, Julien Valentin, Luigi Di Stefano, and Philip HS Torr. On-the-fly adaptation of regression forests for online camera relocalisation. In *CVPR*, 2017. 2

[17] Tommaso Cavallari, Luca Bertinetto, Jishnu Mukhoti, Philip Torr, and Stuart Golodetz. Let's take this online: Adapting scene coordinate regression network predictions for online RGB-D camera relocalisation. In *3DV*, 2019.

[18] Tommaso Cavallari, Stuart Golodetz, Nicholas A. Lord, Julien Valentin, Victor A. Prisacariu, Luigi Di Stefano, and Philip H. S. Torr. Real-time RGB-D camera pose estimation in novel scenes using a relocalisation cascade. *IEEE TPAMI*, 2019. 2

[19] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An information-rich 3D model repository. *arXiv*, 2015. 2, 3

[20] Shuai Chen, Zirui Wang, and Victor Prisacariu. Direct-PoseNet: Absolute pose regression with photometric consistency. In *3DV*, 2021. 2

[21] Shuai Chen, Xinghui Li, Zirui Wang, and Victor Prisacariu. DFNet: Enhance absolute pose regression with direct feature matching. In *ECCV*, 2022. 2

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

[23] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *CVPR*, 2017. 2, 3, 6

[24] Angela Dai, Matthias Nießner, Michael Zollöfer, Shahram Izadi, and Christian Theobalt. BundleFusion: Real-time globally consistent 3D reconstruction using on-the-fly surface re-integration. *ACM TOG*, 2017. 6

[25] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, Eli VanderBilt, Aniruddha Kembhavi, Carl Vondrick, Georgia Gkioxari, Kiana Ehsani, Ludwig Schmidt, and Ali Farhadi. Objaverse-XL: A universe of 10M+ 3D objects. *arXiv*, 2023. 2

[26] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3D objects. In *CVPR*, 2023. 2

[27] Prafulla Dhariwal and Alex Nichol. Diffusion models beat GANs on image synthesis. In *NeurIPS*, 2021. 3

[28] Tien Do, Ondrej Miksik, Joseph DeGol, Hyun Soo Park, and Sudipta N Sinha. Learning to detect scene landmarks for camera localization. In *CVPR*, 2022. 6, 7

[29] Siyan Dong, Shuzhe Wang, Yixin Zhuang, Juho Kannala, Marc Pollefeys, and Baoquan Chen. Visual localization via few-shot scene region classification. In *3DV*, 2022. 7

[30] Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. Roma: Robust dense feature matching. In *CVPR*, 2024. 2

[31] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981. 3

[32] Ruiqi Gao, Aleksander Holynski, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul P. Srinivasan, Jonathan T. Barron, and Ben Poole. CAT3D: Create anything in 3D with multi-view diffusion models. In *NeurIPS*, 2024. 3

[33] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE TPAMI*, 2003. 3

[34] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2003. 1, 2

[35] Xingyi He, Jiaming Sun, Yifan Wang, Sida Peng, Qixing Huang, Hujun Bao, and Xiaowei Zhou. Detector-free structure from motion. In *CVPR*, 2024. 2

[36] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS Workshop*, 2021. 3

[37] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 3, 5

[38] Martin Humenberger, Yohann Cabon, Nicolas Guerin, Julien Morat, Vincent Leroy, Jérôme Revaud, Philippe Rerole, Noé Pion, Cesar de Souza, and Gabriela Csurka. Robust image retrieval-based visual localization using kapture. *arXiv*, 2020. 7

[39] Martin Humenberger, Yohann Cabon, Nicolas Guerin, Julien Morat, Jérôme Revaud, Philippe Rerole, Noé Pion, Cesar de Souza, Vincent Leroy, and Gabriela Csurka. Robust image retrieval-based visual localization using Kapture. *arXiv*, 2020. 2

[40] Yuhe Jin, Dmytro Mishkin, Anastasiia Mishchuk, Jiri Matas, Pascal Fua, Kwang Moo Yi, and Eduard Trulls. Image Matching across Wide Baselines: From Paper to Practice. *IJCV*, 2020. 2

[41] A. Kendall and R. Cipolla. Geometric loss functions for camera pose regression with deep learning. In *CVPR*, 2017. 2

[42] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-DOF camera relocalization. In *CVPR*, 2015. 2

[43] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D gaussian splatting for real-time radiance field rendering. *ACM TOG*, 2023. 6

[44] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv*, 2014. 6

[45] Jumin Lee, Woobin Im, Sebin Lee, and Sung-Eui Yoon. Diffusion probabilistic models for scene-scale 3D categorical data. *arXiv*, 2023. 3

[46] Vincent Leroy, Yohann Cabon, and Jerome Revaud. Grounding image matching in 3D with MASt3R. In *ECCV*, 2024. 2

[47] Xiaotian Li, Shuzhe Wang, Yi Zhao, Jakob Verbeek, and Juho Kannala. Hierarchical scene coordinate classification and regression for visual localization. In *CVPR*, 2020. 2

[48] Zhengqi Li and Noah Snavely. MegaDepth: Learning single-view depth prediction from internet photos. In *CVPR*, 2018. 2

[49] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. LightGlue: Local feature matching at light speed. In *ICCV*, 2023. 2

[50] Ting-Ru Liu, Hsuan-Kung Yang, Jou-Min Liu, Chun-Wei Huang, Tsung-Chih Chiang, Quan Kong, Norimasa Kobori, and Chun-Yi Lee. Reprojection errors as prompts for efficient scene coordinate regression. In *ECCV*, 2025. 7, 8

[51] Yuheng Liu, Xinke Li, Xueting Li, Lu Qi, Chongshou Li, and Ming-Hsuan Yang. Pyramid diffusion for fine 3D large scene generation. In *ECCV*, 2024. 3

[52] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel cnn for efficient 3D deep learning. *NeurIPS*, 2019. 5

[53] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel CNN for efficient 3D deep learning. In *NeurIPS*, 2019. 3

[54] Calvin Luo. Understanding diffusion models: A unified perspective. *arXiv*, 2022. 3

[55] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3D point cloud generation. In *CVPR*, 2021. 2, 3

[56] Zhaoyang Lyu, Zhifeng Kong, Xudong Xu, Liang Pan, and Dahua Lin. A conditional point diffusion-refinement paradigm for 3D point cloud completion. In *ICLR*, 2022. 3

[57] Luke Melas-Kyriazi, Christian Rupprecht, and Andrea Vedaldi. PC2: Projection-conditioned point cloud diffusion for single-image 3D reconstruction. In *CVPR*, 2023. 2, 3, 5

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

[59] Shentong Mo, Enze Xie, Ruihang Chu, Lanqing HONG, Matthias Nießner, and Zhenguo Li. DiT-3D: Exploring plain diffusion transformers for 3D shape generation. In *NeurIPS*, 2023. 3

[60] Arthur Moreau, Nathan Piasco, Dzmitry Tsishkou, Bogdan Stanciulescu, and Arnaud de La Fortelle. LENS: Localization enhanced by NeRF synthesis. In *CoRL*, 2021. 2

[61] Son Tung Nguyen, Alejandro Fontan, Michael Milford, and Tobias Fischer. Focustune: Tuning visual localization through focus-guided sampling. In *WACV*, 2024. 7

[62] Thang-Anh-Quan Nguyen, Nathan Piasco, Moussab Bennehar Luis Roldão, Dzmitry Tsishkou, Laurent Caraffa, Jean-Philippe Tarel, and Roland Brémond. Pointmap-conditioned diffusion for consistent novel view synthesis. *arXiv*, 2025. 3

[63] Michael Niemeyer, Jonathan T. Barron, Ben Mildenhall, Mehdi S. M. Sajjadi, Andreas Geiger, and Noha Radwan. RegNeRF: Regularizing neural radiance fields for view synthesis from sparse inputs. In *CVPR*, 2022. 2

[64] Lucas Nunes, Rodrigo Marcuzzi, Benedikt Mersch, Jens Behley, and Cyrill Stachniss. Scaling diffusion models to real-world 3D LiDAR scene completion. In *CVPR*, 2024. 3

[65] Linfei Pan, Dániel Baráth, Marc Pollefeys, and Johannes Lutz Schönberger. Global structure-from-motion revisited. In *ECCV*, 2024. 2

[66] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NeurIPS Workshop*, 2017. 6

[67] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. DreamFusion: Text-to-3D using 2D diffusion. In *ICLR*, 2023. 3

[68] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *CVPR*, 2017. 3

[69] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, 2017. 3

[70] Haoxi Ran, Vitor Guizilini, and Yue Wang. Towards realistic scene generation with LiDAR diffusion models. In *CVPR*, 2024. 3

[71] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3D: Large-scale learning and evaluation of real-life 3D category reconstruction. In *ICCV*, 2021. 3

[72] Jerome Revaud, Yohann Cabon, Romain Brégier, JongMin Lee, and Philippe Weinzaepfel. SACReg: Scene-agnostic coordinate regression for visual localization. In *CVPRW*, 2024. 3

[73] Barbara Roessle, Jonathan T. Barron, Ben Mildenhall, Pratul P. Srinivasan, and Matthias Nießner. Dense depth priors for neural radiance fields from sparse input views. In *CVPR*, 2022. 2, 3

[74] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019. 2, 7

[75] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 2, 7

[76] T. Sattler, B. Leibe, and L. Kobbelt. Fast image-based localization using direct 2D-to-3D matching. In *ICCV*, 2011. 2

[77] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient & effective prioritized matching for large-scale image-based localization. *PAMI*, 2016. 7

[78] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient & effective prioritized matching for large-scale image-based localization. *IEEE TPAMI*, 2017. 2

[79] Torsten Sattler, Qunjie Zhou, Marc Pollefeys, and Laura Leal-Taixe. Understanding the limitations of cnn-based absolute camera pose regression. In *CVPR*, 2019. 2

[80] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 2

[81] Yoli Shavit, Ron Ferens, and Yosi Keller. Learning multi-scene absolute pose regression with transformers. In *ICCV*, 2021. 2

[82] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in RGB-D images. In *CVPR*, 2013. 1, 2, 3, 7

[83] Noah Snavely, Steven Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3D. *ACM TOG*, 2006. 2

[84] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015. 3

[85] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. 3, 5

[86] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *NeurIPS*, 2019. 3

[87] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021. 3

[88] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *IROS*, 2012. 6

[89] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. In *CVPR*, 2021. 2

[90] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Justin Kerr, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David McAllister, and Angjoo Kanazawa. Nerfstudio: A modular framework for neural radiance field development. In *SIGGRAPH*, 2023. 1, 6

[91] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 2011. 5

[92] Michael Waechter, Mate Beljan, Simon Fuhrmann, Nils Moehrle, Johannes Kopf, and Michael Goesele. Virtual rephotography: Novel view prediction error for 3D reconstruction. *ACM TOG*, 2017. 6

[93] Fangjinhua Wang, Xudong Jiang, Silvano Galliani, Christoph Vogel, and Marc Pollefeys. GLACE: Global local accelerated coordinate encoding. In *CVPR*, 2024. 2, 3, 6, 7

[94] Jianyuan Wang, Nikita Karaev, Christian Rupprecht, and David Novotny. VGGSfM: Visual geometry grounded deep structure from motion. In *CVPR*, 2024. 2

[95] Shuzhe Wang, Zakaria Laskar, Iaroslav Melekhov, Xiaotian Li, Yi Zhao, Giorgos Tolias, and Juho Kannala. HSCNet++ : Hierarchical scene coordinate classification and regression for visual localization with transformer. *IJCV*, 2024. 2

[96] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3D vision made easy. In *CVPR*, 2024. 2

[97] Changchang Wu. Towards linear-time incremental structure from motion. In *3DV*, 2013. 2

[98] Rundi Wu, Ben Mildenhall, Philipp Henzler, Keunhong Park, Ruiqi Gao, Daniel Watson, Pratul P. Srinivasan, Dor Verbin, Jonathan T. Barron, Ben Poole, and Aleksander Holynski. ReconFusion: 3D reconstruction with diffusion priors. In *CVPR*, 2024. 3

[99] Jamie Wynn and Daniyar Turmukhambetov. DiffusioNeRF: Regularizing neural radiance fields with denoising diffusion models. In *CVPR*, 2023. 2, 3, 5

[100] Haolin Xiong, Sairisheek Muttukuru, Rishi Upadhyay, Pradyumna Chari, and Achuta Kadambi. SparseGS: Real-

time 360° sparse view synthesis using gaussian splatting. *arXiv*, 2023. 2

[101] Luwei Yang, Ziqian Bai, Chengzhou Tang, Honghua Li, Yasutaka Furukawa, and Ping Tan. SANet: Scene agnostic network for camera localization. In *ICCV*, 2019. 3

[102] Vickie Ye, Ruilong Li, Justin Kerr, Matias Turkulainen, Brent Yi, Zhuoyang Pan, Otto Seiskari, Jianbo Ye, Jeffrey Hu, Matthew Tancik, and Angjoo Kanazawa. gsplat: An open-source library for Gaussian splatting. *arXiv*, 2024. 6

[103] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. ScanNet++: A high-fidelity dataset of 3D indoor scenes. In *ICCV*, 2023. 2

[104] Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. LION: Latent point diffusion models for 3D shape generation. In *NeurIPS*, 2022. 3

[105] Linqi Zhou, Yilun Du, and Jiajun Wu. 3D shape generation and completion through point-voxel diffusion. In *ICCV*, 2021. 2, 3

[106] Vlas Zyrianov, Xiyue Zhu, and Shenlong Wang. Learning to generate realistic LiDAR point cloud. In *ECCV*, 2022. 3