

6D-Diff: A Keypoint Diffusion Framework for 6D Object Pose Estimation

Li Xu^{1†} Haoxuan Qu^{1†} Yujun Cai² Jun Liu^{1‡}

¹Singapore University of Technology and Design

²Nanyang Technological University

{li_xu, haoxuan_qu}@mymail.sutd.edu.sg, yujun001@e.ntu.edu.sg,

jun.liu@sutd.edu.sg

Abstract

Estimating the 6D object pose from a single RGB image often involves noise and indeterminacy due to challenges such as occlusions and cluttered backgrounds. Meanwhile, diffusion models have shown appealing performance in generating high-quality images from random noise with high indeterminacy through step-by-step denoising. Inspired by their denoising capability, we propose a novel diffusion-based framework (**6D-Diff**) to handle the noise and indeterminacy in object pose estimation for better performance. In our framework, to establish accurate 2D-3D correspondence, we formulate 2D keypoints detection as a reverse diffusion (denoising) process. To facilitate such a denoising process, we design a Mixture-of-Cauchy-based forward diffusion process and condition the reverse process on the object appearance features. Extensive experiments on the LM-O and YCB-V datasets demonstrate the effectiveness of our framework.

1. Introduction

6D object pose estimation aims to estimate the 6D pose of an object including its location and orientation, which has a wide range of applications, such as augmented reality [39, 47], robotic manipulation [3, 45], and automatic driving [62]. Recently, various methods [4, 5, 19, 22, 27, 44, 53, 61, 64] have been proposed to conduct RGB-based 6D object pose estimation since RGB images are easy to obtain. Despite the increased efforts, a variety of challenges persist in RGB-based 6D object pose estimation, including occlusions, cluttered backgrounds, and changeable environments [8, 40, 44, 60, 63]. These challenges can introduce significant noise and indeterminacy into the pose estimation process, leading to error-prone predictions [8, 40, 44].

Meanwhile, diffusion models [18, 52] have achieved ap-

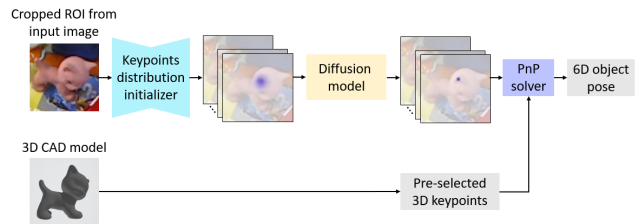


Figure 1. Overview of our proposed **6D-Diff** framework. As shown, given the 3D keypoints from the object 3D CAD model, we aim to detect the corresponding 2D keypoints in the image to obtain the 6D object pose. Note that when detecting keypoints, there are often challenges such as occlusions (including self-occlusions) and cluttered backgrounds that can introduce noise and indeterminacy into the process, impacting the accuracy of pose prediction.

pealing results in various generation tasks such as image synthesis [7, 18] and image editing [41]. Specifically, diffusion models are able to recover high-quality determinate samples (e.g., clean images) from a noisy and indeterminate input data distribution (e.g., random noise) via a step-by-step denoising process [18, 52]. Motivated by such a strong denoising capability [11, 12, 18], we aim to leverage diffusion models to handle the RGB-based 6D object pose estimation task, since this task also involves tackling noise and indeterminacy. However, it can be difficult to directly use diffusion models to estimate the object pose, because diffusion models often start denoising from random Gaussian noise [18, 52]. Meanwhile, in RGB-based 6D object pose estimation, the object pose is often extracted from an intermediate representation, such as keypoint heatmaps [5], pixel-wise voting vectors [44], or object surface keypoint features [4]. Such an intermediate representation encodes useful distribution priors about the object pose. Thus starting denoising from such an representation shall effectively assist the diffusion model in recovering accurate object poses [11]. To achieve this, we propose a novel diffusion-based object pose estimation framework (**6D-Diff**) that can exploit prior distribution knowledge from the intermediate representation for better performance.

† Equal contribution; ‡ Corresponding author

Overall, our framework is a *correspondence-based* framework, in which to predict an object pose, given the 3D keypoints pre-selected from the object 3D CAD model, we first predict the coordinates of the 2D image keypoints corresponding to the pre-selected 3D keypoints. We then use the 3D keypoints together with the predicted 2D keypoints coordinates to compute the 6D object pose using a Perspective-n-Point (PnP) solver [10, 31]. As shown in Fig. 1, to predict the 2D keypoints coordinates, we first extract an intermediate representation (the 2D keypoints heatmaps) through a keypoints distribution initializer. As discussed before, due to various factors, there often exists noise and indeterminacy in the keypoints detection process and the extracted heatmaps can be noisy as shown in Fig. 2. Thus we pass the distribution modeled from these keypoints heatmaps into a diffusion model to perform the denoising process to obtain the final keypoints coordinates prediction.

Analogous to non-equilibrium thermodynamics [50], given a 2D image keypoint, we can consider all its possible locations in the image as particles in thermodynamics. Under low indeterminacy, the particles (possible locations) w.r.t. each 2D keypoint gather, and each keypoint can be determinately and accurately localized. In contrast, under high indeterminacy, these particles can stochastically spread over the input image, and it is difficult to localize each keypoint. The process of converting particles from low indeterminacy to high indeterminacy is called the *forward process* of the diffusion model. The goal of the diffusion model is to reverse the above forward process (through a *reverse process*), i.e., converting the particles from high indeterminacy to low indeterminacy. Here in our case, we aim to convert the indeterminate keypoints coordinates distribution modeled from the heatmaps into the determinate distribution. Below we briefly introduce the forward process and the reverse process in our diffusion model.

In the forward process, we aim to generate supervision signals that will be used to optimize the diffusion model during the reverse process. Specifically, given a set of pre-selected 3D keypoints, we first acquire ground-truth coordinates of their corresponding 2D keypoints using the ground-truth object pose. Then these determinate ground-truth 2D coordinates are gradually diffused towards the indeterminate distribution modeled from the intermediate representation, and the distributions generated along the way will be used as supervision signals. Note that, as the distribution modeled from the intermediate representation can be complex and irregular, it is difficult to characterize such a distribution via the Gaussian distribution. This means that simply applying diffusion models in most existing generation works [7, 18, 52], which start denoising from the random Gaussian noise, can introduce potentially large errors. To tackle this challenge, we draw inspiration from the fact that the Mixture of Cauchy (MoC) model can effectively char-

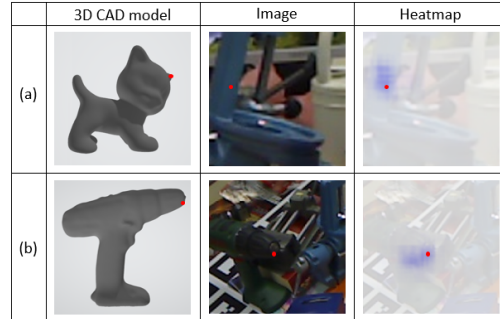


Figure 2. Above we show two examples of keypoint heatmaps, which serve as the intermediate representation [4, 5, 44] in our framework. The red dots indicate the ground-truth locations of the keypoints. In the example (a), the target object is the pink cat, which is heavily occluded in the image and is shown in a different pose compared to the 3D model. As shown above, due to occlusions and cluttered backgrounds, the keypoint heatmaps are noisy, which reflects the noise and indeterminacy during the keypoints detection process.

acterize complex and intractable distributions. Moreover, the MoC model is robust to potential outliers in the distribution to be characterized [26]. Thus we propose to model the intermediate representation using a MoC distribution instead of simply treating it as a random Gaussian noise. In this way, we gradually diffuse the determinate distribution (ground truth) of keypoints coordinates towards the modeled MoC distribution during the forward process.

Correspondingly, in the reverse process, starting from the MoC distribution modeled in the forward process, we aim to learn to recover the ground-truth keypoints coordinates. To achieve this, we leverage the distributions generated step-by-step during the forward process as the supervision signals to train the diffusion model to learn the reverse process. In this way, the diffusion model can learn to convert the indeterminate MoC distribution of keypoints coordinates into a determinate one smoothly and effectively. After the reverse process, the 2D keypoints coordinates obtained from the final determinate distribution are used to compute the 6D object pose with the pre-selected 3D keypoints. Moreover, we further facilitate the model learning of such a reverse process by injecting object appearance features as context information.

Our work makes the following contributions. 1) We propose a novel **6D-Diff** framework, in which we formulate keypoints detection for 6D object pose estimation as a reverse diffusion process to effectively eliminate the noise and indeterminacy in object pose estimation. 2) To take advantage of the intermediate representation that encodes useful prior distribution knowledge for handling this task, we propose a novel MoC-based diffusion process. Besides, we facilitate the model learning by utilizing object features.

2. Related Work

RGB-based 6D Object Pose Estimation has received a lot of attention [4, 13–16, 23, 32, 33, 36, 38, 43, 44, 46, 53, 54, 56, 63–67]. Some works [22, 27, 61, 63] proposed to directly regress object poses. However, the non-linearity of the rotation space makes direct regression of object poses difficult [32]. Compared to this type of *direct methods*, *correspondence-based methods* [5, 19, 43, 44, 46, 53, 56] often demonstrate better performance, which estimate 6D object poses via learning 2D-3D correspondences between the observed image and the object 3D model.

Among *correspondence-based methods*, several works [42, 44, 46, 48, 56] aim to predict the 2D keypoints coordinates corresponding to specific 3D keypoints. BB8 [46] proposed to detect the 2D keypoints corresponding to the 8 corners of the object’s 3D bounding box. Later, PVNet [44] achieved better performance by estimating 2D keypoints for sampled points on the surface of the object 3D model via pixel-wise voting. Moreover, various methods [19, 43, 53, 61, 67] establish 2D-3D correspondences by localizing the 3D model point corresponding to each observed object pixel. Among these methods, DPOD [67] explored the use of UV texture maps to facilitate model training, and ZebraPose [53] proposed to encode the surface of the object 3D model efficiently through a hierarchical binary grouping. Besides, several pose refinement methods [23, 33, 38, 64] have been proposed, which conducted pose refinement given an initial pose estimation.

In this paper, we also regard object pose estimation as a 2D-3D correspondence estimation problem. Different from previous works, here by formulating 2D-3D correspondence estimation as a distribution transformation process (denoising process), we propose a new framework (**6D-Diff**) that trains a diffusion model to perform progressive denoising from an indeterminate keypoints distribution to the desired keypoints distribution with low indeterminacy.

Diffusion Models [7, 9, 18, 50, 52] are originally introduced for image synthesis. Showing appealing generation capabilities, diffusion models have also been explored in various other tasks [11, 12, 20, 25, 30, 37, 41, 58], such as image editing [41] and image inpainting [37]. Here we explore a new framework that tackles object pose estimation with a diffusion model. Different from previous generation works [7, 37, 41] that start denoising from random noise, to aid the denoising process for 6D object pose estimation, we design a novel MoC-based diffusion mechanism that enables the diffusion model to start denoising from a distribution containing useful prior distribution knowledge regarding the object pose. Moreover, we condition the denoising process on the object appearance features, to further guide the diffusion model to obtain accurate predictions.

3. Method

To handle the noise and indeterminacy in RGB-based 6D object pose estimation, inspired by [11], from a novel perspective of distribution transformation with progressive denoising, we propose a framework (**6D-Diff**) that represents a new brand of diffusion-based solution for 6D object pose estimation. Below we first revisit diffusion models in Sec. 3.1. Then we discuss our proposed framework in Sec. 3.2, and introduce its training and testing scheme in Sec. 3.3. We finally detail the model architecture in Sec. 3.4.

3.1. Revisiting Diffusion Models

The diffusion model [18, 52], which is a kind of probabilistic generative model, consists of two parts, namely the forward process and the reverse process. Specifically, given an original sample d_0 (e.g., a clean image), the process of diffusing the sample d_0 iteratively towards the noise (typically Gaussian noise) $d_K \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ (i.e., $d_0 \rightarrow d_1 \rightarrow \dots \rightarrow d_K$) is called the forward process. In contrast, the process of denoising the noise d_K iteratively towards the sample d_0 (i.e., $d_K \rightarrow d_{K-1} \rightarrow \dots \rightarrow d_0$) is called the reverse process. Each process is defined as a Markov chain.

Forward Process. To obtain supervision signals for training the diffusion model to learn to perform the reverse process in a stepwise manner, we need to acquire the intermediate step results $\{d_k\}_{k=1}^{K-1}$. Thus the forward process is first performed to generate these intermediate step results for training purpose. Specifically, the posterior distribution $q(d_{1:K}|d_0)$ from d_1 to d_K is formulated as:

$$q(d_{1:K}|d_0) = \prod_{k=1}^K q(d_k|d_{k-1}) \quad (1)$$

$$q(d_k|d_{k-1}) = \mathcal{N}(d_k; \sqrt{1 - \beta_k}d_{k-1}, \beta_k\mathbf{I})$$

where $\{\beta_k \in (0, 1)\}_{k=1}^K$ denotes a set of fixed variance controllers that control the scale of the injected noise at different steps. According to Eq. (1), we can derive $q(d_k|d_0)$ in closed form as:

$$q(d_k|d_0) = \mathcal{N}(d_k; \sqrt{\bar{\alpha}_k}d_0, (1 - \bar{\alpha}_k)\mathbf{I}) \quad (2)$$

where $\alpha_k = 1 - \beta_k$ and $\bar{\alpha}_k = \prod_{s=1}^k \alpha_s$. Based on Eq. (2), d_k can be further expressed as:

$$d_k = \sqrt{\bar{\alpha}_k}d_0 + \sqrt{1 - \bar{\alpha}_k}\epsilon \quad (3)$$

where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. From Eq. (3), we can observe that when the number of diffusion steps K is sufficiently large and $\bar{\alpha}_K$ correspondingly decreases to nearly zero, the distribution of d_K is approximately a standard Gaussian distribution, i.e., $d_K \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. This means d_0 is gradually corrupted into Gaussian noise, which conforms to the non-equilibrium thermodynamics phenomenon of the diffusion process [50].

Reverse Process. With the intermediate step results $\{d_k\}_{k=1}^{K-1}$ acquired in the forward process, the diffusion

model is trained to learn to perform the reverse process. Specifically, in the reverse process, each step can be formulated as a function f that takes d_k and the diffusion model M_{diff} as inputs and generate d_{k-1} as the output, i.e., $d_{k-1} = f(d_k, M_{diff})$.

After training the diffusion model, during inference, we do not need to conduct the forward process. Instead, we only conduct the reverse process, which converts a random Gaussian noise $d_K \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ into a sample d_0 of the desired distribution using the trained diffusion model.

3.2. Proposed Framework

Similar to previous works [21, 44, 53], our framework predicts 6D object poses via a two-stage pipeline. Specifically, (i) we first select N 3D keypoints on the object CAD model and detect the corresponding N 2D keypoints in the image; (ii) we then compute the 6D pose using a PnP solver. Here we mainly focus on the first stage and aim to produce more accurate keypoint detection results.

When detecting 2D keypoints, factors like occlusions and cluttered backgrounds can bring noise and indeterminacy into this process, and affect the accuracy of detection results [21, 44]. To handle this problem, inspired by that diffusion models can iteratively reduce indeterminacy and noise in the initial distribution (e.g., standard Gaussian distribution) to generate determinate and high-quality samples of the desired distribution [11, 12], we formulate keypoints detection as generating a determinate distribution of keypoints coordinates (D_0) from an indeterminate initial distribution (D_K) via a diffusion model.

Moreover, to effectively adapt to the 6D object pose estimation task, the diffusion model in our framework does not start the reverse process from the common initial distribution (i.e., the standard Gaussian distribution) as in most existing diffusion works [7, 18, 52]. Instead, inspired by recent 6D object pose estimation works [4, 5, 61], we first extract an intermediate representation (e.g., heatmaps), and use this representation to initialize a keypoints coordinates distribution (i.e., D_K), which will serve as the starting point of the reverse process. Such an intermediate representation encodes useful prior distribution information about keypoints coordinates. Thus by starting the reverse process from this representation, we effectively exploit the distribution priors in the representation to aid the diffusion model in recovering accurate keypoints coordinates [11]. Below, we first describe how we initialize the keypoints distribution D_K , and then discuss the corresponding forward and reverse processes in our new framework.

Keypoints Distribution Initialization. We initialize the keypoints coordinates distribution D_K with extracted heatmaps. Specifically, similar to [29, 34, 53], we first use an off-the-shelf object detector (e.g., Faster RCNN [49]) to detect the bounding box of the target object, and then crop

the detected Region of Interest (ROI) from the input image. We send the ROI into a sub-network (i.e., the keypoints distribution initializer) to predict a number of heatmaps where each heatmap corresponds to one 2D keypoint. We then normalize each heatmap to convert it to a probability distribution. In this way, each normalized heatmap naturally represents the distribution of the corresponding keypoint coordinates, and thus we can use these heatmaps to initialize D_K .

Forward Process. After distribution initialization, the next step is to iteratively reduce the noise and indeterminacy in the initialized distribution D_K by performing the reverse process ($D_K \rightarrow D_{K-1} \rightarrow \dots \rightarrow D_0$). To train the diffusion model to perform such a reverse process, we need to obtain the distributions generated along the way (i.e., $\{D_k\}_{k=1}^{K-1}$) as the supervision signals. Thus, we first need to conduct the forward process to obtain samples from $\{D_k\}_{k=1}^{K-1}$. Specifically, given the ground-truth keypoints coordinates distribution D_0 , we define the forward process as: $D_0 \rightarrow D_1 \rightarrow \dots \rightarrow D_K$, where K is the number of diffusion steps. In this forward process, we iteratively add noise to the determinate distribution D_0 , i.e., increasing the indeterminacy of generated distributions, to transform it into the initialized distribution D_K with indeterminacy. Via this process, we can generate $\{D_k\}_{k=1}^{K-1}$ along the way and use them as supervision signals to train the diffusion model to perform the reverse process.

However, in our framework, we do not aim to transform the ground-truth keypoints coordinates distribution D_0 towards a standard Gaussian distribution via the forward process, because our initialized distribution D_K is not a random noise. Instead, as discussed before, D_K is initialized with heatmaps (as shown in Fig. 3), since the heatmaps can provide rough estimations about the keypoints coordinates distribution. To effectively utilize such priors in D_K to facilitate the reverse process, we aim to enable the diffusion model to start the reverse process (denoising process) from D_K instead of random Gaussian noise [11]. Thus, the basic forward process (described in Sec. 3.1) in existing generative diffusion models is not suitable in our framework, which motivates us to design a new forward process for our task.

However, it is non-trivial to design such a forward process, as the initialized distribution D_K is based on extracted heatmaps, and thus D_K can be complex and irregular, as shown in Fig. 4. Hence modeling D_K as a Gaussian distribution can result in potentially large errors. To handle this challenge, motivated by that the Mixture of Cauchy (MoC) model can effectively and reliably characterize complex and intractable distributions [26], we leverage MoC to characterize D_K . Based on the characterized distribution, we can then perform a corresponding MoC-based forward process.

Specifically, we denote the number of Cauchy kernels

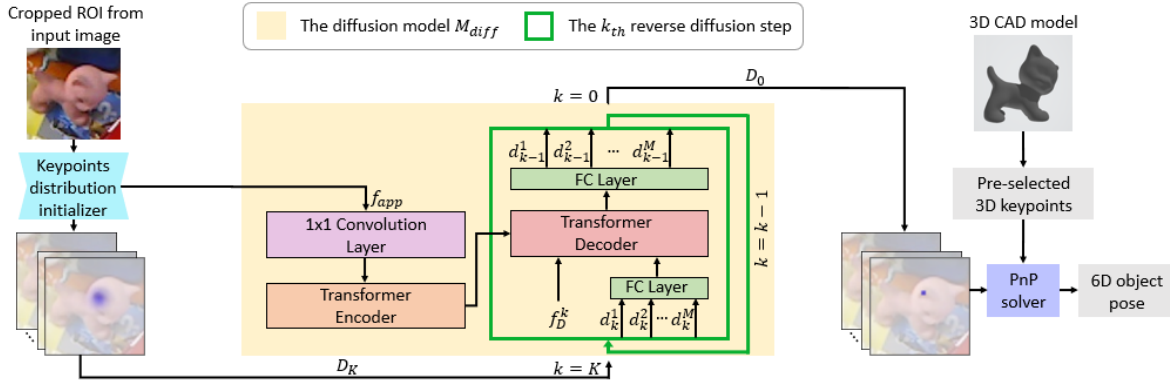


Figure 3. Illustration of our framework. During testing, given an input image, we first crop the Region of Interest (ROI) from the image through an object detector. After that, we feed the cropped ROI to the keypoints distribution initializer to obtain the heatmaps that can provide useful distribution priors about keypoints, to initialize D_K . Meanwhile, we can obtain object appearance features f_{app} . Next, we pass f_{app} into the encoder, and the output of the encoder will serve as conditional information to aid the reverse process in the decoder. We sample M sets of 2D keypoints coordinates from D_K , and feed these M sets of coordinates into the decoder to perform the reverse process iteratively together with the step embedding f_D^k . At the final reverse step (K -th step), we average $\{d_0^i\}_{i=1}^M$ as the final keypoints coordinates prediction d_0 , and use d_0 to compute the 6D pose with the pre-selected 3D keypoints via a PnP solver.

in the MoC distribution as U , and use the Expectation-Maximum-type (EM) algorithm [26, 55] to optimize the MoC parameters η^{MoC} to characterize the distribution D_K as:

$$\eta_*^{\text{MoC}} = \text{EM} \left(\prod_{v=1}^V \sum_{u=1}^U \pi_u \text{Cauchy}(d_K^v | \mu_u, \gamma_u) \right) \quad (4)$$

where $\{d_K^v\}_{v=1}^V$ denotes V sets of keypoints coordinates sampled from the distribution D_K . Note each set of keypoints coordinates d_K^v contains all the N keypoints coordinates (i.e., $d_K^v \in \mathbb{R}^{N \times 2}$). π_u denotes the weight of the u -th Cauchy kernel ($\sum_{u=1}^U \pi_u = 1$), and $\eta^{\text{MoC}} = \{\mu_1, \gamma_1, \dots, \mu_U, \gamma_U\}$ denotes the MoC parameters in which μ_u and γ_u are the location and scale of the u -th Cauchy kernel. Via the above optimization, we can use the optimized parameters η_*^{MoC} to model D_K as the characterized distribution (\hat{D}_K). Given \hat{D}_K , we aim to conduct the forward process from the ground-truth keypoints coordinates distribution D_0 , so that after K steps of forward diffusion, the generated distribution reaches \hat{D}_K . To this end, we modify Eq. (3) as follows:

$$\hat{d}_k = \sqrt{\bar{\alpha}_k} d_0 + (1 - \sqrt{\bar{\alpha}_k}) \mu^{\text{MoC}} + \sqrt{1 - \bar{\alpha}_k} \epsilon^{\text{MoC}} \quad (5)$$

where $\hat{d}_k \in \mathbb{R}^{N \times 2}$ represents a sample (i.e., a set of N keypoints coordinates) from the generated distribution \hat{D}_k , $\mu^{\text{MoC}} = \sum_{u=1}^U \mathbb{1}_u \mu_u$, and $\epsilon^{\text{MoC}} \sim \text{Cauchy}(\mathbf{0}, \sum_{u=1}^U (\mathbb{1}_u \gamma_u))$. Note that $\mathbb{1}_u$ is a zero-one indicator and $\sum_{u=1}^U \mathbb{1}_u = 1$ and $\text{Prob}(\mathbb{1}_u = 1) = \pi_u$.

From Eq. (5), we can observe that when K is sufficiently large and $\bar{\alpha}_K$ correspondingly decreases to nearly zero, the distribution of \hat{d}_K reaches the MoC distribution, i.e., $\hat{d}_K = \mu^{\text{MoC}} + \epsilon^{\text{MoC}} \sim \text{Cauchy}(\sum_{u=1}^U (\mathbb{1}_u \mu_u), \sum_{u=1}^U (\mathbb{1}_u \gamma_u))$. Af-

ter the above MoC-based forward process, we can use the generated $\{\hat{D}_k\}_{k=1}^{K-1}$ as supervision signals to train the diffusion model M_{diff} to learn the reverse process. More details about Eq. (5) can be found in Supplementary material. Such a forward process is only conducted to generate supervision signals for training the diffusion model, while we only need to conduct the reverse process during testing.

Reverse Process. In the reverse process, we aim to recover a desired determinate keypoints distribution D_0 from the initial distribution D_K . As discussed above, we characterize D_K via a MoC model and then generate $\{\hat{D}_k\}_{k=1}^{K-1}$ as supervision signals to optimize the diffusion model to learn to perform the reverse process ($\hat{D}_K \rightarrow \hat{D}_{K-1} \rightarrow \dots \rightarrow D_0$), in which the model iteratively reduces the noise and indeterminacy in \hat{D}_K to generate D_0 .

However, it can still be difficult to generate D_0 by directly performing the reverse process from \hat{D}_K , because the object appearance features are lacking in \hat{D}_K . Such features can help constrain the model reverse process based on the input image to get accurate predictions. Thus we further leverage the appearance features from the image as context to guide M_{diff} in the reverse process. Specifically, we reuse the features extracted from the keypoints distribution initializer as the appearance features f_{app} and feed f_{app} into the diffusion model, as shown in Fig. 3.

Our reverse process aims to generate a determinate distribution D_0 from the indeterminate distribution \hat{D}_K (during training) or D_K (during testing). Below we describe the reverse process during testing. We first obtain f_{app} from the input image. Then to help the diffusion model to learn to perform denoising at each reverse step, following [18, 52], we generate the unique step embedding f_D^k to inject the step number (k) information into the model. In this way, given a

set of noisy keypoints coordinates $d_k \in \mathbb{R}^{N \times 2}$ drawn from D_k at the k^{th} step, we use diffusion model M_{diff} , conditioned on the step embedding f_D^k and the object appearance features f_{app} , to recover d_{k-1} from d_k as:

$$d_{k-1} = M_{\text{diff}}(d_k, f_{\text{app}}, f_D^k) \quad (6)$$

3.3. Training and Testing

Training. Following [44], we first select N 3D keypoints from the surface of the object CAD model using the farthest point sampling (FPS) algorithm. Then we conduct the training process in the following two stages.

In the first stage, to initialize the distribution D_K , we optimize the keypoints distribution initializer. Specifically, for each training sample, given the pre-selected N 3D keypoints, we can obtain the ground-truth coordinates of the corresponding N 2D keypoints using the ground-truth 6D object pose. Then for each keypoint, based on the corresponding ground-truth coordinates, we generate a ground-truth heatmap following [42] for training the initializer. Thus for each training sample, we generate N ground-truth heatmaps. In this way, the loss function L_{init} for optimizing the initializer can be formulated as:

$$L_{\text{init}} = \left\| \mathbf{H}_{\text{pred}} - \mathbf{H}_{\text{GT}} \right\|_2^2 \quad (7)$$

where \mathbf{H}_{pred} and \mathbf{H}_{GT} denote the predicted heatmaps and ground-truth heatmaps, respectively.

In the second stage, we optimize the diffusion model M_{diff} . For each training sample, to optimize M_{diff} , we perform the following steps. (1) We first send the input image into an off-the-shelf object detector [57] and then feed the detected ROI into the trained initializer to obtain N heatmaps. Meanwhile, we can also obtain f_{app} . (2) We use the N predicted heatmaps to initialize D_K , and leverage the EM-type algorithm to characterize D_K as a MoC distribution \hat{D}_K . (3) Based on \hat{D}_K , we use the ground-truth keypoints coordinates d_0 to directly generate M sets of $(\hat{d}_1, \dots, \hat{d}_K)$ (i.e., $\{\hat{d}_1^i, \dots, \hat{d}_K^i\}_{i=1}^M$) via the forward process (Eq. (5)). (4) Then, we aim to optimize the diffusion model M_{diff} to recover \hat{d}_{k-1}^i from \hat{d}_k^i iteratively. Following previous diffusion works [18, 52], we formulate the loss L_{diff} for optimizing M_{diff} as follows ($\hat{d}_0^i = d_0$ for all i):

$$L_{\text{diff}} = \sum_{i=1}^M \sum_{k=1}^K \left\| M_{\text{diff}}(\hat{d}_k^i, f_{\text{app}}, f_D^k) - \hat{d}_{k-1}^i \right\|_2^2 \quad (8)$$

Testing. During testing, for each testing sample, by feeding the input image to the object detector and the keypoints distribution initializer consecutively, we can initialize D_K and meanwhile obtain f_{app} . Then, we perform the reverse process. During the reverse process, we sample M sets of noisy keypoints coordinates from D_K (i.e., $\{d_K^i\}_{i=1}^M$) and feed them into the trained diffusion model. Here we sample M sets of keypoints coordinates, because we are converting from a distribution (D_K) towards another distribution (D_0).

Then the model iteratively performs the reverse steps. After K reverse diffusion steps, we obtain M sets of predicted keypoints coordinates (i.e., $\{d_0^i\}_{i=1}^M$). To obtain the final keypoints coordinates prediction d_0 , we compute the mean of the M predictions. Finally, we can solve for the 6D object pose using a PnP solver, like [44, 53].

3.4. Model Architecture

Our framework mainly consists of the diffusion model (M_{diff}) and the keypoints distribution initializer.

Diffusion Model M_{diff} . As illustrated in Fig. 3, our proposed diffusion model M_{diff} mainly consists of a transformer encoder-decoder architecture. The appearance features f_{app} are sent into the encoder for extracting context information to aid the reverse process in the decoder. f_D^k and $\{d_k^i\}_{i=1}^M$ (or $\{\hat{d}_k^i\}_{i=1}^M$ during training) are sent into the decoder for the reverse process. Both the encoder and the decoder contain a stack of three transformer layers.

More specifically, as for the encoder part, we first map $f_{\text{app}} \in \mathbb{R}^{16 \times 16 \times 512}$ through a 1×1 convolution layer to a latent embedding $e_{\text{app}} \in \mathbb{R}^{16 \times 16 \times 128}$. To retain the spatial information, following [59], we further incorporate positional encodings into e_{app} . Afterwards, we flatten e_{app} into a feature sequence ($\mathbb{R}^{256 \times 128}$), and send it into the encoder. The encoder output f_{enc} containing the extracted object information will be sent into the decoder to aid the reverse process. Note that during testing, for each sample, we only need to conduct the above computation process once to obtain the corresponding f_{enc} .

The decoder part iteratively performs the reverse process. For notation simplicity, below we describe the reverse process for a single sample d_k instead of the M samples ($\{d_1^i, \dots, d_K^i\}_{i=1}^M$). Specifically, at the k -th reverse step, to inject the current step number (k) information into the decoder, we first generate the step embedding $f_D^k \in \mathbb{R}^{1 \times 128}$ using the sinusoidal function following [18, 52]. Meanwhile, we use an FC layer to map the input $d_k \in \mathbb{R}^{N \times 2}$ to a latent embedding $e_k \in \mathbb{R}^{N \times 128}$. Then we concatenate f_D^k and e_k along the first dimension, and send it into the decoder. By interacting with the encoder output f_{enc} (extracted object information) via cross-attention at each layer, the decoder produces f_{dec} , which is further mapped into the keypoints coordinates prediction $d_{k-1} \in \mathbb{R}^{N \times 2}$ via an FC layer. Then we send d_{k-1} back to the decoder as the input to perform the next reverse step.

Keypoints Distribution Initializer. The initializer adopts a ResNet-34 backbone, which is commonly used in 6D pose estimation methods [4, 53, 61]. To generate heatmaps to initialize the distribution D_K , we add two deconvolution layers followed by a 1×1 convolution layer after the ResNet-34 backbone, and then we obtain predicted heatmaps $\mathbf{H}_{\text{pred}} \in \mathbb{R}^{N \times \frac{H}{4} \times \frac{W}{4}}$ where H and W denote the height and width of the input ROI image respec-

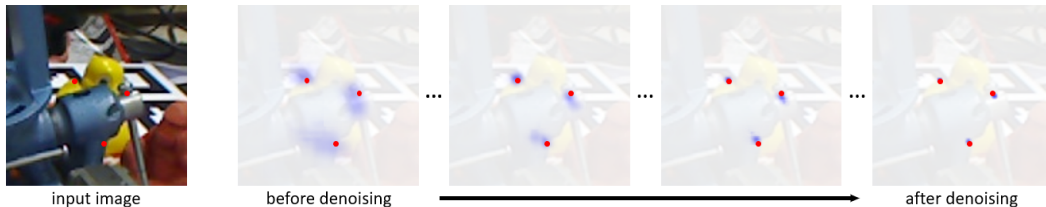


Figure 4. Visualization of the denoising process of a sample with our framework. In this example, the target object is the yellow duck and for clarity, we here show three keypoints only. The red dots indicate the ground-truth locations of these three keypoints. The noisy heatmap before denoising reflects that factors like occlusions and clutter in the scene can introduce noise and indeterminacy when detecting keypoints. As shown, our diffusion model can effectively and smoothly reduce the noise and indeterminacy in the initial distribution step by step, finally recovering a high-quality and determinate distribution of keypoints coordinates. (Better viewed in color)

Table 1. Comparisons with RGB-based 6D object pose estimation methods on the LM-O dataset. (*) denotes symmetric objects.

Method	PVNet [44]	HybridPose [51]	RePose [24]	DeepIM [33]	GDR-Net [61]	SO-Pose [8]	CRT-6D [4]	ZebraPose [53]	CheckerPose [35]	Ours
ape	15.8	20.9	31.1	59.2	46.8	48.4	53.4	57.9	58.3	60.6
can	63.3	75.3	80.0	63.5	90.8	85.8	92.0	95.0	95.7	97.9
cat	16.7	24.9	25.6	26.2	40.5	32.7	42.0	60.6	62.3	63.2
driller	65.7	70.2	73.1	55.6	82.6	77.4	81.4	94.8	93.7	96.6
duck	25.2	27.9	43.0	52.4	46.9	48.9	44.9	64.5	69.9	67.2
eggbox*	50.2	52.4	51.7	63.0	54.2	52.4	62.7	70.9	70.0	73.5
glue*	49.6	53.8	54.3	71.7	75.8	78.3	80.2	88.7	86.4	92.0
holepuncher	39.7	54.2	53.6	52.5	60.1	75.3	74.3	83.0	83.8	85.5
Mean	40.8	47.5	51.6	55.5	62.2	62.3	66.3	76.9	77.5	79.6

Table 2. Comparisons with RGB-based 6D object pose estimation methods on the YCB-V dataset. (-) indicates the corresponding result is not reported in the original paper.

Method	ADD(-S)	AUC of ADD-S	AUC of ADD(-S)
SegDriven[21]	39.0	-	-
SingleStage[22]	53.9	-	-
CosyPose [29]	-	89.8	84.5
RePose [24]	62.1	88.5	82.0
GDR-Net [61]	60.1	91.6	84.4
SO-Pose [8]	56.8	90.9	83.9
ZebraPose [53]	80.5	90.1	85.3
CheckerPose [35]	81.4	91.3	86.4
Ours	83.8	91.5	87.0

tively. Moreover, the features outputted by the ResNet-34 backbone, combined with features obtained from methods [35, 53], are used as the object features f_{app} .

4. Experiments

4.1. Datasets & Evaluation Metrics

Given that previous works [8, 24, 67] have reported the evaluation accuracy over 95% on the Linemod (LM) dataset [17], the performance on this dataset has become saturated. Thus recent works [4, 53] mainly focus on using the LM-O dataset [2] and the YCB-V dataset [63] that are more challenging, which we follow.

LM-O Dataset. The Linemod Occlusion (LM-O) dataset contains 1214 images and is a challenging subset of the LM dataset. In this dataset, around 8 objects are annotated on each image and the objects are often heavily occluded. Following [4, 53], we use both the real images from the LM dataset and the publicly available physically-based rendering (pbr) images [6] as the training images for LM-O. Following [53, 61], on LM-O dataset, we evaluate the model performance using the commonly-used ADD(-S) metric.

For this metric, we compute the mean distance between the model points transformed using the predicted pose and the same model points transformed using the ground-truth pose. For symmetric objects, following [63], the mean distance is computed based on the closest point distance. If the mean distance is less than 10% of the model diameter, the predicted pose is regarded as correct.

YCB-V Dataset. The YCB-V dataset is a large-scale dataset containing 21 objects and over 100k real images. The samples in this dataset often exhibit occlusions and cluttered backgrounds. Following [4, 53], we use both the real images from the training set of the YCB-V dataset and the publicly available pbr images as the training images for YCB-V. Following [53, 61], we evaluate the model performance using the following metrics: ADD(-S), AUC (Area Under the Curve) of ADD-S, and AUC of ADD(-S). For calculating AUC, we set the maximum distance threshold to 10 cm following [63].

4.2. Implementation Details

We conduct our experiments on an Nvidia V100 GPU. We set the number of pre-selected 3D keypoints N to 128. During training, following [34, 53], we utilize the dynamic zoom-in strategy to produce augmented ROI images. During testing, we use the detected bounding box with Faster RCNN [49] and FCOS [57] provided by CDPNv2 [34]. The cropped ROI image is resized to the shape of $3 \times 256 \times 256$ ($H = W = 256$). We characterize D_K via a MoC model with 9 Cauchy kernels ($U = 9$) for the forward diffusion process. We optimize the diffusion model M_{diff} for 1500 epochs using the Adam optimizer [28] with an initial learning rate of $4e-5$. Moreover, we set the number of sampled sets M to 5, and the number of diffusion steps K to

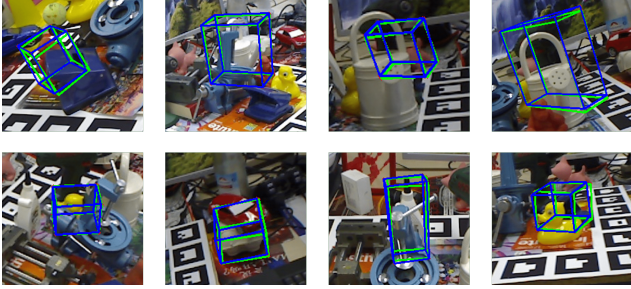


Figure 5. **Qualitative results.** Green bounding boxes represent the ground-truth poses and blue bounding boxes represent the predicted poses of our method. As shown, even facing severe occlusions, clutter in the scene or varying environment, our framework can still accurately recover the object poses, showing the effectiveness of our method for handling the noise and indeterminacy caused by various factors in object pose estimation.

100. Following [53], we use Progressive-X [1] as the PnP solver. Note that during testing, instead of performing the reverse process with all the K steps, we accelerate the process with DDIM [52], a recently proposed diffusion acceleration method. With DDIM acceleration, we only need to perform 10 steps to finish the reverse process during testing.

4.3. Comparison with State-of-the-art Methods

Results on LM-O Dataset. As shown in Tab. 1, compared to existing methods, our method achieves the best mean performance, showing the superiority of our method. We also show qualitative results on the LM-O dataset in Fig. 5. As shown, even in the presence of large occlusions (including self-occlusions) and cluttered backgrounds, our method still produces accurate predictions.

Results on YCB-V Dataset. As shown in Tab. 2, our framework achieves the best performance on both the ADD(-S) and the AUC of ADD(-S) metrics, and is comparable to the state-of-the-art method on the AUC of ADD-S metric, showing the effectiveness of our method.

4.4. Ablation Studies

We conduct extensive ablation experiments on the LM-O dataset, and we report the model performance on ADD(-S) metric averaged over all the objects.

Impact of denoising process. In our framework, we predict keypoints coordinates via performing the denoising process. To evaluate the efficacy of this process, we test three variants. In the first variant (*Variant A*), we remove the diffusion model

M_{diff} and predict keypoints coordinates directly from the heatmaps produced by the keypoints distribution initializer. The second variant (*Variant B*) has the same model architec-

Table 3. Evaluation on the effectiveness of the denoising process.

Method	ADD(-S)
Variant A	49.2
Variant B	57.3
Variant C	61.1
6D-Diff	79.6

ture as our framework, but the diffusion model is optimized to directly predict the coordinates instead of learning the reverse process. Same as *Variant B*, the third variant (*Variant C*) is also optimized to directly predict coordinates without denoising process. For *Variant C*, we stack our diffusion model structure multiple times to produce a deep network, which has similar computation complexity with our framework. As shown in Tab. 3, compared to our framework, the performance of these variants significantly drops, showing that the effectiveness of our framework mainly lies in the designed denoising process.

Impact of object appearance features f_{app} . In our framework, we send the appearance features f_{app} into the diffusion model M_{diff} to aid the reverse process. To evaluate its effect,

we test a variant in which we do not send f_{app} into M_{diff} (*w/o f_{app}*). As shown in Tab. 4, our framework performs better than this variant, showing that f_{app} can aid M_{diff} to get more accurate predictions.

Impact of MoC design. During training, we model the distribution D_K from the intermediate representation

(heatmaps) as a MoC distribution \hat{D}_K , and train the diffusion model M_{diff} to perform the reverse process from \hat{D}_K . To investigate the impact of this design, we evaluate two variants that train M_{diff} in different ways. In the first variant (*Standard diffusion w/o MoC*), we train the model to start the reverse process from the standard Gaussian noise, i.e., following the basic forward process in Eq. (3) for model training. In the second variant (*Heatmaps as condition*), we still train the model to start denoising from the random Gaussian noise but we use the heatmaps as the condition for the reverse process. As shown in Tab. 5, our framework consistently outperforms both variants, showing effectiveness of the designed MoC-based forward process.

Table 4. Evaluation on the effectiveness of the object appearance features f_{app} .

Method	ADD(-S)
w/o f_{app}	74.4
6D-Diff	79.6

Table 5. Evaluation on the effectiveness of the MoC design.

Method	ADD(-S)
Standard diffusion w/o MoC	73.1
Heatmaps as condition	76.2
6D-Diff	79.6

5. Conclusion

In this paper, we proposed a novel diffusion-based 6D object pose estimation framework, which effectively handles noise and indeterminacy in object pose estimation. In our framework, we formulate object keypoints detection as a carefully-designed reverse diffusion process. We design a novel MoC-based forward process to effectively utilize the distribution priors in intermediate representations. Our framework achieves superior performance.

Acknowledgement. This work was supported by the National Research Foundation Singapore under the AI Singapore Programme (Award Number: AISG-100E-2023-121).

References

- [1] Daniel Barath and Jiri Matas. Progressive-x: Efficient, anytime, multi-model fitting algorithm. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3780–3788, 2019. 8
- [2] Eric Brachmann, Frank Michel, Alexander Krull, Michael Ying Yang, Stefan Gumhold, et al. Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3364–3372, 2016. 7
- [3] Benjamin Busam, Marco Esposito, Simon Che’Rose, Nassir Navab, and Benjamin Frisch. A stereo vision approach for cooperative robotic movement therapy. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 127–135, 2015. 1
- [4] Pedro Castro and Tae-Kyun Kim. Crt-6d: Fast 6d object pose estimation with cascaded refinement transformers. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5746–5755, 2023. 1, 2, 3, 4, 6, 7
- [5] Bo Chen, Alvaro Parra, Jiewei Cao, Nan Li, and Tat-Jun Chin. End-to-end learnable geometric vision by backpropagating pnp optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8100–8109, 2020. 1, 2, 3, 4
- [6] Maximilian Denninger, Martin Sundermeyer, Dominik Winkelbauer, Youssef Zidan, Dmitry Olefir, Mohamad Elbadrawy, Ahsan Lodhi, and Harinandan Katam. Blenderproc. *arXiv preprint arXiv:1911.01911*, 2019. 7
- [7] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021. 1, 2, 3, 4
- [8] Yan Di, Fabian Manhardt, Gu Wang, Xiangyang Ji, Nassir Navab, and Federico Tombari. So-pose: Exploiting self-occlusion for direct 6d pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12396–12405, 2021. 1, 7
- [9] Lin Geng Foo, Hossein Rahmani, and Jun Liu. Aigc for various data modalities: A survey. *arXiv preprint arXiv:2308.14177*, 2023. 3
- [10] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE transactions on pattern analysis and machine intelligence*, 25(8):930–943, 2003. 2
- [11] Jia Gong, Lin Geng Foo, Zhipeng Fan, Qihong Ke, Hossein Rahmani, and Jun Liu. Diffpose: Toward more reliable 3d pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13041–13051, 2023. 1, 3, 4
- [12] Tianpei Gu, Guangyi Chen, Junlong Li, Chunze Lin, Yongming Rao, Jie Zhou, and Jiwen Lu. Stochastic trajectory prediction via motion indeterminacy diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17113–17122, 2022. 1, 3, 4
- [13] Shuxuan Guo, Yinlin Hu, Jose M Alvarez, and Mathieu Salzmann. Knowledge distillation for 6d pose estimation by aligning distributions of local predictions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18633–18642, 2023. 3
- [14] Yang Hai, Rui Song, Jiaojiao Li, and Yinlin Hu. Shape-constraint recurrent flow for 6d object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4831–4840, 2023.
- [15] Yang Hai, Rui Song, Jiaojiao Li, Mathieu Salzmann, and Yinlin Hu. Rigidity-aware detection for 6d object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8927–8936, 2023.
- [16] Rasmus Laurvig Haugaard and Anders Glent Buch. Surfemb: Dense and continuous correspondence distributions for object pose estimation with learnt surface embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6749–6758, 2022. 3
- [17] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary Bradski, Kurt Konolige, and Nassir Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *Computer Vision—ACCV 2012: 11th Asian Conference on Computer Vision, Daejeon, Korea, November 5-9, 2012, Revised Selected Papers, Part I 11*, pages 548–562. Springer, 2013. 7
- [18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, pages 6840–6851. Curran Associates, Inc., 2020. 1, 2, 3, 4, 5, 6
- [19] Tomas Hodan, Daniel Barath, and Jiri Matas. Epos: Estimating 6d pose of objects with symmetries. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11703–11712, 2020. 1, 3
- [20] Tsu-Ching Hsiao, Hao-Wei Chen, Hsuan-Kung Yang, and Chun-Yi Lee. Confronting ambiguity in 6d object pose estimation via score-based diffusion on se (3). *arXiv preprint arXiv:2305.15873*, 2023. 3
- [21] Yinlin Hu, Joachim Hugonot, Pascal Fua, and Mathieu Salzmann. Segmentation-driven 6d object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3385–3394, 2019. 4, 7
- [22] Yinlin Hu, Pascal Fua, Wei Wang, and Mathieu Salzmann. Single-stage 6d object pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2930–2939, 2020. 1, 3, 7
- [23] Shun Iwase, Xingyu Liu, Rawal Khirodkar, Rio Yokota, and Kris M. Kitani. Repose: Fast 6d object pose refinement via deep texture rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3303–3312, 2021. 3
- [24] Shun Iwase, Xingyu Liu, Rawal Khirodkar, Rio Yokota, and Kris M Kitani. Repose: Fast 6d object pose refinement via deep texture rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3303–3312, 2021. 7

- [25] Haobo Jiang, Mathieu Salzmann, Zheng Dang, Jin Xie, and Jian Yang. Se (3) diffusion model-based point cloud registration for robust 6d object pose estimation. *Advances in Neural Information Processing Systems*, 36, 2024. 3
- [26] Zakiah I. Kalantan and Jochen Einbeck. Quantile-based estimation of the finite cauchy mixture model. *Symmetry*, 11(9), 2019. 2, 4, 5
- [27] 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. 1, 3
- [28] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7
- [29] Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic. Cosypose: Consistent multi-view multi-object 6d pose estimation. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII 16*, pages 574–591. Springer, 2020. 4, 7
- [30] Junhyeok Lee, Junghwa Kang, Yoonho Nam, and TaeYoung Lee. Bias field correction in MRI with hampel noise denoising diffusion probabilistic model. In *Medical Imaging with Deep Learning, short paper track*, 2023. 3
- [31] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Ep n p: An accurate o (n) solution to the p n p problem. *International journal of computer vision*, 81:155–166, 2009. 2
- [32] Hongyang Li, Jiehong Lin, and Kui Jia. Dcl-net: Deep correspondence learning network for 6d pose estimation. In *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IX*, pages 369–385. Springer, 2022. 3
- [33] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. Deepim: Deep iterative matching for 6d pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 683–698, 2018. 3, 7
- [34] Zhigang Li, Gu Wang, and Xiangyang Ji. Cdpn: Coordinates-based disentangled pose network for real-time rgb-based 6-dof object pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7678–7687, 2019. 4, 7
- [35] Ruyi Lian and Haibin Ling. Checkerpose: Progressive dense keypoint localization for object pose estimation with graph neural network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14022–14033, 2023. 7
- [36] Xingyu Liu, Ruida Zhang, Chenyangguang Zhang, Bowen Fu, Jiwen Tang, Xiquan Liang, Jingyi Tang, Xiaotian Cheng, Yukang Zhang, Gu Wang, and Xiangyang Ji. Gdrnpp. https://github.com/shanice-1/gdrnpp_bop2022, 2022. 3
- [37] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11461–11471, 2022. 3
- [38] Fabian Manhardt, Wadim Kehl, Nassir Navab, and Federico Tombari. Deep model-based 6d pose refinement in rgb. In *The European Conference on Computer Vision (ECCV)*, 2018. 3
- [39] Eric Marchand, Hideaki Uchiyama, and Fabien Spindler. Pose estimation for augmented reality: a hands-on survey. *IEEE transactions on visualization and computer graphics*, 22(12):2633–2651, 2015. 1
- [40] Jianhan Mei, Xudong Jiang, and Henghui Ding. Spatial feature mapping for 6dof object pose estimation. *Pattern Recognition*, 131:108835, 2022. 1
- [41] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2021. 1, 3
- [42] Markus Oberweger, Mahdi Rad, and Vincent Lepetit. Making deep heatmaps robust to partial occlusions for 3d object pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 119–134, 2018. 3, 6
- [43] Kiru Park, Timothy Patten, and Markus Vincze. Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7668–7677, 2019. 3
- [44] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. Pvnnet: Pixel-wise voting network for 6dof pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4561–4570, 2019. 1, 2, 3, 4, 6, 7
- [45] Luis Pérez, Íñigo Rodríguez, Nuria Rodríguez, Rubén Usamentiaga, and Daniel F García. Robot guidance using machine vision techniques in industrial environments: A comparative review. *Sensors*, 16(3):335, 2016. 1
- [46] Mahdi Rad and Vincent Lepetit. Bb8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth. In *Proceedings of the IEEE international conference on computer vision*, pages 3828–3836, 2017. 3
- [47] Jason Raphael Rambach, Alain Pagani, Michael Schneider, Oleksandr Artemenko, and Didier Stricker. 6dof object tracking based on 3d scans for augmented reality remote live support. *Comput.*, 7:6, 2018. 1
- [48] Hong Ren, Lin Lin, Yanjie Wang, and Xin Dong. Robust 6-dof pose estimation under hybrid constraints. *Sensors*, 22(22):8758, 2022. 3
- [49] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 4, 7
- [50] 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. 2, 3
- [51] Chen Song, Jiaru Song, and Qixing Huang. Hybridpose: 6d object pose estimation under hybrid representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 431–440, 2020. 7

- [52] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. 1, 2, 3, 4, 5, 6, 8
- [53] Yongzhi Su, Mahdi Saleh, Torben Fetzner, Jason Rambach, Nassir Navab, Benjamin Busam, Didier Stricker, and Federico Tombari. ZebraPose: Coarse to fine surface encoding for 6dof object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6738–6748, 2022. 1, 3, 4, 6, 7, 8
- [54] Martin Sundermeyer, Zoltán-Csaba Márton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3d orientation learning for 6d object detection from rgb images. In *European Conference on Computer Vision*, 2018. 3
- [55] Mahdi Teimouri. Statistical inference for mixture of cauchy distributions. *arXiv preprint arXiv:1809.05722*, 2018. 5
- [56] Bugra Tekin, Sudipta N Sinha, and Pascal Fua. Real-time seamless single shot 6d object pose prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 292–301, 2018. 3
- [57] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9627–9636, 2019. 6, 7
- [58] Julen Urain, Niklas Funk, Jan Peters, and Georgia Chalvatzaki. Se(3)-diffusionfields: Learning smooth cost functions for joint grasp and motion optimization through diffusion. *IEEE International Conference on Robotics and Automation (ICRA)*, 2023. 3
- [59] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 6
- [60] Gu Wang, Fabian Manhardt, Xingyu Liu, Xiangyang Ji, and Federico Tombari. Occlusion-aware self-supervised monocular 6d object pose estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 1
- [61] Gu Wang, Fabian Manhardt, Federico Tombari, and Xiangyang Ji. Gdr-net: Geometry-guided direct regression network for monocular 6d object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16611–16621, 2021. 1, 3, 4, 6, 7
- [62] Di Wu, Zhaoyong Zhuang, Canqun Xiang, Wenbin Zou, and Xia Li. 6d-vnet: End-to-end 6-dof vehicle pose estimation from monocular rgb images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 1
- [63] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. 2018. 1, 3, 7
- [64] Yan Xu, Kwan-Yee Lin, Guofeng Zhang, Xiaogang Wang, and Hongsheng Li. Rnnpose: Recurrent 6-dof object pose refinement with robust correspondence field estimation and pose optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 1, 3
- [65] Heng Yang and Marco Pavone. Object pose estimation with statistical guarantees: Conformal keypoint detection and geometric uncertainty propagation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8947–8958, 2023.
- [66] Jun Yang, Wenjie Xue, Sahar Ghavidel, and Steven L Waslander. 6d pose estimation for textureless objects on rgb frames using multi-view optimization. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2905–2912. IEEE, 2023.
- [67] Sergey Zakharov, Ivan S. Shugurov, and Slobodan Ilic. Dpod: 6d pose object detector and refiner. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1941–1950, 2019. 3, 7