PoseDiffusion: Solving Pose Estimation via Diffusion-aided Bundle Adjustment

Jianyuan Wang[1,2] Christian Rupprecht[1] David Novotny[2]
jianyuan@robots.ox.ac.uk chrisr@robots.ox.ac.uk dnovotny@meta.com

[1]Visual Geometry Group, University of Oxford [2]Meta AI

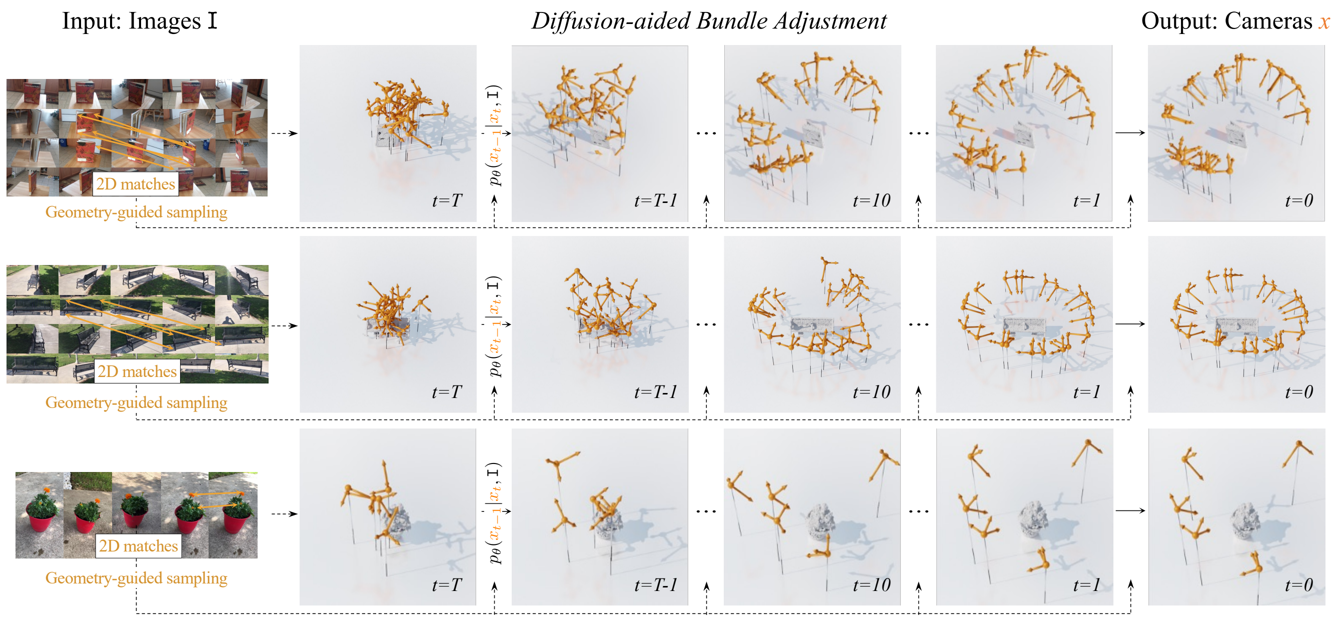

Figure 1: **Camera Pose Estimation with PoseDiffusion.** We present a method to predict the camera parameters (extrinsics and intrinsiscs) for a given collection of scene images. Our model combines the strengths of traditional epipolar constraints from point correspondences with the power of diffusion models to iteratively refine an initially random set of poses.

Abstract

Camera pose estimation is a long-standing computer vision problem that to date often relies on classical methods, such as handcrafted keypoint matching, RANSAC and bundle adjustment. In this paper, we propose to formulate the Structure from Motion (SfM) problem inside a probabilistic diffusion framework, modelling the conditional distribution of camera poses given input images. This novel view of an old problem has several advantages. (i) The nature of the diffusion framework mirrors the iterative procedure of bundle adjustment. (ii) The formulation allows a seamless integration of geometric constraints from epipolar geometry. (iii) It excels in typically difficult scenarios such as sparse views with wide baselines. (iv) The method can predict intrinsics and extrinsics for an arbitrary amount of images. We demonstrate that our method PoseDiffusion significantly improves over the classic SfM pipelines and the learned approaches on two real-world datasets. Finally, it is observed that our method can generalize across datasets without further training. Project page: `https://posediffusion.github.io/`

1. Introduction

Camera pose estimation, *i.e.* extracting the camera intrinsics and extrinsics given a set of free-form multi-view scene-centric images (*e.g.* tourist photos of Rome [2]), is a traditional Computer Vision problem with a history stretching long before the inception of modern computers [20].

It is a crucial task in various applications, including augmented and virtual reality, and has recently regained the attention of the research community due to the emergence of implicit novel-view synthesis methods [33, 38, 25].

The classic dense pose estimation task estimates the parameters of many cameras with overlapping frusta, leveraging correspondence pairs between keypoints visible across images. It is typically addressed through a Structure-from-Motion (SfM) framework, which not only estimates the camera pose (Motion) but also extracts the 3D shape of the observed scene (Structure). During the last 30 years, SfM pipelines matured into a technology capable of reconstructing thousands [2] if not millions [14] of free-form views.

Surprisingly, the structure of dense-view SfM pipeline [41] has remained mostly unchanged until today, even though individual components have incorporated deep learning advances [8, 40, 17, 49, 53, 24]. SfM first estimates reliable image-to-image correspondences and, later, uses Bundle Adjustment (BA) to align all cameras into a common scene-consistent reference frame. Due to the high complexity of the BA optimization landscape, a modern SfM pipeline [44] comprises a carefully engineered iterative process alternating between expanding the set of registered poses and a precise 2nd-order BA optimizer [1].

With the recent proliferation of deep geometry learning, the sparse pose problem, operating on a significantly smaller number of input views separated by wide baselines, has become of increasing interest. For many years, this sparse setting has been the Achilles' Heel of traditional pose estimation methods. Recently, RelPose [60] leveraged a deep network to implicitly learn a bundle-adjustment prior from a large dataset of images and corresponding camera poses. The method has demonstrated performance superior to SfM in settings with less than ten input frames. However, in the many-image case, its accuracy cannot match the precise solution of the second-order BA optimizer from iterative SfM. Besides, it is limited to predicting rotations only.

In this paper, we propose PoseDiffusion - a novel camera pose estimation approach that elegantly marries deep learning with correspondence-based constraints and therefore, is able to reconstruct camera positions at high accuracy both in the sparse-view and dense-view regimes.

PoseDiffusion introduces a diffusion framework to solve the bundle adjustment problem by modelling the probability $p(x|\mathtt{I})$ of camera parameters x given observed images $\mathtt{I}$. Following the recent successes of diffusion models in modelling complex distributions (*e.g.* over images [15], videos [45], and point clouds [28]), we leverage diffusion models to learn $p(x|\mathtt{I})$ from a large dataset of images with known camera poses. Once learned, given a previously unseen sequence, we estimate the camera poses x by sampling $p(x|\mathtt{I})$. The latter mildly assumes that $p(x|\mathtt{I})$ forms a near-delta distribution so that any sample from $p(x|\mathtt{I})$ will yield a valid pose. The stochastic sampling process of diffusion models has been shown to efficiently navigate the log-likelihood landscape of complex distributions [15], and therefore is a perfect fit for the intricate BA optimization. An additional benefit of the diffusion process is that it can be trained one step at a time without the need for unrolling gradients through the whole optimization.

Additionally, in order to increase the precision of our camera estimation, we guide the sampling process with traditional epipolar constraints (expressed by means of reliable 2D image-to-image correspondences), which is inspired by classifier diffusion guidance [9]. We use this classical constraint to bias samples towards more geometrically consistent solutions throughout the sampling process, arriving at a more precise camera estimation.

PoseDiffusion yields state-of-the-art accuracy on the object-centric scenes of CO3Dv2 [38], as well as on outdoor/indoor scenes of RealEstate10k [62]. Crucially, PoseDiffusion also outperforms SfM methods when used to supervise NeRF training [33], which demonstrates the superior accuracy of both the extrinsic and intrinsic estimation.

2. Related Work

Geometric Pose Estimation. The technique of estimating camera poses given image-to-image point correspondences has been extensively explored in the last three decades [12, 37]. This process typically begins with keypoint detection, conducted by handcrafted methods like SIFT [26, 27] and SURF [3], or alternatively, learned methods [8, 58]. The correspondences can then be established using nearest neighbour search or learned matchers [40, 31, 59]. Given these correspondences, five-point or eight-point algorithms compute camera poses [12, 13, 21, 36] with the help of RANSAC and its variants [10, 4, 5]. Typically, Bundle Adjustment [50] further optimizes the camera poses. The entire pipeline, from keypoint detection to bundle adjustment, is highly interdependent and needs careful tuning to be sufficiently robust, which allows for scaling to thousands of images [11, 39]. COLMAP [44, 43] is an open-source implementation of the whole camera estimation procedure and has become a valuable asset to the community.

Learned Pose Estimation. Geometric pose estimation techniques struggle when only few image-to-image matches can be established, or more generally, in a setting with sparse views and wide baselines [7]. Thus, instead of constructing geometric constraints on top of potentially unreliable point matches, learning-based approaches directly estimate the camera motion between frames. Learning can be driven by ground truth annotations or unsupervisedly through reprojecting points from one frame to another, measuring photometric reconstruction [61, 51, 49]. Learned methods that directly predict the relative transformation

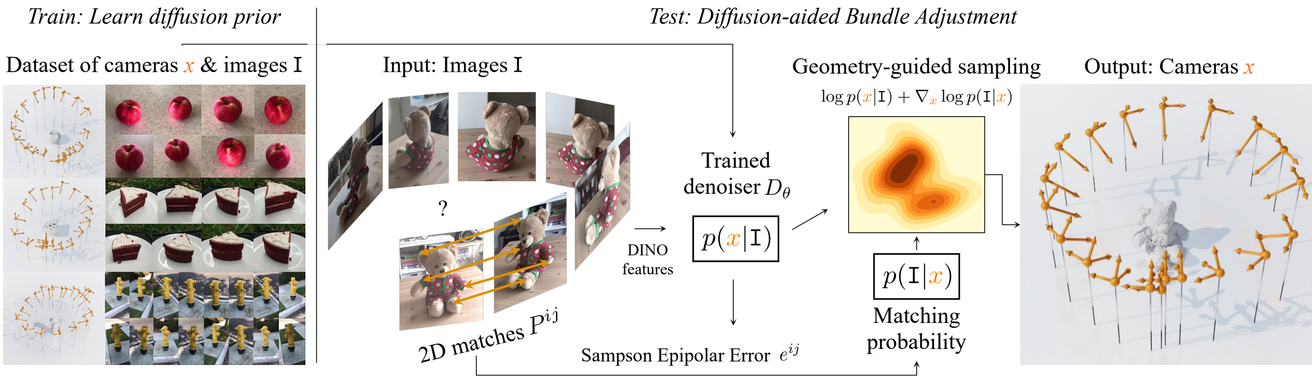

Figure 2: **PoseDiffusion overview.** Training is supervised given a multi-view dataset of images and camera poses to learn a diffusion model D_θ to model $p(x|\mathtt{I})$. During inference the reverse diffusion process is guided through the gradient that minimizes the Sampson Epipolar Error between image pairs, optimizing geometric consistency between poses.

between camera poses are often category-specific or object centric [18, 57, 30, 56, 55]. Recently, RelPose [60] shows category-agnostic camera pose estimation, however, is limited to predicting rotations. The concurrent work SparsePose [46] first regresses camera poses followed by iterative refinement, while RelPose++ [22] decouples the ambiguity in rotation estimation from translation prediction by defining a new coordinate system.

Diffusion Model. Diffusion models are a category of generative models that, inspired by non-equilibrium thermodynamics [47], approximate the data distribution by a Markov Chain of diffusion steps. Recently, they have shown impressive results on image [48, 15], video [45, 16], and even 3D point cloud [28, 29, 32] generation. Their ability to accurately generate diverse high-quality samples has marked them as a promising tool in various fields.

3. PoseDiffusion

Problem setting. We consider the problem of estimating intrinsic and extrinsic camera parameters given corresponding images of a single scene (*e.g.* frames from an object-centric video, or free-form pictures of a scene).

Formally, given a tuple $\mathtt{I} = (I^i)_{i=1}^N$ of $N \in \mathbb{N}$ input images $I^i \in \mathbb{R}^{3\times H\times W}$, we seek to recover the tuple $x = (x^i)_{i=1}^N$ of corresponding camera parameters $x^i = (K^i, g^i)$ consisting of intrinsics $K^i \subset \mathbb{R}^{3\times 3}$ and extrinsics $g^i \in \mathbb{SE}(3)$ respectively. We defer the details of the camera parametrization to Sec. 3.4.

Extrinsics g^i map a 3D point $\mathbf{p}_w \in \mathbb{R}^3$ from world coordinates to a 3D point $\mathbf{p}_c \in \mathbb{R}^3 = g^i(\mathbf{p}_w)$ in camera coordinates. Intrinsics K^i then perspectively project $\mathbf{p}_c$ to a 2D point $\mathbf{p}_s \in \mathbb{R}^2$ in the screen coordinates with $K^i\mathbf{p}_c \sim \lambda[\mathbf{p}_s; 1], \lambda \in R$ where "$\sim$" indicates homogeneous equivalence.

3.1. Preliminaries of Diffusion models

Diffusion models [15, 47, 48] are a class of likelihood-based models. They model a complex data distribution by learning to invert a diffusion process from data to a simple distribution, usually by means of noising and denoising. The noising process gradually converts the data sample x into noise by a sequence of $T \in \mathbb{N}$ steps. The model is then trained to learn the *de*noising process.

A Denoising Diffusion Probabilistic Model (DDPM) specifically defines the noising process to be Gaussian. Given a variance schedule $\beta_1, ..., \beta_T$ of T steps, the noising transitions are defined as follows:

$$q(x_t \mid x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbb{I}), \tag{1}$$

where $\mathbb{I}$ is the identity matrix. The variance schedule is set so that x_T follows an isotropic Gaussian distribution, *i.e.*, $q(x_T) \approx \mathcal{N}(\mathbf{0}, \mathbb{I})$. Define $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$, then a closed-form solution [15] exists to directly sample x_t given a datum x_0:

$$x_t \sim q(x_t \mid x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbb{I}). \tag{2}$$

The reverse $p_\theta(x_{t-1}|x_t)$ is still Gaussian if β_t is small enough. Therefore, it can be approximated by a model $\mathcal{D}_\theta$:

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}(x_{t-1}; \sqrt{\alpha_t}\mathcal{D}_\theta(x_t, t), (1-\alpha_t)\mathbb{I}). \tag{3}$$

3.2. Diffusion-aided Bundle Adjustment

PoseDiffusion models the conditional probability distribution $p(x|\mathtt{I})$ of the samples x (*i.e.* camera parameters) given the images $\mathtt{I}$. Following the diffusion framework [47] (discussed above), we model $p(x|\mathtt{I})$ by means of the denoising process. More specifically, $p(x|\mathtt{I})$ is first estimated by training a diffusion model $\mathcal{D}_\theta$ on a large training set $\mathcal{T} = \{(x_j, \mathtt{I}_j)\}_{j=1}^S$ of $S \in \mathbb{N}$ scenes with ground truth image batches $\mathtt{I}_j$ and their camera parameters x_j. At inference time, for a new set of observed images $\mathtt{I}$, we sample $p(x|\mathtt{I})$ in order to estimate the corresponding camera parameters x. Note that, unlike for the noising process (Eq. (1)) which is independent of $\mathtt{I}$, the denoising process is conditioned on the input image set $\mathtt{I}$, *i.e.*, $p_\theta(x_{t-1} \mid x_t, \mathtt{I})$:

$$p_\theta(x_{t-1}|x_t, \mathtt{I}) = \mathcal{N}(x_{t-1}; \sqrt{\alpha_t}\mathcal{D}_\theta(x_t, t, \mathtt{I}), (1-\alpha_t)\mathbb{I}). \tag{4}$$

Denoiser $\mathcal{D}_\theta$. We implement the denoiser $\mathcal{D}_\theta$ as a Transformer Trans [52]:

$$\mathcal{D}_\theta(x_t, t, \mathtt{I}) = \text{Trans}\left[\left(\text{cat}(x_t^i, t, \psi(I^i)\right)_{i=1}^N\right] = \mu_{t-1}. \tag{5}$$

Here, Trans accepts a sequence of noisy pose tuples x_t^i, diffusion time t, and feature embeddings $\psi(I^i) \in \mathbb{R}^{D_\psi}$ of the input images I^i. The denoiser outputs the tuple of corresponding denoised camera parameters $\mu_{t-1} = (\mu_{t-1}^i)_{i=1}^N$. Feature embeddings come from a vision transformer model initialized with weights of pre-trained DINO [6].

At train time, $\mathcal{D}_\theta$ is supervised with the denoising loss:

$$\mathcal{L}_{\text{diff}} = E_{t\sim[1,T], x_t\sim q(x_t|x_0,\mathtt{I})} \|\mathcal{D}_\theta(x_t, t, \mathtt{I}) - x_0\|^2, \tag{6}$$

where the expectation aggregates over all diffusion timesteps t, the corresponding diffused samples $x_t \sim q(x_t|x_0, \mathtt{I})$, and a training set $\mathcal{T} = \{(x_{0,j}, \mathtt{I}_j)\}_{j=1}^S$ of $S \in \mathbb{N}$ scenes with images $\mathtt{I}_j$ and their cameras $x_{0,j}$.

Solving Bundle Adjustment by Sampling p_θ. The trained denoiser $\mathcal{D}_\theta$ (Eq. (6)) is later leveraged to sample $p_\theta(x|\mathtt{I})$ which effectively solves our task of inferring camera parameters x given input images $\mathtt{I}$. Note that we assume $p(x|\mathtt{I})$ forms a near-delta distribution and, hence, any sample from $p(x|\mathtt{I})$ will yield a valid pose. Such mild assumption allows to avoid a maximum-aposteriori probability (MAP) estimate of $p(x|\mathtt{I})$.

In more detail, following DDPM sampling [15], we start from random cameras $x_T \sim \mathcal{N}(\mathbf{0}, \mathbb{I})$ and, in each iteration $t \in (T, ..., 0)$, the next step x_{t-1} is sampled from

$$x_{t-1} \sim \mathcal{N}(x_{t-1}; \sqrt{\bar\alpha_{t-1}}\mathcal{D}_\theta(x_t, t, \mathtt{I}), (1-\bar\alpha_{t-1})\mathbb{I}). \tag{7}$$

3.3. Geometry-Guided sampling

So far, our feed-forward network maps images directly to the space of camera parameters. Since deep networks are notoriously bad at regressing precise quantities, such as camera translation vectors or angles of rotation matrices [19], we significantly increase the accuracy of PoseDiffusion by leveraging two-view geometry constraints which form the backbone of state-of-the-art SfM methods.

To this end, we extract reliable 2D correspondences between scene images and guide DDPM sampling iterations (Eq. (7)) so that the estimated poses satisfy the correspondence-induced two-view epipolar constraints.

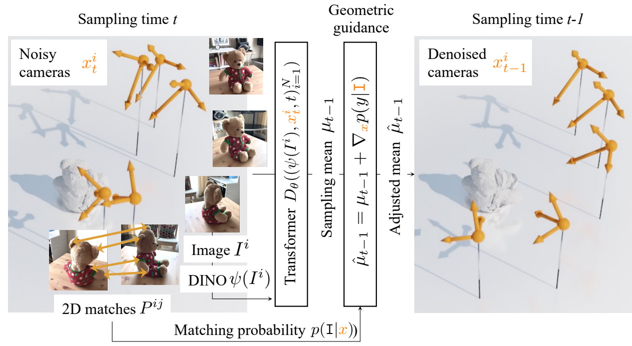

Figure 3: **Inference.** For each step t, Geometry-Guided Sampling (GGS) samples the distribution $p_\theta(x_{t-1} \mid x_t, \mathtt{I}, t)$ of refined cameras x_{t-1} conditioned on input images $\mathtt{I}$ and the previous estimate x_t, while being guided by the gradient of the Sampson matching density $p(\mathtt{I}|x)$.

Sampson Epipolar Error. Specifically, let $P^{ij} = \{(\mathbf{p}_k^i, \mathbf{p}_k^j)\}_{k=1}^{N_{P^{ij}}}$ denote a set of 2D correspondences between image points $\mathbf{p}_k \in \mathbb{R}^2$ for a pair of scene images (I^i, I^j), and denote (x^i, x^j) the corresponding camera poses. Given the latter, we evaluate the compatibility between the cameras and the 2D correspondences via a robust version of Sampson Epipolar Error $e^{ij} \in \mathbb{R}$ [12]:

$$e^{ij}(x^i, x^j, P^{ij}) = \sum_{k=1}^{|P^{ij}|}\left[\frac{\tilde{\mathbf{p}}_k^{j\top} F^{ij}\tilde{\mathbf{p}}_k^i}{(F^{ij}\tilde{\mathbf{p}}_k^i)_1^2 + (F^{ij}\tilde{\mathbf{p}}_k^i)_2^2 + (F^{ij\top}\tilde{\mathbf{p}}_k^j)_1^2 + (F^{ij\top}\tilde{\mathbf{p}}_k^j)_2^2}\right]_\epsilon,$$

where $\tilde{\mathbf{p}} = [\mathbf{p}; 1]$ denotes $\mathbf{p}$ in homogeneous coordinates, $[z]_\epsilon = \min(z, \epsilon)$ is a robust clamping function, $(\mathbf{z})_l$ retrieves l-th element of a vector $\mathbf{z}$, and $F^{ij} \in \mathbb{R}^{3\times3}$ is the Fundamental Matrix [12] mapping points $\mathbf{p}_k^i$ from image I^i to lines in image I^j and vice-versa. Directly optimizing the epipolar constraint $\tilde{\mathbf{p}}_k^{j\top} F^{ij}\tilde{\mathbf{p}}_k^i$ usually provides sub-optimal results [12], which is also observed in our experiments.

Sampson-guided sampling. We follow the classifier diffusion guidance [9] to guide the sampling towards a solution which minimizes the Sampson Epipolar Error and, as such, satisfies the image-to-image epipolar constraint.

In each sampling iteration, classifier guidance perturbs the predicted mean $\mu_{t-1} = \mathcal{D}_\theta(x_t, t, \mathtt{I})$ with a gradient of x_t-conditioned guidance distribution $p(\mathtt{I}|x_t)$:

$$\hat{\mathcal{D}}_\theta(x_t, t, \mathtt{I}) = \mathcal{D}_\theta(x_t, t, \mathtt{I}) + s\nabla_{x_t}\log p(\mathtt{I}|x_t), \tag{8}$$

where $s \in \mathbb{R}$ controls the strength of the guidance. $\hat{\mathcal{D}}_\theta(x_t, t, \mathtt{I})$ then replaces $\mathcal{D}_\theta(x_t, t, \mathtt{I})$ in Eqs. (4) and (7).

Assuming a uniform prior over cameras x allows modeling $p(\mathtt{I}|x_t)$ from Eq. (8) as a product of independent exponential distributions over the pairwise Sampson Errors e^{ij}:

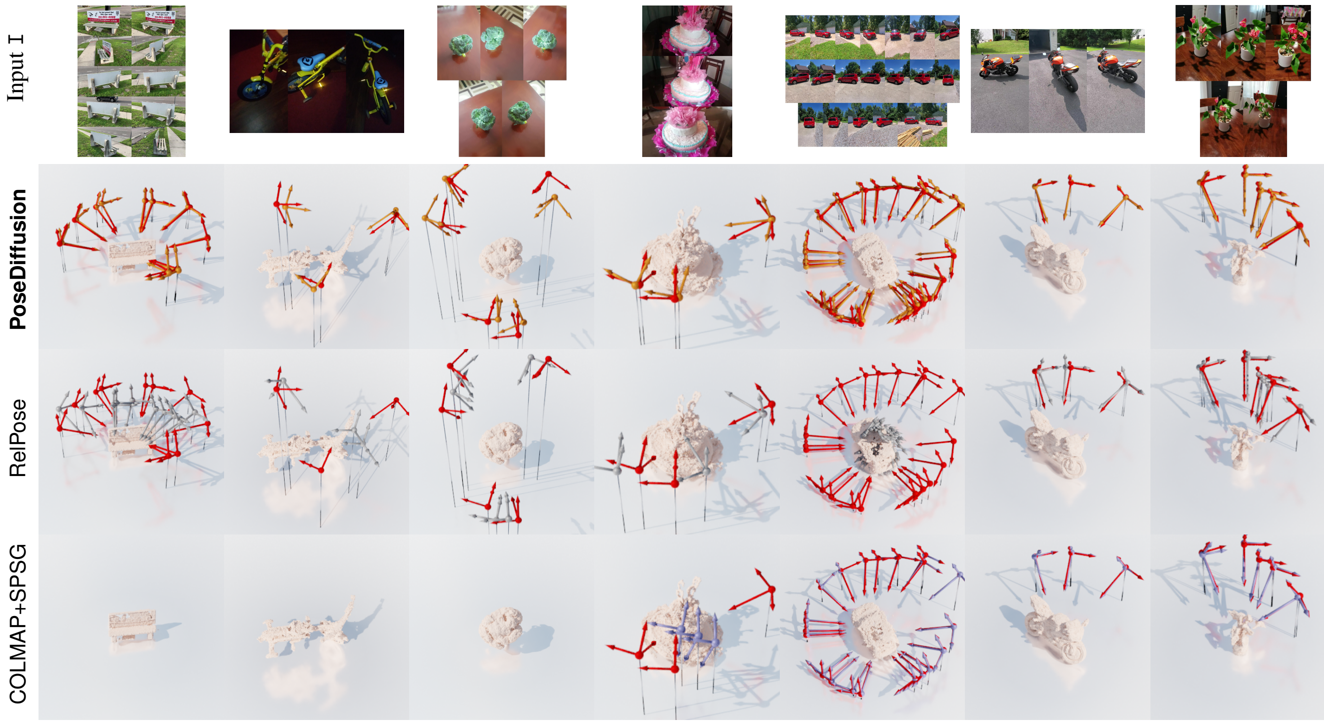

Figure 4: **Pose estimation on CO3Dv2.** Estimated cameras given input images $\mathtt{I}$ (first row). Our PoseDiffusion (2nd row) is compared to RelPose (3rd row), COLMAP+SPSG (4th row), and the ground truth. Missing cameras indicate failure.

$$p(\mathtt{I}|x_t) = \prod_{i,j} p(I^i, I^j | x_t^i, x_t^j) \propto \prod_{i,j} \exp(-e^{ij}). \quad (9)$$

Note that our choice of $p(\mathtt{I}|x_t)$ is meaningful since its mode is attained when Sampson Errors between all image pairs is 0 (*i.e.* all epipolar constraints are satisfied).

3.4. Method details

Representation details. We represent the extrinsics $g^i = (\mathbf{q}^i, \mathbf{t}^i)$ as a 2-tuple comprising the quaternion $\mathbf{q}^i \in \mathbb{H}$ of the rotation matrix $R^i \in \mathbb{SO}(3)$ and the camera translation vector $\mathbf{t}^i \in \mathbb{R}^3$. As such, $g^i(\mathbf{p}_w)$ represents a linear world-to-camera transformation $\mathbf{p}_c = g^i(\mathbf{p}_w) = R^i\mathbf{p}_w + \mathbf{t}^i$. We use a camera calibration matrix $K^i = \left[f^i, 0, p_x; 0, f^i, p_y; 0, 0, 1\right] \in \mathbb{R}^{3\times 3}$, with one degree of freedom defined by the focal length $f^i \in \mathbb{R}^+$. Following common practice in SfM [42, 43], the principal point coordinates $p_x, p_y \in \mathbb{R}$ are fixed to the center of the image. To ensure strictly positive focal length f^i, we represent it as $f^i = \exp(\hat{f}^i)$, where $\hat{f}^i \in \mathbb{R}$ is the quantity predicted by the denoiser $\mathcal{D}_\theta$. Therefore, the transformer Trans (Eq. (5)) outputs a tuple of raw predictions $\left((\hat{f}^i, \mathbf{q}^i, \mathbf{t}^i)\right)_{i=1}^N$ which is converted (in close-form) to a tuple of cameras $x = \left((K^i, g^i)\right)_{i=1}^N$.

Tackling Coordinate Frame Ambiguity. Because our training set $\mathcal{T}$ is constructed by SfM reconstructions [42], the training poses $\{\hat{g}_j\}_{j=1}^S$ are defined up to an arbitrary scene-specific similarity transformation. To prevent overfitting to the scene-specific training coordinate frames, we canonicalize the input before passing to the denoiser: we normalize the extrinsics $g_j = (\hat{g}_j^1, \ldots \hat{g}_j^N)$ as relative camera poses to a randomly selected pivot camera $\hat{g}_j^\star$. We inform the denoiser about the pivot camera by appending a binary flag $p^i_{\text{pivot}} \in \{0, 1\}$ to the image features $\psi(I^i)$ (Eq. (5)). Furthermore, in order to canonicalize the scale, we divide the input camera translations by the median of the norms of the pivot-normalized translations.

4. Experiments

We experiment on two real-world datasets, ablate the design choices of the model, and compare with prior work.

Datasets. We consider two datasets with different statistics. The first is **CO3Dv2** [38] containing roughly 37k turntable-like videos of objects from 51 MS-COCO categories [23]. The dataset provides cameras automatically annotated by COLMAP [44] using 200 frames in each video. Secondly, we evaluate on **RealEstate10k** [62] which comprises 80k YouTube clips capturing the interior and exterior of real estate. Its camera annotations were auto-generated with ORB-SLAM 2 [34] and refined with bundle adjust-

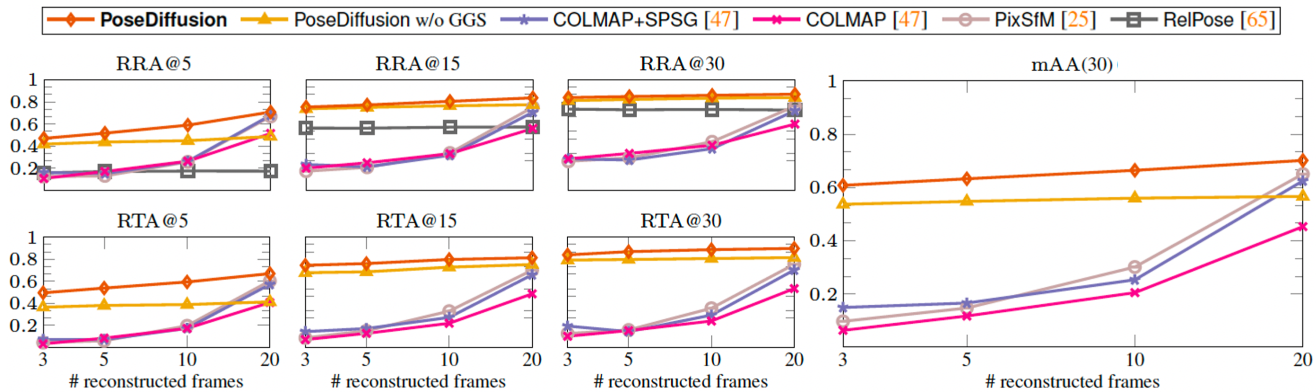

Figure 5: **Pose estimation accuracy on CO3Dv2.** Metrics $\mathrm{RRA@}\tau, \mathrm{RTA@}\tau$ at different thresholds τ and $\mathrm{mAA}(30)$ (y-axes, higher-better) as a function of the number of input frames (x-axes). RelPose does not predict camera translation and hence is omitted in the respective figures.

ment. We use the same training set as in [54], *i.e.* 57k training scenes and, as some baselines are time-consuming, a random smaller 1.8k-video subset of the original 7K test videos. Naturally, we always test on unseen videos.

Baselines and comparisons. We chose COLMAP [44], one of the most popular SfM pipelines, as a dense-pose estimation baseline. Besides the classic version leveraging RANSAC-matched SIFT features, we also benchmark COLMAP+SPSG which builds on SuperPoints [8] matched with SuperGlue [40]. PixSfM [24] further improves accuracy by directly aligning deep features. We also compare to RelPose [60] which is the current State of the Art in sparse pose estimation. Finally, to ablate Geometry Guided Sampling (GGS - Eq. (9)), PoseDiffusion w/o GGS leverages our denoiser without GGS.

Training. We train the denoiser $\mathcal{D}_\theta$ using the Adam optimizer with the initial learning rate of 0.0005 until convergence of $\mathcal{L}_{\text{diff}}$ - learning rate is decayed ten-fold after 30 epochs. The latter takes two days on 8 GPUs. In each training batch, we randomly sample between 3-20 frames and their cameras from a random scene of the training dataset.

Geometry-guided sampling. PoseDiffusion's GGS leverages the SuperPoint features [8] matched with SuperGlue [40], where the Sampson error is clamped at $\epsilon = 10$ (Sec. 3.3). To avoid spurious local minima, we apply GGS to the last 10 diffusion sampling steps. During each step t, we adjust the sampling mean by running 100 GGS iterations. We observed improved sampling stability when the guidance strength s (Eq. (8)) is set adaptively so that the norm of the guidance gradient $\nabla p(\mathtt{I}|x)$ does not exceed a multiple $\alpha\|\mu_t\|$ ($\alpha = 0.0001$) of the current mean's norm.

Evaluation metrics. We compute the **Relative Rotation Accuracy** (RRA) to compare the relative rotation $R_i R_j^\top$ from i-th to j-th camera to the ground truth $R_i^\star R_j^{\star\top}$. Similarly, the **Relative Translation Accuracy** $\mathrm{RTA}(\mathbf{t}_{ij}, \mathbf{t}_{ij}^\star) = \arccos(\mathbf{t}_{ij}^\top \mathbf{t}_{ij}^\star / (\|\mathbf{t}_{ij}\| \|\mathbf{t}_{ij}^\star\|))$ evaluates the angle between the predicted and ground-truth vector $\mathbf{t}_{ij}$ / $\mathbf{t}_{ij}^\star$ pointing from camera i to j. RRA/RTA are invariant to the absolute coordinate frame ambiguity. For a given threshold τ, we report $\mathrm{RTA@}\tau/\mathrm{RRA@}\tau$ ($\tau \in \{5, 15, 30\}$), *i.e.* the percentage of camera pairs with RRA/RTA below a threshold τ.

Additionally, following the Image Matching Benchmark [17], we report **mean Average Accuracy** (mAA) (also known as Area under Curve - AUC). Specifically, mAA calculates the area under the curve recording the accuracies of the angular differences between the ground-truth and predicted cameras for a range of angular accuracy thresholds. For an image pair, mAA defines the accuracy at a threshold τ as $min(\mathrm{RRA@}\tau, \mathrm{RTA@}\tau)$. Following RelPose's [60] upper angular threshold of 30°, we report $\mathrm{mAA}(30)$ which is integrated over $\tau \in [1, 30]$.

4.1. Camera pose estimation

Object-centric pose. We first compare on CO3Dv2 where each scene comprises frames capturing a single object from a variety of viewpoints with approximately constant distance from the object. Fig. 5 contains quantitative results while Fig. 4 illustrates example camera estimates. PoseDiffusion significantly improves over all baselines in all metrics in both the sparse and dense setting. Note that, here ground truth cameras were obtained with COLMAP itself (but using 200 frames), likely favouring COLMAP reconstructions. Importantly, removing GGS (PoseDiffusion w/o GGS) leads to a drop in performance for tighter accuracy thresholds across all metrics. This clearly demonstrates that GGS facilitates accurate camera estimates. The latter also validates the accuracy of our intrinsics since they are an important component of GGS.

Scene-centric pose. Here, we reconstruct camera poses in free-form in/outdoor scenes of RealEstate10k which, historically, has been the domain of traditional SfM methods. We

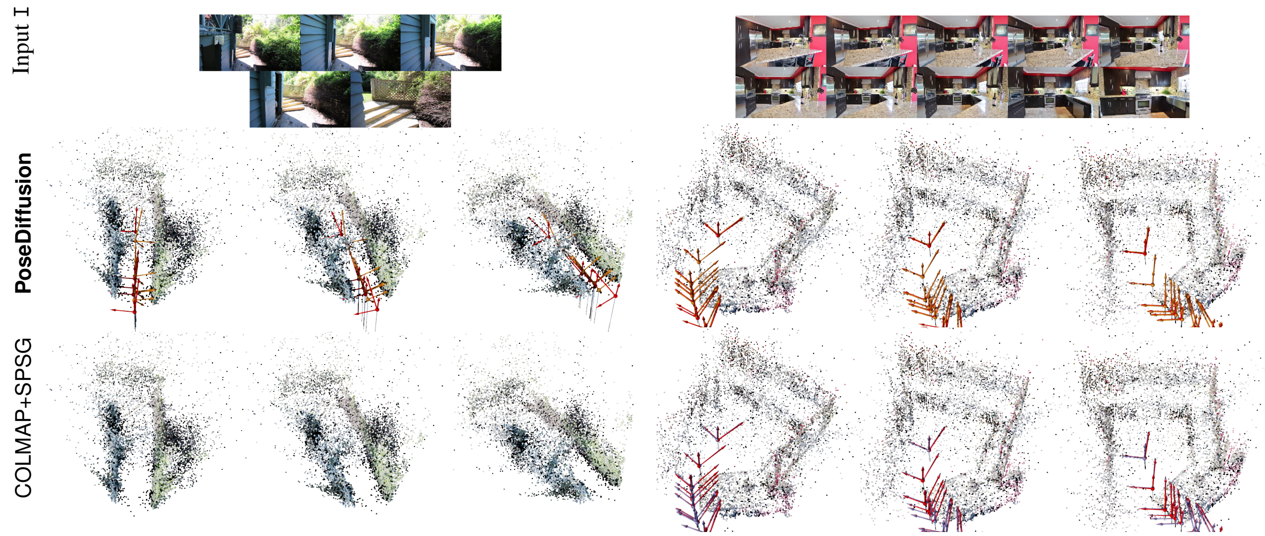

Figure 6: **Pose estimation on RealEstate10k** visualizing the cameras estimated given input images I (first row). Our PoseDiffusion (2nd row) is compared to COLMAP+SPSG (3rd row), and the ground truth. Missing cameras indicate failure. For better visualization, we display each scene from three different viewpoints.

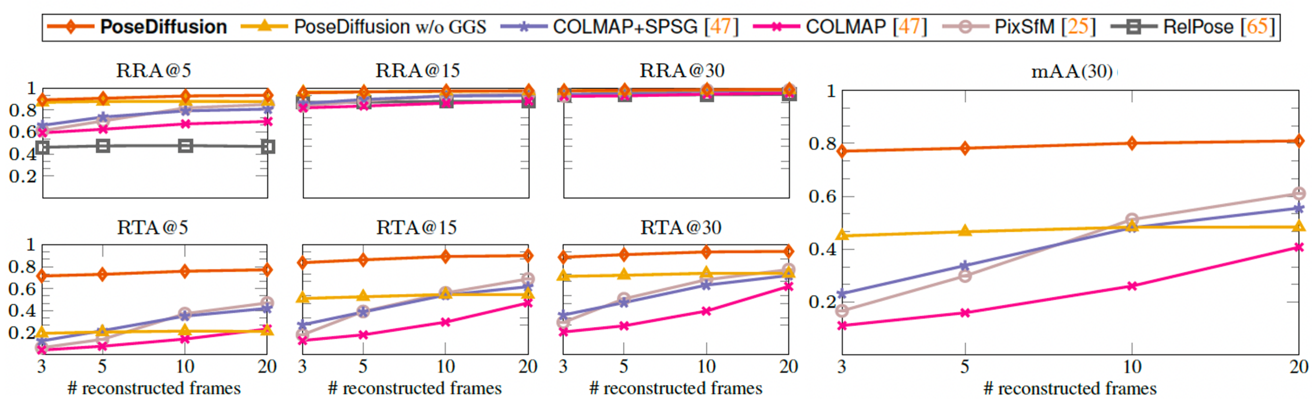

Figure 7: **Pose estimation on RealEstate10k.** Metrics $\mathrm{RRA@}\tau, \mathrm{RTA@}\tau$ at different thresholds τ and $\mathrm{mAA}(30)$ (y-axes, higher-better) as a function of the number of input frames (x-axes).

evaluate quantitatively in Fig. 7 and qualitatively in Fig. 6. PoseDiffusion significantly outperforms all baselines in all metrics. Here, the comparison to COLMAP is fairer than on CO3Dv2, as RealEstate10k used ORB-SLAM2 [35] to obtain the ground-truth cameras.

Importance of diffusion. To validate the effect of the diffusion model, we also provide the PoseReg baseline, which uses the same architecture and training hyper-parameters as our method but directly regresses poses. PoseReg shows clearly lower performance (cf. Tab. 1). Moreover, without the iterative refinement of our diffusion model, the gain of applying GGS to PoseReg (PoseReg+GGS) is limited.

Generalization. We also evaluate the ability of different methods to generalize to different data distributions. First, following RelPose [60], we train on a set of 41 training categories from CO3Dv2, and evaluate the remaining 10 held-out categories. As shown in Tab. 2, our method outperforms all baselines indicating superior generalizability, even without the help of GGS.

Moreover, we evaluate a significantly more difficult scenario: transfer from the CO3Dv2 to RealEstate10k. This setting brings a considerable difficulty: CO3Dv2 predominantly contains indoor objects with circular fly-around trajectories while RealEstate10k comprises outdoor scenes and linear fly-through camera motion (see Figs. 4 and 6). Surprisingly, our results are still comparable to PixSfM, while better than COLMAP and RelPose.

4.2. Novel-view synthesis.

To evaluate the quality of the camera pose prediction for downstream tasks, we train NeRF models using predicted camera parameters and measure the RGB reconstruction er-

Metric	RelPose	COLMAP +SPSG	PixSfM	PoseReg	Ours w/o GGS	PoseReg +GGS	Ours
RRA@15	57.1	31.6	33.7	53.2	75.9	57.0	**80.5**
RTA@15	-	27.3	32.9	49.1	72.8	53.4	**79.8**
mAA(30)	-	25.3	30.1	45.0	56.0	48.2	**66.5**

Table 1: **Pose regression ablation** comparing a diffusion-free pose regressor PoseReg (with/without GGS) to our PoseDiffusion on CO3Dv2 with 10 input frames (**Bold** denotes the top result and an underline signifies the second best).

Test Set	COLMAP	COLMAP+SPSG	PixSfM	Ours w/o GGS	Ours
CO3Dv2 Unseen	25.8	30.3	34.2	40.1	**50.8**
RealEstate10k	26.1	45.2	**49.4**	18.7	48.0

Table 2: **Generalization** reporting $\mathrm{mAA}(30)$ for 10 input frames. We first train on 41 CO3Dv2 seen categories. Testing is conducted on 11 unseen categories (top row), and on RealEstate10k (bottom) (**Bold** denotes the top result and an underline signifies the second best).

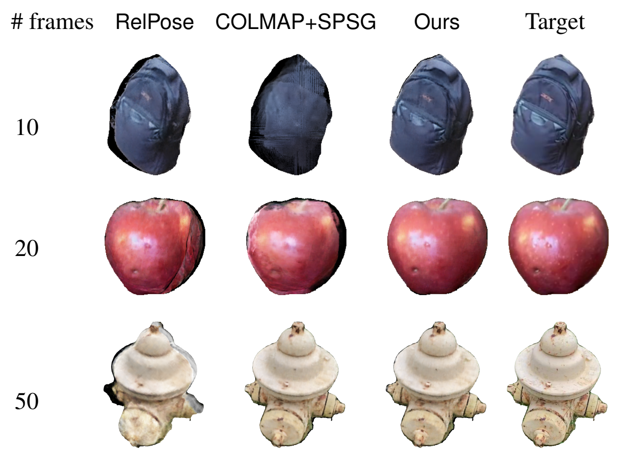

Figure 8: **Synthesized novel views.** NeRF trained with camera poses estimated by various methods. This metric is more fair as it does not rely on GT pose annotations by another method.

ror in novel views. Note that, as opposed to the camera pose evaluation on CO3Dv2, here, we fairly evaluate against unbiased image ground truth. We generate a dataset of 10, 20, and 50 frames for 50 random sequences of CO3Dv2. Each sequence contains 4 validation frames with the remaining ones used to train the NeRF. We report PSNR averaged over all validation frames as an indirect measure of camera pose accuracy. Furthermore, the experiment also evaluates the accuracy of the predicted intrinsics (focal lengths) since these are an inherent part of the NeRF's camera model significantly affecting the rendering quality.

In Tab. 3, our method outperforms COLMAP+SPSG, demonstrating the better suitability of our predicted cameras for NVS. Moreover, Ours + GT Focal Length, which

Method	# frames 10	20	50
RelPose [60]*	21.33	23.12	25.09
Ours + GT Focal Length	24.72	26.58	28.61
COLMAP+SPSG	15.78	25.17	**28.66**
Ours	**24.37**	**26.96**	28.53

Table 3: **Novel View Synthesis.** PSNR for NeRFs [33] trained on CO3Dv2 using cameras estimated by various methods. RelPose * does not predict translation vectors and focal lengths, and uses the ground truth here instead.

replaces the predicted focal lengths with the ground truth, is perfectly on par with Ours, signifying the reliability of our intrinsics. Fig. 8 provides the qualitative comparison.

Execution time. Our method without GGS typically takes around 1 second for inference on a sequence of 20 frames, and enabling GGS increases the execution time to 60-90 seconds. GGS is currently unoptimized (a simple *for* loop in Python), compared to common C++ implementations for SfM methods which can be adopted here.

5. Conclusion

This paper presents PoseDiffusion, a learned camera estimator enjoying both the power of traditional epipolar geometry constraint and diffusion model. We show how the diffusion framework is ideally compatible with the task of camera parameter estimation. The iterative nature of this classical task is mirrored in the denoising diffusion formulation. Additionally, point-matching constraints between image pairs can be used to guide the model and refine the final prediction. In our experiments, we improve over traditional SfM methods such as COLMAP, as well as the learned approaches. We are able to show improvements regarding the pose prediction accuracy as well as on the novel-view synthesis task, which is one of the most popular current applications of COLMAP. Finally, we are able to demonstrate that our method can overcome one of the main limitations of learned methods: generalization across datasets, even when trained on a dataset with different pose distributions.

Acknowledgements.

We would like to thank Nikita Karaev, Luke Melas-Kyriazi, and Shangzhe Wu for their insightful discussions. We appreciate the great help from Jason Y. Zhang for sharing the baseline/benchmark code. Jianyuan Wang is supported by Facebook Research. Christian Rupprecht is supported by ERC-CoG UNION 101001212 and VisualAI EP/T028572/1.

References

[1] Sameer Agarwal, Keir Mierle, and The Ceres Solver Team. Ceres Solver, 3 2022. 2

[2] Sameer Agarwal, Noah Snavely, Ian Simon, Steven M. Seitz, and Richard Szeliski. Building Rome in a day. In *Proc. ICCV*, 2009. 1, 2

[3] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-Up Robust Features (SURF). *CVIU*, 110(3), 2008. 2

[4] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. Dsac-differentiable ransac for camera localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6684–6692, 2017. 2

[5] Eric Brachmann and Carsten Rother. Neural-guided ransac: Learning where to sample model hypotheses. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4322–4331, 2019. 2

[6] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 4

[7] Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun. Robust reconstruction of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5556–5565, 2015. 2

[8] D DeTone, T Malisiewicz, and A'Superpoint Rabinovich. Self-supervised interest point detection and description'. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops.(*, 2018. 2, 6

[9] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021. 2, 4

[10] 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*, 24(6), 1981. 2

[11] Yasutaka Furukawa, Brian Curless, Steven M. Seitz, and Richard Szeliski. Towards Internet-scale multi-view stereo. In *Proc. CVPR*. IEEE, 2010. 2

[12] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. 2, 4

[13] Richard I Hartley. In defense of the eight-point algorithm. *IEEE Transactions on pattern analysis and machine intelligence*, 19(6):580–593, 1997. 2

[14] Jared Heinly, Johannes L. Schonberger, Enrique Dunn, and Jan-Michael Frahm. Reconstructing the world* in six days *(as captured by the yahoo 100 million image dataset). In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 2

[15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 2, 3, 4

[16] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv preprint arXiv:2204.03458*, 2022. 3

[17] 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. *International Journal of Computer Vision*, 129(2):517–547, 2021. 2, 6

[18] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In *Proceedings of the IEEE international conference on computer vision*, pages 1521–1529, 2017. 3

[19] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pages 2938–2946, 2015. 4

[20] Erwin Kruppa. *Zur Ermittlung eines Objektes aus zwei Perspektiven mit innerer Orientierung*. 1913. 1

[21] Hongdong Li and Richard Hartley. Five-point motion estimation made easy. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 1, pages 630–633. IEEE, 2006. 2

[22] Amy Lin, Jason Y Zhang, Deva Ramanan, and Shubham Tulsiani. Relpose++: Recovering 6d poses from sparse-view observations. *arXiv preprint arXiv:2305.04926*, 2023. 3

[23] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *Proc. ECCV*, 2014. 5

[24] Philipp Lindenberger, Paul-Edouard Sarlin, Viktor Larsson, and Marc Pollefeys. Pixel-perfect structure-from-motion with featuremetric refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5987–5997, 2021. 2, 6

[25] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural Volumes: Learning Dynamic Renderable Volumes from Images. *ACM Trans. Graph.*, 38(4):65:1–65:14, July 2019. Place: New York, NY, USA Publisher: ACM. 2

[26] David G. Lowe. Object Recognition from Local Scale-Invariant Features. In *Proc. ICCV*, 1999. 2

[27] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 60(2), 2004. 2

[28] Shitong Luo and Wei Hu. Diffusion Probabilistic Models for 3D Point Cloud Generation, 2021. 2, 3

[29] Zhaoyang Lyu, Zhifeng Kong, Xudong Xu, Liang Pan, and Dahua Lin. A conditional point diffusion-refinement paradigm for 3d point cloud completion. *arXiv preprint arXiv:2112.03530*, 2021. 3

[30] Wei-Chiu Ma, Anqi Joyce Yang, Shenlong Wang, Raquel Urtasun, and Antonio Torralba. Virtual correspondence: Humans as a cue for extreme-view geometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15924–15934, 2022. 3

[31] Runyu Mao, Chen Bai, Yatong An, Fengqing Zhu, and Cheng Lu. 3dg-stfm: 3d geometric guided student-teacher

feature matching. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVIII*, pages 125–142. Springer, 2022. 2

[32] Luke Melas-Kyriazi, Christian Rupprecht, and Andrea Vedaldi. Pc2: Projection-conditioned point cloud diffusion for single-image 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12923–12932, June 2023. 3

[33] 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. *Proc. ECCV*, 2020. 2, 8

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

[35] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015. 7

[36] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE transactions on pattern analysis and machine intelligence*, 26(6):756–770, 2004. 2

[37] Onur Özyeşil, Vladislav Voroninski, Ronen Basri, and Amit Singer. A survey of structure from motion*. *Acta Numerica*, 26:305–364, 2017. 2

[38] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10901–10911, 2021. 2, 5

[39] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12716–12725, 2019. 2

[40] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020. 2, 6

[41] Frederik Schaffalitzky and Andrew Zisserman. Multi-view Matching for Unordered Image Sets, or "How Do I Organize My Holiday Snaps?". In *Proc. ECCV*, 2002. 2

[42] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 5

[43] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. 2, 5

[44] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-Motion Revisited. In *Proc. CVPR*, 2016. 2, 5, 6

[45] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022. 2, 3

[46] Samarth Sinha, Jason Y Zhang, Andrea Tagliasacchi, Igor Gilitschenski, and David B Lindell. Sparsepose: Sparse-view camera pose regression and refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21349–21359, 2023. 3

[47] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015. 3

[48] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019. 3

[49] Chengzhou Tang and Ping Tan. Ba-net: Dense bundle adjustment network. *arXiv preprint arXiv:1806.04807*, 2018. 2

[50] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle Adjustment - A Modern Synthesis. In *Proc. ICCV Workshop*, 2000. 2

[51] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. Demon: Depth and motion network for learning monocular stereo. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5038–5047, 2017. 2

[52] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Proc. NeurIPS*, 2017. 4

[53] Jianyuan Wang, Yiran Zhong, Yuchao Dai, Stan Birchfield, Kaihao Zhang, Nikolai Smolyanskiy, and Hongdong Li. Deep two-view structure-from-motion revisited. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8953–8962, June 2021. 2

[54] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. In *Proc. CVPR*, pages 7467–7477, 2020. 6

[55] Shangzhe Wu, Ruining Li, Tomas Jakab, Christian Rupprecht, and Andrea Vedaldi. MagicPony: Learning articulated 3d animals in the wild. 2023. 3

[56] Shangzhe Wu, Christian Rupprecht, and Andrea Vedaldi. Unsupervised learning of probably symmetric deformable 3d objects from images in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1–10, 2020. 3

[57] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017. 3

[58] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. LIFT: Learned Invariant Feature Transform. In *Proc. ECCV*, 2016. 2

[59] Jiahui Zhang, Dawei Sun, Zixin Luo, Anbang Yao, Lei Zhou, Tianwei Shen, Yurong Chen, Long Quan, and Hongen

Liao. Learning two-view correspondences and geometry using order-aware network. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5845–5854, 2019. 2

[60] Jason Y Zhang, Deva Ramanan, and Shubham Tulsiani. Relpose: Predicting probabilistic relative rotation for single objects in the wild. In *ECCV*, pages 592–611. Springer, 2022. 2, 3, 6, 7, 8

[61] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1851–1858, 2017. 2

[62] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018. 2, 5