Monte Carlo Diffusion for Generalizable Learning-Based RANSAC

Jiale Wang[1*], **Chen Zhao**[2*], **Wei Ke**[1†], **Tong Zhang**[2,3]

[1]Xi'an Jiaotong University, China
[2]EPFL, Switzerland
[3]University of Chinese Academy of Sciences, China

Abstract

Random Sample Consensus (RANSAC) is a fundamental approach for robustly estimating parametric models from noisy data. Existing learning-based RANSAC methods utilize deep learning to enhance the robustness of RANSAC against outliers. However, these approaches are trained and tested on the data generated by the same algorithms, leading to limited generalization to out-of-distribution data during inference. Therefore, in this paper, we introduce a novel diffusion-based paradigm that progressively injects noise into ground-truth data, simulating the noisy conditions for training learning-based RANSAC. To enhance data diversity, we incorporate Monte Carlo sampling into the diffusion paradigm, approximating diverse data distributions by introducing different types of randomness at multiple stages. We evaluate our approach in the context of feature matching through comprehensive experiments on the ScanNet and MegaDepth datasets. The experimental results demonstrate that our Monte Carlo diffusion mechanism significantly improves the generalization ability of learning-based RANSAC. We also develop extensive ablation studies that highlight the effectiveness of key components in our framework.

Code — https://comedy0913.github.io/projects/MCD.html

Introduction

Robust geometric estimation is crucial for 3D computer vision tasks such as structure-from-motion (Snavely, Seitz, and Szeliski 2008), SLAM (Mur-Artal, Montiel, and Tardos 2015), virtual reality (Szeliski 1994), and augmented reality (Yu et al. 2009). The goal is to estimate a reliable geometric transformation model from noisy data with outliers. As the most established robust estimator, RANSAC (Fischler and Bolles 1981) has been extensively studied and widely adopted for decades. RANSAC operates under the assumption of a parametric model, relying on the consistency with this model to distinguish inliers from noisy data. Therefore, RANSAC is applicable to data generated by any method for the same task. This property allows RANSAC to integrate seamlessly with geometric estimation frameworks regardless of the source of the raw data. For instance, in camera pose estimation (Hartley and Zisserman 2003), RANSAC is compatible with various pixel-level correspondences, including those obtained from handcrafted detectors/descriptors (Lowe 2004; Rublee et al. 2011; Bay et al. 2008) and learning-based alternatives (Sarlin et al. 2020; Sun et al. 2021; Edstedt et al. 2024).

The major limitation of RANSAC lies in its sensitivity against outliers. As shown in the literature (Hartley and Zisserman 2003; Kim, Dunn, and Frahm 2017; Zhao et al. 2020), RANSAC becomes less effective when the initial data contains a high proportion of outliers. To handle this issue, some approaches have been proposed to improve the quality of initial data (Bian et al. 2017; Yi et al. 2018; Ma

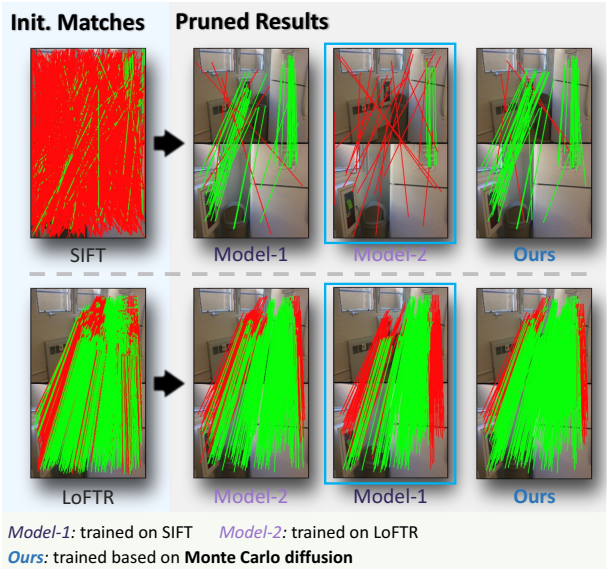

Figure 1: Advantage of Monte Carlo diffusion. Model-1 and Model-2 denote NG-RANSAC trained on SIFT and LoFTR, respectively. The green lines indicate inliers, and the red ones are outliers. The models trained on specific patterns show limited generalization on out-of-distribution data, e.g., Model-2 trained on LoFTR performs poorly when tested on SIFT. In contrast, NG-RANSAC trained based on our Monte Carlo diffusion demonstrates better generalization across different initial matches.

*These authors contributed equally.
†Corresponding author

et al. 2019; Zhang et al. 2019; Zhao et al. 2021). However, outliers are still inevitable, particularly in challenging scenarios. Therefore, some learning-based RANSAC algorithms (Brachmann et al. 2017; Brachmann and Rother 2019; Wei et al. 2023) have been introduced, leveraging deep learning to improve the robustness of RANSAC against outliers. In this paper, we focus on investigating these learning-based RANSACs. These methods demonstrate promising effectiveness when tested on in-distribution data. Specifically, in the context of feature matching (Baumberg 2000), learning-based RANSACs are trained and tested on pixel-level correspondences obtained using the *same* algorithm, such as SIFT (Lowe 2004). These methods exhibit limited generalization when applied to out-of-distribution data. As shown in Fig. 1, the correspondences established via different approaches exhibit significant differences in keypoint positions, spatial patterns, and outlier ratios. A model trained on LoFTR (Sun et al. 2021) struggles when applied to SIFT (Lowe 2004) and vice versa, despite both addressing the same geometric problem. This observation indicates that the development of learning-based RANSAC variants (Brachmann et al. 2017; Brachmann and Rother 2019; Wei et al. 2023) has inadvertently weakened the core strength of RANSAC, limiting its applicability in real-world scenarios where data is typically from diverse algorithms rather than a specific one.

Consequently, we propose a novel training paradigm, incorporating a diffusion-driven module that eliminates dependence on specific data distributions. Our key insight is to decouple the training process from specific data generation approaches by simulating diverse data patterns. Specifically, we progressively inject the noise into the ground-truth data through a diffusion process. To enhance the data diversity, we develop a Monte Carlo sampling mechanism where we introduce different types of randomness at multiple stages of the diffusion module. This stochastic property enables our method to simulate noisy data with diverse distributions. We evaluate the applicability of RANSAC in feature matching through comprehensive experiments on ScanNet (Dai et al. 2017) and MegaDepth (Li and Snavely 2018) datasets that cover diverse indoor and outdoor scenarios, respectively. We utilize SIFT (Lowe 2004) and LoFTR (Sun et al. 2021) to establish pixel-wise correspondences, which represent two types of distributions. Experimental results show that a learning-based RANSAC trained on one distribution fails to generalize to the other during testing, whereas the method trained based on our Monte Carlo diffusion achieves significantly better generalization. Moreover, we conduct ablation studies where the results demonstrate the compatibility of our method with learning-based RANSACs and highlight the effectiveness of key components in our framework. In summary, our primary contributions are threefold:

- We investigate the generalization problem in learning-based RANSAC and identify existing training strategies as the primary factor limiting generalization.
- We propose a diffusion-based mechanism that simulates noisy data independent of specific data generation algorithms.
- We introduce a Monte Carlo sampling module that enhances the data diversity by injecting multiple sources of randomness at different stages of the diffusion process.

Related Work

Random sample consensus. RANASC (Fischler and Bolles 1981) has been widely explored in the literature (Zhao et al. 2020; Hartley and Zisserman 2003), aiming to robustly compute a parametric model from noisy data. The handcrafted RANSAC (Chum, Matas, and Kittler 2003; Raguram et al. 2012; Barath, Matas, and Noskova 2019) excels in improving the robustness of model estimation in scenarios where raw data is of sufficient quality. However, the performance deteriorates when a large proportion of outliers exists. To improve the robustness against outliers, recent advances have combined deep neural networks with RANSAC. For instance, some methods, such as (Kim, Dunn, and Frahm 2017; Zhao et al. 2019, 2021), propose to utilize a network as a pruner to filter out outliers from raw data. Notably, these methods act as independent pruners, separate from RANSAC itself. In this context, some approaches (Brachmann et al. 2017; Brachmann and Rother 2019; Wei et al. 2023) develop learning-based alternatives within the pipeline of RANSAC. These methods achieve promising performance on in-distribution data during inference but struggle to generalize to out-of-distribution data. In this paper, we address this issue by decoupling training from fixed data distributions through a diffusion-driven simulation mechanism.

Pixel-wise feature matching. Given two images, pixel-wise correspondences are established to compute the geometric transformation for downstream tasks such as image alignment (Gao et al. 2013; Brown and Lowe 2007) and 3D reconstruction (Mildenhall et al. 2021; Kerbl et al. 2023). Traditional methods generate correspondences between keypoints based on feature similarities, utilizing handcrafted keypoint detectors and feature descriptors such as SIFT (Lowe 2004) and ORB (Rublee et al. 2011). These methods often produce numerous false matches, i.e., outliers, particularly in textureless regions and under severe viewpoint changes. In contrast, learning-based alternatives such as SuperGlue (Sarlin et al. 2020), LoFTR (Sun et al. 2021), and VGGT (Wang et al. 2025) employ deep neural networks to establish correspondences, leading to advanced matching quality. Since outliers inevitably exist in initial correspondences, RANSAC plays a crucial role in improving the accuracy of model estimation. Moreover, different matching methods produce distinct initial correspondences, characterized by variations in matching density, outlier ratio, and keypoint position. This variability presents significant challenges for learning-based RANSAC (Brachmann and Rother 2019; Wei et al. 2023).

Diffusion paradigm. Diffusion is originally rooted in physics and stochastic processes, describing how systems evolve under random perturbations (Einstein 1905; Stroock and Varadhan 1997). Although diffusion models have recently gained popularity in image generation (Rombach et al. 2022; Zhang, Rao, and Agrawala 2023; Croitoru et al. 2023), their applicability is much broader and not limited to

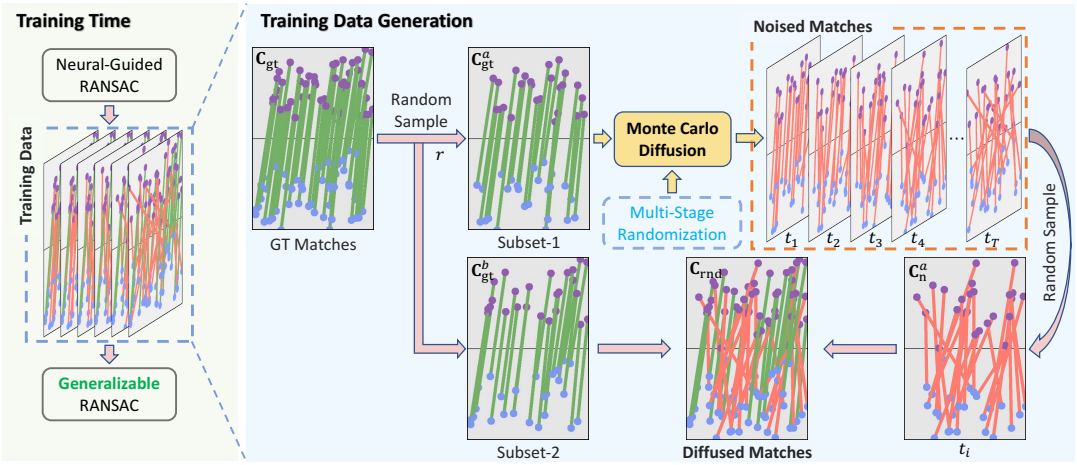

Figure 2: Pipeline of the diffusion process. We leverage diffusion to simulate noisy data for training learning-based RANSAC. Given ground-truth matches $\mathbf{C}_{\text{gt}}$ between two images, we randomly split them into two subsets $\mathbf{C}_{\text{gt}}^a$ and $\mathbf{C}_{\text{gt}}^b$. $\mathbf{C}_{\text{gt}}^a$ is processed by a Monte Carlo diffusion module with multi-stage randomization, generating multiple sets of noised matches at different timesteps. The final diffused matches are formed by combining $\mathbf{C}_{\text{gt}}^b$ as inliers with $\mathbf{C}_n^a$ sampled at timestep t_i as outliers. The learning-based RANSAC is then trained on the resulting diffused matches.

this specific domain. Diffusion serves as a general modeling principle that simulates dynamic, noise-driven transformations in data. For instance, anisotropic diffusion for image processing focuses on forward-only diffusion to smooth images while preserving edges (Perona and Malik 2002); diffusion maps utilize forward diffusion to reveal the intrinsic low-dimensional structure of data for dimensionality reduction and clustering (Coifman and Lafon 2006). These examples highlight the broader interpretation of diffusion beyond image generation. In this broader context, our method achieves a diffusion process by progressively adding noise to correct matches across timesteps, thereby modeling the dynamic change of data.

Method

As illustrated in Fig. 2, our primary contribution lies in employing a diffusion mechanism to progressively transform clean data into noisy variants, with noise intensity increasing over time. Furthermore, we enhance the diversity of the diffused data through Monte Carlo sampling in a multi-stage randomization module. The diffused data points are then used to train a learning-based RANSAC, aiming to robustly estimate the parametric model and identify inliers.

Problem Formulation

Note that we focus on the applicability of RANSAC in feature matching. Let $\mathcal{M} = \{M_1, M_2, \ldots, M_K\}$ represent the collection of all feature matching methods. For an image pair $(\mathbf{I}, \mathbf{I}')$, each matcher $M_k \in \mathcal{M}$ generates a set of correspondences $\mathbf{C}_k = [\mathbf{c}_1^{(k)}, \ldots, \mathbf{c}_N^{(k)}] \in \mathbb{R}^{N\times 4}$, where $\mathbf{c}_i^{(k)} = [x_i, y_i, x_i', y_i']$ indicates a correspondence between a keypoint (x_i, y_i) in $\mathbf{I}$ and a keypoint (x_i', y_i') in $\mathbf{I}'$. Let $\mathcal{D}_{\text{all}} = \bigcup_k \mathcal{D}_{M_k}$ denotes the union of distributions produced by all matchers. The objective of training a learning-based RANSAC is to learn parameters θ^* that minimize the expected loss over $\mathcal{D}_{\text{all}}$:

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{\mathbf{C}\sim\mathcal{D}_{\text{all}}} \left[\mathcal{L}(\mathbf{P}_\theta \mid \mathbf{C}, \mathbf{C}_{\text{gt}})\right], \tag{1}$$

where θ denotes the learnable parameters in learning-based RANSAC; $\mathbf{P}_\theta$ represents the results predicted from input correspondences $\mathbf{C}$; $\mathcal{L}(\cdot)$ denotes the loss function; $\mathbf{C}_{\text{gt}}$ is the ground-truth correspondences.

However, the optimization over $\mathcal{D}_{\text{all}}$ is impractical due to its computational complexity and cost. Instead, existing methods are typically trained on a specific $\mathcal{D}_{M_k}$. They fail to generalize to other $\mathcal{D}_{M_j} \in \mathcal{D}_{\text{all}}$, $j \neq k$, due to differences in distribution. To overcome this issue, a straightforward solution is to employ multiple matchers for training data generation. Nevertheless, incorporating multiple matchers increases computational cost, and the limited number of matchers lacks sufficient diversity. In contrast, we develop a stochastic optimization strategy, approximating $\mathcal{D}_{\text{all}}$ via Monte Carlo sampling. Monte Carlo methods (Rubinstein and Kroese 2016; Hammersley 2013; Metropolis and Ulam 1949) are grounded in repeated random sampling, excelling in modeling complex or implicit distributions and sufficiently exploring large solution spaces. In the pipeline of image matching, we integrate Monte Carlo sampling into a match diffusion process, introducing a multi-stage randomization module. At each training iteration, random correspondences $\mathbf{C}_{\text{rnd}}$ are generated based on Monte Carlo sampling. $\mathbf{C}_{\text{rnd}}$, along with the ground truth $\mathbf{C}_{\text{gt}}$, are then utilized to optimize the learnable parameters in learning-based RANSAC. We reformulate the expected loss over $\mathcal{D}_{\text{all}}$ as an empirical approximation:

$$\mathbb{E}_{\mathbf{C}\sim\mathcal{D}_{\text{all}}}[\mathcal{L}] \approx \frac{1}{H}\sum_{i=1}^{H} \mathcal{L}\left(\mathbf{P}_\theta \mid \mathbf{C}_{\text{rnd}}^{(i)}, \mathbf{C}_{\text{gt}}^{(i)}\right), \tag{2}$$

where H denotes the number of samplings and $\mathcal{L}(\cdot)$ represents the loss used in NG-RANSAC (Brachmann and Rother 2019). As the sample size increases ($H \to \infty$), the Monte Carlo estimate of the expected loss converges to the true expectation.

Match Diffusion

To generate $\mathbf{C}_{\text{rnd}}$ in Eq. 2, we introduce a match diffusion mechanism, injecting random noise into correspondences. Notably, instead of employing diffusion for image generation (Rombach et al. 2022; Zhang, Rao, and Agrawala 2023; Croitoru et al. 2023), we utilize diffusion to simulate diverse noisy correspondences from available ground truth with varying noise across different timesteps. As the timestep $t \in \{t_1, t_2, \cdots, t_T\}$ increases, the ground truth gradually transitions to pure noise. Specifically, given ground-truth correspondences $\mathbf{C}_{\text{gt}}$ between two images, its noised version at timestep t is generated through the recursive relation:

$$\mathbf{c}_t^{(i)} = \sqrt{1-\beta_t} \cdot \mathbf{c}_{t-1}^{(i)} + \sqrt{\beta_t} \cdot \epsilon_{t-1}, \quad \epsilon_{t-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{3}$$

where $\mathbf{c}_t^{(i)}$ denotes a correspondence derived from $\mathbf{C}_{\text{gt}}$ at timestep t, ϵ indicates the 4D noise vectors sampled from standard normal distribution, and $\beta_t \in [\beta_{\text{start}}, \beta_{\text{end}}]$ controls the noise injection rate at t. We update β_t across timesteps based on a linear schedule:

$$\beta_t = \beta_{\text{start}} + \frac{t}{T}(\beta_{\text{end}} - \beta_{\text{start}}). \tag{4}$$

The recursive formulation is simplified, directly computing $\mathbf{c}_t^{(i)}$ from $\mathbf{c}_0^{(i)}$ as:

$$\mathbf{c}_t^{(i)} = \sqrt{\bar{\alpha}_t} \cdot \mathbf{c}_0^{(i)} + \sqrt{1-\bar{\alpha}_t} \cdot \epsilon, \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{5}$$

with $\bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s)$. As t increases, the injected noise in $\mathbf{c}_t^{(i)}$ grows, causing $\mathbf{c}_t^{(i)}$ to deviate further from the ground truth. Notably, in our method, $\mathbf{c}_0^{(i)}$ represents an inlier, and $\mathbf{c}_t^{(i)}$ serves as an outlier.

However, diffusing all correspondences in $\mathbf{C}_{\text{gt}}$ may lead to a scenario where the raw data lacks sufficient inliers required for model estimation. Therefore, we introduce a diffusion ratio r to control the proportion of correspondences processed by the diffusion module. As shown in Fig. 2, we randomly sample a subset of ground-truth matches, referred to as $\mathbf{C}_{\text{gt}}^a$, with a ratio of r. The remaining matches are denoted as $\mathbf{C}_{\text{gt}}^b$. Each correspondence $\mathbf{c}_0^{(i)} \in \mathbf{C}_{\text{gt}}^a$ is perturbed with noise following Eq. 5, resulting in the noised counterpart $\mathbf{C}_n^a$. In addition, since the noise is sampled from a fixed distribution $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, similar coordinate shifts may occur in correspondences from $\mathbf{C}_{\text{gt}}^a$ to $\mathbf{C}_n^a$. To enhance the diversity, we add a noise scale s during the diffusion process as:

$$\hat{\epsilon} = \epsilon \cdot s \cdot \max(W, H), \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{6}$$

where W and H represent the image width and height, respectively. Eq. 5 is then updated as:

$$\mathbf{c}_t^{(i)} = \sqrt{\bar{\alpha}_t} \cdot \mathbf{c}_0^{(i)} + \sqrt{1-\bar{\alpha}_t} \cdot \hat{\epsilon}, \quad \mathbf{c}_0^{(i)} \in \mathbf{C}_{\text{gt}}^a. \tag{7}$$

The final diffused matches are generated by combining noised and clean subsets as:

$$\mathbf{C}_{\text{rnd}} = \mathbf{C}_n^a \cup \mathbf{C}_{\text{gt}}^b, \quad \mathbf{C}_{\text{rnd}} \in \mathbb{R}^{N \times 4}. \tag{8}$$

Multi-Stage Randomization

The proposed match diffusion module generates diffused matches controlled by three hyperparameters: timestep t, diffusion ratio r, and noise scale s. Varying these hyperparameters results in different distributions for $\mathbf{C}_{\text{rnd}}$. To determine the hyperparameters, a simple approach is to fix them at predefined values. Nevertheless, such fixed hyperparameters contradict our expectation of Monte Carlo approximation, approximating $\mathcal{D}_{\text{all}}$ based on random sampling. Consequently, to enhance the randomness in the diffusion module, we propose a multi-stage randomization (MSR) method, where we inject randomness at multiple stages throughout the diffusion process.

The framework of MSR is illustrated in Fig. 3. When partitioning $\mathbf{C}_{\text{gt}}$ into $\mathbf{C}_{\text{gt}}^a$ and $\mathbf{C}_{\text{gt}}^b$, we randomly sample ratio RND-r from the range $[r_{\text{min}}, r_{\text{max}}]$ instead of using a fixed r. The ratio of noised matches in $\mathbf{C}_{\text{rnd}}$ thereby varies between r_{min} and r_{max}, leading to diverse outlier ratios. We then scale the noise ϵ using a scalar RND-s randomly sampled from the range $[s_{\text{min}}, s_{\text{max}}]$, as formulated in Eq. 6. Due to the randomness in RND-s, the scaled noise $\hat{\epsilon}$ can represent perturbations at varying levels. In addition, for each correspondence in $\mathbf{C}_{\text{gt}}^a$, we randomly sample a timestep, denoted as RND-t, from $\{t_1, t_2, \cdots, t_T\}$, and inject the noise into the correspondence as defined in Eq. 7.

Notably, during the diffusion process, some correspondences may fall outside the image bounds, making them invalid for training. To handle this problem, we check the validity of the generated correspondences after noise injection and replace invalid matches with RND-$matches$. RND-$matches$ indicate matches that are regenerated through uniform sampling:

$$x_{\text{rnd}}, x'_{\text{rnd}} \sim \mathcal{U}(0, W), \quad y_{\text{rnd}}, y'_{\text{rnd}} \sim \mathcal{U}(0, H). \tag{9}$$

This replacement ensures the validity of noised matches while further increasing the randomness in the diffusion module. By randomly sampling (RND-r, RND-s, RND-t, RND-$matches$), our MSR introduces randomness at multiple stages of the diffusion module. Fig. 4 illustrates the impact of these hyperparameters on diffused matches. For the same image pair, varying r and s results in significantly different distributions of diffused matches. Therefore, our multi-stage randomization ensures data diversity and thus facilitates effective Monte Carlo approximation. Please refer to the supplementary material for more visualization results.

Experiments

Setup

We conduct experiments on ScanNet (Dai et al. 2017) and MegaDepth (Li and Snavely 2018) that include diverse indoor and outdoor scenarios, respectively. We follow the benchmark in SuperGlue (Sarlin et al. 2020) on ScanNet, splitting the dataset into 1,513 training scenes and 100 test scenes. We randomly sample 20 image pairs per training scene to construct our training set and employ the same image pairs used in SuperGlue (Sarlin et al. 2020) during inference. On MegaDepth, we follow the setup in LoFTR (Sun et al. 2021), using 368 scenes for training and 5 scenes

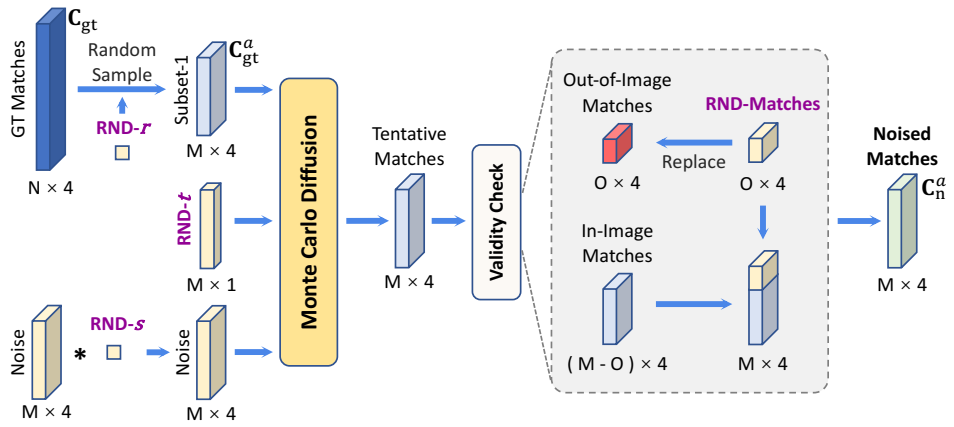

Figure 3: Illustration of the multi-stage randomization module. We randomly sample the three hyperparameters, timestep t, diffusion ratio r, and noise scale s, in the diffusion mechanism. This multi-stage randomization introduces different sources of randomness into the noised matches, affecting the diffusion intensity, outlier ratio, and noise level, respectively. Invalid matches in the tentative set are replaced by randomly sampled matches, which ensures the validity of the final diffused matches.

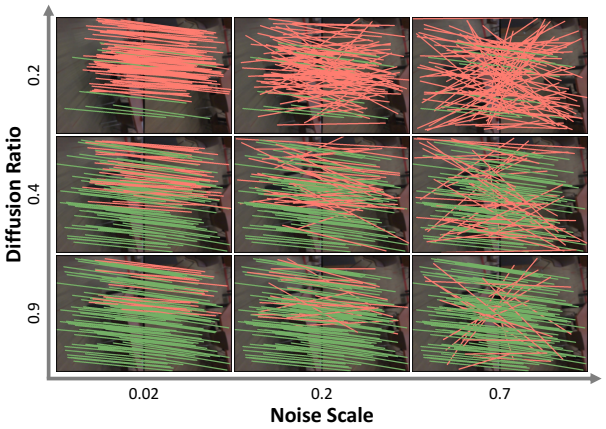

Figure 4: Visualization of diffused matches. Given the same image pair, different values of diffusion ratio and noise scale result in significantly different diffused matches.

for testing. We randomly sample 80 image pairs per scene during training and conduct evaluations on the same image pairs as (Sun et al. 2021). We employ SIFT (Lowe 2004) and LoFTR (Lowe 2004) as baselines for feature matching. SIFT represents traditional handcrafted approaches, while LoFTR exemplifies learning-based methods. Both of them have distinct advantages depending on the application and are widely used in 3D computer vision tasks. The evaluation is then performed using NG-RANSAC (Brachmann and Rother 2019) as a learning-based robust estimator by default. Specifically, we train NG-RANSAC separately on SIFT, LoFTR, and our diffused matches. We evaluate the trained models in multiple scenarios, employing AUC of the pose error as a metric (Sun et al. 2021; Sarlin et al. 2020). Note that our method is also compatible with other learning-based RANSACs, as will be evidenced in our experiments.

Training	Testing	AUC @5°	AUC @10°	AUC @20°
LoFTR	SIFT	1.5	4.4	8.5
MCD	SIFT	**7.6** (+6.1)	**16.2** (+11.8)	**26.2** (+17.7)
SIFT	LoFTR	13.0	29.7	48.8
MCD	LoFTR	**22.4** (+9.4)	**42.7** (+13.0)	**60.8** (+12.0)

Table 1: Generalization to out-of-distribution data on ScanNet. MCD indicates the diffused matches generated via our Monte Carlo diffusion. NG-RANSAC trained on SIFT is evaluated on LoFTR, and vice versa. AUCs of the pose error with different thresholds are reported, and the best results are highlighted in bold.

Implementation Details

Dataset construction. For each image pair $(\mathbf{I}, \mathbf{I}')$, the corresponding depth maps and camera parameters are utilized to reconstruct 3D points (Hartley and Zisserman 2003). These points are aligned in the world coordinate system, resulting in the ground-truth pixel-wise correspondences $\mathbf{C}_{gt}$. We randomly subsample them to obtain 2000 correct correspondences per image pair.

Training details. We set the diffusion parameters to $\beta_{start} = 0.0005$, $\beta_{end} = 0.0025$, and $T = 500$. RND-r and RND-s are randomly sampled within the ranges $[0.2, 0.9]$ and $[0.02, 0.7]$, respectively. We maintain the hyperparameters in learning-based RANSAC at their default settings to ensure fair comparisons in our benchmarks. We train the model on a GTX 2080 Ti, and the training process takes 60 hours.

Generalization to Out-of-Distribution Data

To assess the generalization to out-of-distribution (OOD) data, we conduct experiments on SIFT and LoFTR correspondences. Specifically, NG-RANSAC trained on SIFT is evaluated on LoFTR, and vice versa. Notably, the presented Monte Carlo diffusion (MCD) is agnostic to specific matchers. Therefore, for NG-RANSAC trained on diffused

Training	Testing	AUC @5°	AUC @10°	AUC @20°
LoFTR	SIFT	3.1	6.7	13.4
MCD	SIFT	**16.9** (+13.8)	**26.2** (+19.5)	**36.8** (+23.4)
SIFT	LoFTR	41.8	59.0	73.5
MCD	LoFTR	**53.2** (+11.4)	**67.7** (+8.7)	**79.2** (+5.7)

Table 2: Generalization to out-of-distribution data on MegaDepth.

Training	Testing	AUC @5°	AUC @10°	AUC @20°
SIFT	SIFT	7.6	16.1	25.5
MCD	SIFT	**7.6**	**16.2**	**26.2**
LoFTR	LoFTR	**22.7**	42.7	60.1
MCD	LoFTR	22.4	**42.7**	**60.8**

Table 3: Comparisons in in-distribution scenarios on ScanNet. The baseline trains and tests NG-RANSAC on the same matcher, while our method trains NG-RANSAC only using diffused matches.

Training	Testing	AUC @5°	AUC @10°	AUC @20°
SIFT	SIFT	16.6	**27.0**	**38.8**
MCD	SIFT	**16.9**	26.2	36.8
LoFTR	LoFTR	**53.3**	67.5	**79.3**
MCD	LoFTR	53.2	**67.7**	79.2

Table 4: Comparisons in in-distribution scenarios on MegaDepth.

matches, we directly test it on SIFT and LoFTR. Table 1 and Table 2 list the results on ScanNet and MegaDepth, respectively. The models trained on the diffused matches obtained through Monte Carlo match diffusion exhibit significant improvements in generalization ability. For instance, on ScanNet, MCD improves AUC @20° by 12% from 48.8% to 60.8% on LoFTR, when compared with the model trained on SIFT. Notably, NG-RANSAC trained on LoFTR yields limited AUCs on SIFT, e.g., 8.5% in AUC @20°. Our method significantly enhances the generalization in this case, improving AUC @20° by 17.7% from 8.5% to 26.2%. Additionally, we achieve consistent improvements on MegaDepth, increasing AUC @20° by 5.7% and 23.4% when tested on LoFTR and SIFT, respectively.

Comparisons in In-Distribution Scenarios

In some scenarios, conducting inference on in-distribution data is practical. For example, LoFTR can be deployed in a system with enough computational resources for both training and testing. To assess the applicability of our method in such a setting, we compare our method with NG-RANSAC trained and tested on the same matcher. More specifically, we train NG-RANSAC on SIFT and test it on SIFT, and then repeat this process on LoFTR. In contrast, for our method, we retain the diffused matches during training and test the trained NG-RANSAC on both SIFT and LoFTR. As reported in Table 3 and Table 4, our method achieves performance comparable to the baseline, where NG-RANSAC is trained and tested on the same matcher. This evidences that our method not only shows superior generalization but also remains highly competitive in scenarios where the same matcher can be applied for both training and testing.

Scenario	Training	Testing	AUC @5°	AUC @10°	AUC @20°
OOD	LoFTR	SIFT	2.7	4.4	10.5
	MCD	SIFT	**7.6**	**15.8**	**26.2**
OOD	SIFT	LoFTR	22.0	41.4	58.8
	MCD	LoFTR	**23.5**	**43.1**	**60.2**
ID	SIFT	SIFT	7.3	**16.6**	**28.5**
	MCD	SIFT	**7.6**	15.8	26.2
ID	LoFTR	LoFTR	23.1	42.8	60.0
	MCD	LoFTR	**23.5**	**43.1**	**60.2**

Table 5: Compatibility with ∇-RANSAC on ScanNet. AUCs on out-of-distribution (OOD) and in-distribution (ID) scenarios are listed.

Ablation Study

Compatibility with Learning-Based RANSACs Recall that we conduct experiments using NG-RANSAC by default. To shed more light on the compatibility with other learning-based RANSACs, we repeat the experiments in Table 1 and Table 3, employing another representative learning-based RANSAC, ∇-RANSAC (Wei et al. 2023). We report the results on ScanNet in Table 5. As demonstrated, our method maintains its superiority when combined with ∇-RANSAC. The model trained on the diffused matches exhibits better generalization to out-of-distribution data, and its results are comparable to those trained and tested on the same matcher. These experiments highlight Monte Carlo diffusion as a universal enhancer for learning-based RANSACs, improving the generalization ability.

Comparisons with Traditional RANSACs In addition to learning-based RANSACs, we conduct comparisons between our method and traditional RANSAC variants. Specifically, we maintain the experimental setup, training NG-RANSAC on diffused matches as our method. We evaluate the trained model and the handcrafted methods, RANSAC and MAGSAC++, on SIFT and LoFTR. The results on ScanNet are presented in Table 6. Traditional methods exhibit limited performance on SIFT, resulting in an AUC @5° of 0.6 for RANSAC and 0.8 for MAGSAC++. As SIFT typically produces noisy initial matches with numerous outliers, this finding indicates that handcrafted RANSACs are sensitive to outlier ratios, limiting their effectiveness in applications requiring robust estimation from low-quality data. In contrast, NG-RANSAC trained on our diffused matches demonstrates superior adaptability across scenarios with varying data quality.

Adaptability to Foundation Models Recently, foundation models (Wang et al. 2024, 2025) have attracted grow-

Method	Testing	AUC @5°	AUC @10°	AUC @20°
RANSAC	SIFT	0.6	2.1	4.8
MAGSAC++	SIFT	0.8	2.3	5.4
Ours	SIFT	**7.6** (+6.8)	**16.2** (+13.9)	**26.2** (+20.8)
RANSAC	LoFTR	20.4	39.4	58.1
MAGSAC++	LoFTR	21.6	41.1	59.3
Ours	LoFTR	**22.4** (+0.8)	**42.7** (+1.6)	**60.8** (+1.5)

Table 6: Comparison with handcrafted RANSAC variants on ScanNet. We train NG-RANSAC based on MCD and compare the model with RANSAC and MAGSAC++.

ing attention for their strong performance in 3D computer vision tasks. Recall that the major advantage of generalizable learning-based RANSAC is its seamless integration with newly developed matchers. Therefore, we conduct an experiment where we employ VGGT (Wang et al. 2025), a state-of-the-art matcher (Cong et al. 2025), to generate correspondences during testing. To better highlight the effect of RANSAC, we use unfiltered VGGT matches, which still maintain higher matching quality than LoFTR (Sun et al. 2021). As shown in Table 7, models trained on our MCD exhibit better adaptability to VGGT compared with both the vanilla RANSAC and those trained on a specific matcher.

Training	Testing	AUC @5°	AUC @10°	AUC @20°
SIFT	VGGT	22.4	42.8	61.8
LoFTR	VGGT	24.8	45.4	64.0
MCD	VGGT	**27.3** (+2.5)	**48.6** (+3.2)	**67.0** (+3.0)
RANSAC + VGGT		25.2	47.1	66.5

Table 7: Adaptability to VGGT. RANSAC+VGGT indicates a baseline that directly applies RANSAC to VGGT. AUCs on ScanNet (Dai et al. 2017) are reported.

Monte Carlo Approximation MCD approximates $\mathcal{D}_{\text{all}}$ through Monte Carlo sampling (Metropolis and Ulam 1949). A straightforward alternative is to combine multiple matchers for data generation. To this end, we construct a training dataset using both SIFT and LoFTR correspondences, doubling the size of the original dataset built with a single matcher. Another baseline applies a simple data augmentation by perturbing ground-truth correspondences with Gaussian noise sampled from $\mathcal{N}(\mathbf{0}, \mathbf{I})$, thereby generating noised correspondences without our Monte Carlo diffusion. We evaluate these baselines on ScanNet and report results in Table 8. During testing, SuperGlue is used to generate correspondences, serving as out-of-distribution data. The baselines perform worse than our MCD, demonstrating that the fixed matchers or noise distribution struggle with sufficiently exploring $\mathcal{D}_{\text{all}}$. In contrast, our method is independent of specific matchers, and the inherent randomness of the Monte Carlo diffusion module enables diverse sampling from $\mathcal{D}_{\text{all}}$. This advantage thereby leads to improved generalization.

Multi-Stage Randomization To assess the contribution of the parameters, i.e., timestep t, diffusion ratio r, and noise scale s, in the MSR module, we progressively incorporate them into our framework. We train NG-RANSAC on diffused matches generated by each variant, and Table 9 lists the ablation results on ScanNet. The ablation starts with random t while keeping $r = 0.5$ and $s = 0.1$ fixed. The inclusion of random r and s improves performance on both SIFT and LoFTR during testing, increasing AUC @20° by 19.6% and 4.0% on SIFT and LoFTR, respectively. These parameters introduce different types of randomness: t determines the distribution of the noised data, r controls the outlier ratio in diffused matches, and s defines the noise level. Their combined effects enhance the diversity of generated data, leading to an optimal solution in our pipeline.

Training	Testing	AUC @5°	AUC @10°	AUC @20°
SIFT+LoFTR	SG	15.5	33.4	51.8
$\mathcal{N}(\mathbf{0}, \mathbf{I})$	SG	11.6	27.1	45.9
MCD	SG	**17.4** (+1.9)	**35.5** (+2.1)	**54.0** (+2.2)

Table 8: Effectiveness of Monte Carlo approximation. $\mathcal{N}(\mathbf{0}, \mathbf{I})$ denotes a simple data augmentation perturbing GT correspondences with Gaussian noise. The trained NG-RANSAC is tested on out-of-distribution data generated via SuperGlue (SG). AUCs on ScanNet are reported.

t	r	s	Testing	AUC @5°	AUC @10°	AUC @20°°
✓	-	-	SIFT	1.1	3.2	6.6
✓	-	-	LoFTR	20.1	39.0	56.8
✓	✓	-	SIFT	4.9 (+3.8)	10.6 (+7.4)	18.3 (+11.7)
✓	✓	-	LoFTR	21.7 (+1.6)	42.2 (+3.2)	59.8 (+3.0)
✓	✓	✓	SIFT	7.6 (+2.7)	16.2 (+5.6)	26.2 (+7.9)
✓	✓	✓	LoFTR	22.4 (+0.7)	42.7 (+0.5)	60.8 (+1.0)

Table 9: Ablation study on MSR. t, r, and s represent timestep, diffusion ratio, and noise scale, respectively. Each of these components introduces different types of randomness in MSR. We progressively add them to our pipeline, and the results on ScanNet are shown.

Conclusion

In this paper, we have presented a Monte Carlo diffusion mechanism that enhances the generalization ability of learning-based RANSAC. Our approach decouples training from fixed data sources by simulating diverse noise patterns through a diffusion-driven process. By incorporating Monte Carlo sampling, we inject randomness at multiple stages of the diffusion module, further increasing data diversity and robustness. We have evaluated our method on ScanNet and MegaDepth, demonstrating that learning-based RANSAC trained based on Monte Carlo diffusion achieves significantly better generalization when tested on out-of-distribution data and competitive performance in in-distribution scenarios.

References

Barath, D.; Matas, J.; and Noskova, J. 2019. Magsac: marginalizing sample consensus. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 10197–10205.

Baumberg, A. 2000. Reliable feature matching across widely separated views. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, 774–781. IEEE.

Bay, H.; Ess, A.; Tuytelaars, T.; and Van Gool, L. 2008. Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3): 346–359.

Bian, J.; Lin, W.-Y.; Matsushita, Y.; Yeung, S.-K.; Nguyen, T.-D.; and Cheng, M.-M. 2017. Gms: Grid-based motion statistics for fast, ultra-robust feature correspondence. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 4181–4190.

Brachmann, E.; Krull, A.; Nowozin, S.; Shotton, J.; Michel, F.; Gumhold, S.; and Rother, C. 2017. Dsac-differentiable ransac for camera localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 6684–6692.

Brachmann, E.; and Rother, C. 2019. Neural-guided RANSAC: Learning where to sample model hypotheses. In *Int. Conf. Comput. Vis.*, 4322–4331.

Brown, M.; and Lowe, D. G. 2007. Automatic panoramic image stitching using invariant features. *Int. J. Comput. Vis.*, 74(1): 59–73.

Chum, O.; Matas, J.; and Kittler, J. 2003. Locally optimized RANSAC. In *Joint Pattern Recognition Symposium*, 236–243. Springer.

Coifman, R. R.; and Lafon, S. 2006. Diffusion maps. *Applied and computational harmonic analysis*, 21(1): 5–30.

Cong, W.; Liang, Y.; Zhang, Y.; Yang, Z.; Wang, Y.; Ivanovic, B.; Pavone, M.; Chen, C.; Wang, Z.; and Fan, Z. 2025. E3D-Bench: A Benchmark for End-to-End 3D Geometric Foundation Models. *arXiv preprint arXiv:2506.01933*.

Croitoru, F.-A.; Hondru, V.; Ionescu, R. T.; and Shah, M. 2023. Diffusion models in vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(9): 10850–10869.

Dai, A.; Chang, A. X.; Savva, M.; Halber, M.; Funkhouser, T.; and Nießner, M. 2017. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 5828–5839.

Edstedt, J.; Sun, Q.; Bökman, G.; Wadenbäck, M.; and Felsberg, M. 2024. RoMa: Robust Dense Feature Matching. *IEEE Conference on Computer Vision and Pattern Recognition*.

Einstein, A. 1905. Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen. *Annalen der physik*, 4.

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

Gao, J.; Li, Y.; Chin, T.-J.; and Brown, M. S. 2013. Seam-driven image stitching. In *Eurographics (Short Papers)*, 45–48. Girona.

Hammersley, J. 2013. *Monte carlo methods*. Springer Science & Business Media.

Hartley, R.; and Zisserman, A. 2003. *Multiple view geometry in computer vision*. Cambridge university press.

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

Kim, H. J.; Dunn, E.; and Frahm, J.-M. 2017. Learned contextual feature reweighting for image geo-localization. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 3251–3260. IEEE.

Li, Z.; and Snavely, N. 2018. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2041–2050.

Lowe, D. G. 2004. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.*, 60(2): 91–110.

Ma, J.; Zhao, J.; Jiang, J.; Zhou, H.; and Guo, X. 2019. Locality preserving matching. *Int. J. Comput. Vis.*, 127(5): 512–531.

Metropolis, N.; and Ulam, S. 1949. The monte carlo method. *Journal of the American statistical association*, 44(247): 335–341.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1): 99–106.

Mur-Artal, R.; Montiel, J. M. M.; and Tardos, J. D. 2015. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5): 1147–1163.

Perona, P.; and Malik, J. 2002. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on pattern analysis and machine intelligence*, 12(7): 629–639.

Raguram, R.; Chum, O.; Pollefeys, M.; Matas, J.; and Frahm, J.-M. 2012. USAC: a universal framework for random sample consensus. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8): 2022–2038.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10684–10695.

Rubinstein, R. Y.; and Kroese, D. P. 2016. *Simulation and the Monte Carlo method*. John Wiley & Sons.

Rublee, E.; Rabaud, V.; Konolige, K.; and Bradski, G. 2011. ORB: An efficient alternative to SIFT or SURF. In *Int. Conf. Comput. Vis.*, 2564–2571. Ieee.

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

Snavely, N.; Seitz, S. M.; and Szeliski, R. 2008. Modeling the world from internet photo collections. *Int. J. Comput. Vis.*, 80(2): 189–210.

Stroock, D. W.; and Varadhan, S. S. 1997. *Multidimensional diffusion processes*. Springer Science & Business Media.

Sun, J.; Shen, Z.; Wang, Y.; Bao, H.; and Zhou, X. 2021. LoFTR: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8922–8931.

Szeliski, R. 1994. Image mosaicing for tele-reality applications. In *Proceedings of 1994 IEEE Workshop on Applications of Computer Vision*, 44–53. IEEE.

Wang, J.; Chen, M.; Karaev, N.; Vedaldi, A.; Rupprecht, C.; and Novotny, D. 2025. Vggt: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5294–5306.

Wang, S.; Leroy, V.; Cabon, Y.; Chidlovskii, B.; and Revaud, J. 2024. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20697–20709.

Wei, T.; Patel, Y.; Shekhovtsov, A.; Matas, J.; and Barath, D. 2023. Generalized differentiable RANSAC. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 17649–17660.

Yi, K. M.; Trulls, E.; Ono, Y.; Lepetit, V.; Salzmann, M.; and Fua, P. 2018. Learning to find good correspondences. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2666–2674.

Yu, J.-J.; Kim, J.-H.; Kim, H.-m.; Choi, I.; and Jeong, I.-K. 2009. Real-time camera tracking for augmented reality. In *2009 11th International Conference on Advanced Communication Technology*, volume 2, 1286–1288. IEEE.

Zhang, J.; Sun, D.; Luo, Z.; Yao, A.; Zhou, L.; Shen, T.; Chen, Y.; Quan, L.; and Liao, H. 2019. Learning two-view correspondences and geometry using order-aware network. In *Int. Conf. Comput. Vis.*, 5845–5854.

Zhang, L.; Rao, A.; and Agrawala, M. 2023. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3836–3847.

Zhao, C.; Cao, Z.; Li, C.; Li, X.; and Yang, J. 2019. NM-Net: Mining reliable neighbors for robust feature correspondences. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 215–224.

Zhao, C.; Cao, Z.; Yang, J.; Xian, K.; and Li, X. 2020. Image Feature Correspondence Selection: A Comparative Study and a New Contribution. *IEEE Trans. Image Process.*, 29: 3506–3519.

Zhao, C.; Ge, Y.; Zhu, F.; Zhao, R.; Li, H.; and Salzmann, M. 2021. Progressive correspondence pruning by consensus learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6464–6473.