

# Environment-Agnostic Pose: Generating Environment-independent Object Representations for 6D Pose Estimation

Shaobo Zhang<sup>1</sup>Yuhang Huang<sup>2</sup>Wanqing Zhao<sup>1\*</sup>Wei Zhao<sup>3</sup>Ziyu Guan<sup>1</sup>Jinye Peng<sup>1\*</sup><sup>1</sup>Northwest University<sup>2</sup>National University of Defense Technology<sup>3</sup>Xidian University

{zhangshaobo, zhaowq, ziyuguan, pjy}@nwu.edu.cn, huangai@nudt.edu.cn, ywzhao@mail.xidian.edu.cn

## Abstract

This paper introduces EA6D, a novel diffusion-based framework for 6D pose estimation that operates effectively in any environment. Traditional pose estimation methods struggle with the variability and complexity of real-world scenarios, often leading to overfitting on controlled datasets and poor generalization to new scenes. To address these challenges, we propose a generative pose estimation paradigm that generates environment-independent object representations for pose estimation, which are robust to environmental variations such as illumination, occlusion, and background clutter. Specifically, we propose the novel Environment Decoupling Diffusion Model (EDDM) which separates object representations from environmental factors while enabling efficient few-step sampling by leveraging input image priors instead of pure noise initialization. We validate our approach on four standard benchmarks and a self-made dataset *DiverseScenes*. The results demonstrate that EA6D, trained using only synthetic data, can outperform the state-of-the-art methods with both synthetic and realistic data. In particular, for fair comparisons with synthetic data, we can exceed the previous SOTA by 18.1% and 33.5% on *Linemod* and *Linemod-Occluded* datasets respectively. Project page: <https://github.com/acmff22/EA6D>

## 1. Introduction

6D Pose Estimation is a pivotal task in the field of computer vision and robotics, which aims to determine the exact position and orientation of an object in three-dimensional space relative to a camera or a coordinate system. The ability to accurately perceive an object’s pose is fundamental for a wide range of applications, including robotic manipulation, augmented reality and autonomous navigation.

Recently, a series of deep-learning-based object 6D pose estimation methods [17, 22, 23, 28, 31, 35, 41, 44] have

\*Corresponding authors

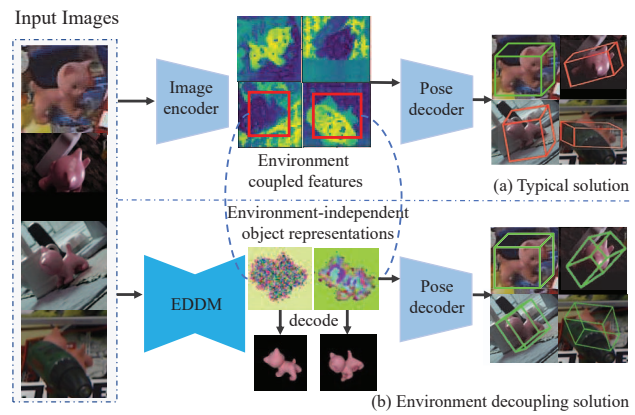


Figure 1. Different from typical RGB-based pose estimation frameworks, which are easily influenced by environmental factors leading to inaccurate pose estimation, we propose a generative paradigm that leverages diffusion models to generate environment-independent object representations for pose estimation.

been proposed. These methods achieve state-of-the-art results, but require large annotated datasets. Generating accurate annotations for pose data is a labor-intensive and time-consuming process, leading most deep-learning-based 6D pose estimation methods to be trained and tested on closed datasets. These datasets [3, 10, 12, 16] typically contain only a few hundred images with limited variation and controlled factors, such as object arrangement, lighting conditions, and background, resulting in overfitting to the training set and the struggle to generalize to new scenes.

Consequently, synthetic datasets have been employed as an additional complement to real datasets, or alternative to real datasets, as they can be easily and automatically annotated in simulated environments. The primary challenge with synthetic data is the domain discrepancy between real and synthetic datasets, which often leads to reduced accuracy when applied to real-world scenarios. To alleviate this problem, some self-supervised methods [20, 37, 39, 40, 46] are proposed, which leverage synthetic images with pose annotations and unannotated real images during training.

These methods harness intrinsic structures to generate supervision signals and implement constraints on the discrepancies between real and synthetic images, thereby mitigating the domain gap.

Practical applications such as AR and autonomous driving, which need to operate in uncontrolled environments, make 6D pose estimation more challenging. Here, object appearances vary due to fluctuating environmental conditions, background clutter, and occlusions, which often exceed the variability present in training datasets. Previous methods usually use CNNs as image encoder to extract features for pose estimation (as shown on the top side of Figure 1). These features are highly sensitive to environmental factors. When the environment changes, CNNs struggle to accurately extract object features. This hinders the model’s ability to generalize to new scenes, potentially diminishing the accuracy of pose estimation and even leading to object pose estimation failure. While self-supervised methods are proficient at leveraging unlabeled data, they may not fully account for the extensive real-world variability found in these uncontrolled environments, leading to models that are not sufficiently robust for novel conditions. Therefore, there is an urgent need for approaches that can adapt to a variety of dynamic and diverse environments, ensuring reliable pose estimation capabilities.

In this paper, we introduce a novel paradigm for pose estimation in uncertain environments, including unseen scenes, varying lighting, occlusion, and other possible situations. We consider a generative strategy that generates environment-independent object representations for pose estimation (bottom side of Figure 1). Diffusion models [7, 11, 34] have recently achieved promising results in various generative tasks, including image generating and editing. These models recover high-quality, deterministic samples from noisy and some conditions through a progressive denoising procedure. Motivated by the above, we design an **Environment-Agnostic** object **6D POSE** estimation framework (named EA6D) that exploits the diffusion model to generate environment-independent representations in latent space for pose estimation, while input images serve as conditional cues, illustrated on the bottom side of Figure 1. Specifically, we propose an Environment Decoupling Diffusion Model (EDDM) which employs a novel environment decoupling diffusion process to intrinsically decouple the environment-independent object representations from input images. By performing validation on four popular benchmarks and a self-made dataset, the results show that our method significantly outperforms state-of-the-art self-supervised methods and provides a significant improvement in generalization to unseen scenarios.

In sum, the main contributions in this paper include:

- We propose a new paradigm for environment-agnostic object 6D pose estimation, by generating environment-

independent object representations rather than estimating poses directly from input images.

- We design a novel Environment Decoupling Diffusion Model which employs an explicit environmental decoupling mechanism via gradient modelling and conditional generation to separate object representations from environment, while enabling efficient several-step sampling.
- We conduct extensive experiments on four public datasets including Linemod, Linemod-Occluded, YCB-V, HomebrewedDB and a self-made dataset DiverseScenes, showing that our method surpasses current state-of-the-art self-supervised methods. Our method exhibits remarkable generalization capabilities, even in previously unseen scenes, which is crucial for practical applications.

## 2. Related work

**Fully-Supervised 6D Object Pose Estimation.** Early approaches rely on local or global features to establish correspondences between keypoints on CAD models [1, 3, 4]. More recently, learning-based methods, particularly those utilizing Convolutional Neural Networks (CNNs), have become dominant. These methods extract deep features and establish mappings from RGB images to 6D object pose [14, 17, 41, 44]. Due to the inherent nonlinearity of the rotation space, the accuracy of directly regressed object pose is often limited [21]. To enhance accuracy, many methods employ CNNs to establish 2D-3D correspondences, followed by solving the object pose using the Perspective-n-Point (PnP) algorithm [19]. These methods can be categorized into sparse and dense correspondence methods. Sparse methods [28, 33, 38, 45] predict the 2D projections of predefined 3D keypoints, while dense methods [22, 23, 27, 35, 41, 48] predict 3D coordinates on the object model corresponding to image pixels thus providing better robustness to occlusions and clutter. However, fully-supervised 6D pose estimation methods are constrained by the need for accurate and diverse training data, which is often scarce due to the labor-intensive nature of precise annotation and the limited variation in available datasets, resulting in poor generalization to real-world and uncontrolled environments.

**Self-Supervised 6D Object Pose Estimation.** Considering the tremendous effort to collect large amounts of annotations for 6D object pose, some methods [26, 29, 36] have turned to using synthetic images for training. However, because of the domain gap between real and synthetic images, the performances of these methods are significantly lower compared to the methods that use real images with object poses for training. To narrow the domain gap, a few recent self-supervised methods [5, 20, 37, 39, 40, 43, 46, 49] have begun to incorporate both synthetic images with known object poses and unannotated real images into their training datasets. However, in diverse real-world application sce-

narios, models must contend with a variety of backgrounds, lighting conditions, and other environmental factors that are difficult to be adequately included in the training dataset. These environmental factors can significantly alter the appearance of objects within images. As a result, features learned by the model for the same object may vary across different scenes, leading to diminished generalization capabilities when encountering new scenes. Different from self-supervised learning methods, our proposed method directly generates environment-independent object representations to fundamentally eliminate the impact of environmental disturbances.

**Diffusion-based Models for Vision Tasks.** Originally introduced for image generation, diffusion models [7, 11, 34] have demonstrated impressive generation capabilities and have been explored in various tasks, including edge detection [47], image inpainting [25], and 6D pose estimation [9, 42, 45]. DiffusionEdge [47] presents a novel diffusion-based edge detector capable of generating precise and sharp edge maps without post-processing. 6D-diff [45] formulates object keypoints detection as a reverse diffusion process and employs a MoC-based forward process to leverage distribution priors in intermediate representations. Method [42] introduces three novel aggregation networks designed to effectively aggregate diffusion features, showing superior generalizability for object pose estimation. Currently, no existing work has yet explored the potential of diffusion models to generate environment-independent object representations, which could significantly improve the robustness of pose estimation in diverse environments.

### 3. Method

#### 3.1. Overview

Traditional pose estimation methods typically estimate pose directly from images. Environmental factors such as background clutter, illumination variations, and occlusion in images can limit the accuracy and robustness of pose estimation. Pursuing a different paradigm, we propose a diffusion-based method that generates environment-independent object representations for pose estimation. Traditional diffusion models are predominantly utilized for image generation tasks, where they excel at creating new synthetic images. These models operate on the principle of gradually adding noise to an input image and then learning to reverse this process, effectively denoising the image over multiple iterations to recover the original content. In this paper, we extend the concept of diffusion models to a completely new domain by recovering environment-independent object representations for pose estimation.

The overall framework of the proposed method EA6D is shown in Figure 2. The first stage is environment-independent object representation encoding. A synthesized

pure object image with the same pose as the input image, but free from background, lighting, and occlusion, is encoded into a latent space using a Variational Autoencoder (VAE) [18]. This encoding results in an object representation  $g_0 \in R^{64 \times 64 \times 3}$ , which can be decoded back to the original pure object image by the decoder of VAE network, demonstrating that the representation contains only information about the object independent of the environment. Following this, a forward diffusion process is applied to  $g_0$  to generate a noisy representation  $g_t$ , which can be split into two sub-processes: the clean object representation  $g_0$  to input image latent representation  $r_0$ , meanwhile Gaussian noise is incrementally added. To denoise and recover the environment-independent object representations from  $g_t$ , an Environment Decoupling Diffusion Model (EDDM) is introduced. A conditional encoder extracts multi-level features from the input images. These features are then fused with the features of the U-Net encoder and are input to U-Net decoders. The EDDM outputs a decoupled object representation, which potentially contains object pose information, independent of environmental factors. Finally, a pose decoder predicts the pose information, including rotation quaternion and translation vector. The inference process in EA6D is thus a combination of diffusion-based denoising and pose decoding, enabling robust and accurate pose estimation in diverse and complex environments.

#### 3.2. Environment Decoupling Diffusion for Generating Object Representations

Current 6D pose estimation methods often face robustness challenges in dynamic or unseen environments, as their object representations are influenced by complex environmental variations, resulting in multimodal distributions of features. CNNs-based methods typically use a single model to fit the entire data distribution. When facing multimodal distributions, the single model struggles to capture all the different patterns simultaneously, thereby resulting in a notable decrease in performance in new environments. To avoid the negative impact, we propose to extract environment-independent representations, enabling more accurate and generalizable pose estimation. Recent diffusion models show remarkable capabilities for modeling complex multimodal distribution, which is helpful for learning about environment-independent representations. Therefore, we propose to construct our framework based on the diffusion models to cover the complex distribution of object representations. More importantly, to acquire the environment-independent component from the input image distribution, we propose a novel environment decoupling diffusion model for generating environment-independent object representations inspired by the decoupled diffusion models [15, 47]. Different from the vanilla decoupled diffusion process that focuses on splitting the image and the

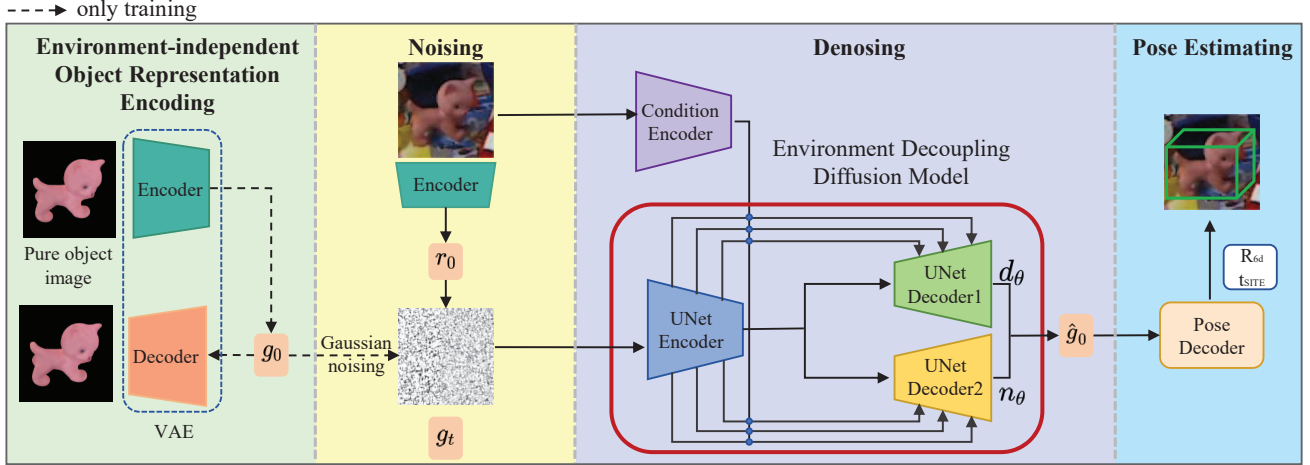


Figure 2. Overview of the proposed method. In training phase, the proposed method involves training a Variational Autoencoder (VAE) to encode pure object images into environment-independent representations. These representations are then sampled to create noisy representations, which are processed by an Environment Decoupling Diffusion Model (EDDM). The EDDM uses a dual structure with one UNet encoder and two UNet decoders to decouple the environment-independent object representations from the input image with complex environments. The pose decoder is trained to predict the pose from the representations, enabling robust pose estimation across various environments. In testing phase, a random Gaussian noise is fed into the EDDM without VAE.

noise components, we aim to decouple the environment-independent object representations from the input image with complex environments. To this end, we introduce the improved forward and reversed diffusion processes.

### 3.2.1. Forward Process

The vanilla decoupled diffusion process reformulates the conventional image-to-noise process into two stages: image-to-zero and zero-to-noise [15]. Corresponding to current object representation generation, there will be a clean object representation to zero process and a zero to noise process. However, such a paradigm only describes how to learn the environment-independent object representation from noise, without a clue on how to decouple it from the input images with complex environments. To address this problem, this paper proposes an environment decoupling diffusion process to intrinsically decouple the environment-independent object representation from the input image. We follow [30] to conduct the diffusion process in the latent space, i.e., our goal is to obtain the environment-independent object latent representations from the input image that mixes the object with the complex environments:

$$q(g_t|g_0) = \mathcal{N}(g_t; g_0 + \int_0^t (r_0 - g_0)dt, tI), \quad (1)$$

where  $g_0$  and  $g_t$  denote the clean and noisy object latent representations respectively,  $t \in [0, 1]$  is the time step,  $r_0$  denotes the latent representation of input image. (A Variational Autoencoder features the latent representation.) According to the above equation, we can sample  $g_t$  by  $g_t = g_0 + \int_0^t (r_0 - g_0)dt + \sqrt{t}n$ , where  $n \in \mathcal{N}(0, I)$ . Ob-

viously, the forward process describes two sub-processes:  $g_0 + \int_0^t (r_0 - g_0)dt$  is the clean object representation to input image latent representation and  $\sqrt{t}n$  is the zero to noise. It is worth noting that the first term contains the decay gradient from the input image to the environment-independent object representation, i.e.,  $r_0 - g_0$ . We call this term the environment decoupling gradient, which guides the generation from the input image representation to object representation. The environment decoupling gradient provides explicit evidence for our goal of decoupling the environment-independent object representations, allowing for accurate and efficient generation.

### 3.2.2. Reversed Process

The goal of the reversed process is to gradually recover the clean object representation  $g_0$  from the noisy input  $g_1$ . Following the derivation of [15], we can formally represent the reversed process as a conditional probability:

$$q(g_{t-\Delta t}|g_t, g_0) = \mathcal{N}(g_{t-\Delta t}; \int_t^{t-\Delta t} (r_0 - g_0)dt - \frac{\Delta t}{\sqrt{t}}n, \frac{\Delta t(t - \Delta t)}{t}I), \quad (2)$$

where  $\Delta t$  is the smallest time step. Inheriting DDM, the reversed process benefits from the analytic gradient  $r_0 - g_0$ , enabling approximate analytic solution for few-step generation (even with only one step). Different from general diffusion models that recover the goal content from a normal distribution, our reversed process aims to generate object representations from the combination of the latent image representations and the noise. We can easily obtain the ini-

tial variable in the reversed process according to the forward process, i.e.,  $g_1 = r_0 + n$ . Intuitively, the generation process starts with the mixture of  $r_0$  and  $n$  instead of pure noise, which provides important prior information, thus speeding up the inference procedure and improving the generation quality. The reversed conditional probability  $q(g_{t-\Delta t}|g_t, g_0)$  includes  $r_0 - g_0$  and  $n$ , which are unknown in the reversed process. Therefore, we need to use a neural network  $\theta$  to parameterize  $q(g_{t-\Delta t}|g_t, g_0)$ . Practically, the neural network has two outputs:  $d_\theta$  and  $n_\theta$ , corresponding to the two unknown terms.

### 3.2.3. Network Architecture

As shown in Figure 2, we construct our network based on a U-Net architecture including one encoder and two decoders. A pre-trained Swin Transformer [24] serves as a conditional encoder, extracting multi-level features from the input images. These features are then fused with the features of the U-Net encoder at corresponding levels through attention mechanisms, acting as inputs for the U-Net decoders. In practice, the input images will be obtained by an off-the-shelf object detector such as Yolo-v4 [2].

### 3.3. Pose Decoder for Object Representations

The pose decoder is designed to predict the 6D pose, including both the rotation and translation, from the object representations generated from EDDM. The decoder consists of three modules: a ResNet block to further extract the features of  $\hat{g}_0^t$ , a rotation head for rotation estimation, and a translation head for translation prediction. The ResNet block enhances feature extraction through three convolutional layers. The first layer employs a  $1 \times 1$  kernel to expand the feature dimension, capturing richer information. The second layer uses a  $3 \times 3$  kernel for spatial feature reduction. The third layer also uses a  $1 \times 1$  kernel. The rotation head is designed with a series of  $3 \times 3$  convolutional layers, followed by a generalized average pooling layer and a fully connected layer, culminating in the prediction of the rotation quaternion  $R$ .

For 3D translation, since directly regressing the translation in 3D space does not work well in practice, we adopt Scale-invariant Translation Estimation (SITE) [22] to decouple the translation into the 2D location  $(o_x, o_y)$  of the projected 3D centroid and the object’s distance  $T_z$  towards the camera. The translation head is constituted by a series of convolutional layers followed by fully connected layers, which predict the scale-invariant translation parameters  $T_{site} = (\Delta x, \Delta y, \Delta z)$ . These parameters are then utilized to compute the object’s 2D coordinates and depth, as delineated by the equations:

$$\begin{cases} o_x = \Delta x \cdot w + c_x \\ o_y = \Delta y \cdot h + c_y \\ T_z = \Delta z \cdot r \end{cases} \quad (3)$$

where  $w$  and  $h$  represent the width and height of the bounding box,  $c_x$  and  $c_y$  denote the coordinates of the box’s center, and  $r$  is the scaling factor derived from the image’s dimensions. Given the camera intrinsics  $k$ , the translation  $T$  can be calculated via back-projection  $T = K^{-1}T_z[o_x, o_y, 1]^T$ . This method allows for a more accurate and robust estimation of the object’s 3D translation, thereby enhancing the overall pose estimation accuracy.

### 3.4. Training Procedure

Our training object is meticulously designed to refine the performance of the VAE, EDDM, and pose decoder within our framework. Initially, the VAE is trained to encode synthetic images, which are environment-free and rendered with poses corresponding to the input images, into latent features. These latent features are then decoded to reconstruct the original synthesized images, ensuring the network’s ability to capture the essential features of the object.

After the VAE’s training, we proceed to train the EDDM while keeping the VAE’s weights fixed. EDDM is trained following the training strategy and objective of DDM [15]:

$$L_{eddm} = \|d_\theta - (r_0 - g_0)\|_2 + \|n_\theta - n\|_2 \quad (4)$$

where the first term and second term correspond to the supervision of predicting noise component and image component, respectively.

At last, we freeze VAE and EDDM to train the pose decoder. We employ a disentangled 6D pose loss via individually supervising the rotation  $R$ , the scale-invariant 2D object center  $(\Delta x, \Delta y)$  and the distance  $\Delta z$ :

$$L_{pose} = L_R + L_{center} + L_z \quad (5)$$

Thereby,

$$\begin{cases} L_R & = & \|R - \bar{R}\|_2 \\ L_{center} & = & \|\Delta x - \Delta \bar{x}\|_2 + \|\Delta y - \Delta \bar{y}\|_2 \\ L_z & = & \|\Delta z - \Delta \bar{z}\|_2 \end{cases} \quad (6)$$

where  $(\Delta x, \Delta y, \Delta z, R)$  and  $(\Delta \bar{x}, \Delta \bar{y}, \Delta \bar{z}, \bar{R})$  denote prediction and ground truth, respectively.

In the inference phase, object images from different environments are fed into the conditional encoder as cue information for the EDDM. The EDDM systematically eliminates environmental factors from the noisy latent space, yielding a representation that encapsulates solely the object’s information. This purified representation is then fed into the pose decoder to extract the object pose, thereby achieving robust generalization across various scenes and conditions.

## 4. Experiments

In this section, we conduct experiments to verify the generalization capability of the proposed method in various environments. Therefore we follow existing works [5, 37] in

self-supervised object pose estimation which focus on narrowing the domain gap and enhancing the network’s generalization ability across different domains. We compare our method to the state-of-the-art methods on four of the core datasets of the BOP challenge [13] and a self-made dataset. We also provide a detailed analysis of our proposed scheme in ablation study.

#### 4.1. Datasets

We evaluate our EA6D on four of the core BOP datasets [13], Linemod(LM) [10], Linemod-Occluded(LM-O) [3], HomebrewedDB(HB) [16] and YCB-V [44]. LM, LM-O, and HB include the same objects under different environments, and YCB-Video involves environmental changes, making them suitable for verifying the proposed method. HB is typically used to study the generalization ability of methods towards new scenes with changing illumination. However, the images in HB are from a single indoor environment. To expand the testing of unseen scenarios, the DiverseScenes Dataset is created by simulating both indoor and outdoor environments, to rigorously evaluate the model’s ability to generalize across varied environments.

**DiverseScenes Dataset.** We select four distinct real-world environments, spanning both indoor and outdoor settings, and employ Physically Based Rendering (PBR) techniques to render the 3D models into these backgrounds. To capture the inherent diversity of real-world lighting, we apply a range of lighting conditions, thus amplifying the disparity with the training data and crafting previously unseen test scenarios. For each object category, we render 150 images per environment, amassing a total of 1950 images. Figure 3 illustrates some samples of the dataset. We give more samples of the dataset in the supplementary material and will release the dataset.



Figure 3. We simulated different indoor and outdoor backgrounds with varying lighting to construct four distinct scenarios, which have a significant difference from the training data.

It should be emphasized that EA6D does not require any real images for training. For the evaluation of EA6D, all images in the above dataset are used for testing only.

#### 4.2. Metrics

We report our results referring to the common Average Distance of Distinguishable Model Points (ADD) metric [10]. This metric reports 6D pose error by transforming all object vertices with estimated pose and ground-truth pose and measuring the average distance of the two sets of points. The estimated pose is considered be correct if the average distance is smaller than 10% of the object’s diameter. For symmetric objects, we use the ADD-S metric [44], where the mean distance is computed based on the closest point distance. For YCB-V, we report the AUC (area under curve) of ADD(-S) with a maximum threshold of 10 cm [44].

#### 4.3. Implementation Details

Our method is implemented by Pytorch, and all experiments are conducted on an NVIDIA 4090 GPU. We use Synthetic PBR Dataset [13] to train the 2D object detector Yolov4 [2] and the entire network. For each image, we utilize the object’s CAD model to render it in the same pose onto a blank image to obtain the pure object image. We use pure object images to train VAE and use them as ground-truth for corresponding input PBR images to train EDDM from scratch. We train VAE using AdamW optimizer with an attenuated learning rate (from  $1e^{-4}$  to  $5e^{-7}$ ). We train the diffusion models using AdamW optimizer with an attenuated learning rate (from  $2e^{-4}$  to  $5e^{-6}$ ) for 20k iterations with a batch size of 12 and each training takes up about 30 GPU hours. The pose decoder is trained with the same optimizer setting for 10k iterations with a batch size of 24 and each training takes about 10 GPU hours. The conditional encoder is pre-trained on ImageNet [6].

#### 4.4. Comparative Evaluation

Here, we evaluate the proposed EA6D on the LM, LM-O and YCB-V in comparison to some state-of-the-art methods including six fully-supervised methods that are trained with annotated real images with object poses (DPOD [48], PVNet [28], SO-Pose [8]), ZebraPose [35], CheckerPose [23] and 6D-diff [45], two self-supervised methods that are trained with only synthetic data (AAE [36], MHP [26]), three self-supervised methods that are trained with both synthetic data and unannotated real images (DSC-PoseNet [46], SMOC-Net [37], TexPose [5]), and three self-supervised methods that are trained with synthetic data, unannotated real images and depth images (Self6D [39], Self6D++ [40], RKHSPose [43]). In addition, for a more comprehensive comparison, we also compare our method with Self6D++ [40] and RKHSPose [43] trained only with synthetic data. Table 1 reports the average metrics on each dataset and we provide detailed results in the supplementary material.

**Comparisons on Linemod.** As seen from Table 1, there is a significant performance gap between fully-supervised

Table 1. Comparison with state-of-the-art methods on LM, LM-O and YCB-V. The table reports results for the Average Recall (%) of ADD(-S) on LM and LM-O, and AUC of ADD-S and ADD(-S) on YCB-V. All results except ours are copied from SMOC-Net [37], TexPose [5] and RKHSPose [43].  $R$ : annotated real RGB data.  $S$ : synthetic RGB data.  $R^-$ : unannotated real RGB data.  $D$ : depth data. The best pose method using the same kind of training data is underlined, and the overall best method is marked in bold. (-) denotes results missing from the original paper.

Methods	Training data	LM	LM-O	YCB-V	
		ADD(-S)	ADD(-S)	ADD-S AUC	ADD(-S) AUC
DPOD [48]	$R + S$	82.6	47.3	-	-
PVNet [28]		86.3	40.8	-	73.8
SO-Pose [8]		<u>96.0</u>	62.3	90.9	83.9
ZebraPose [35]		-	76.9	90.1	85.3
CheckerPose [23]		-	77.5	91.3	86.4
6D-diff [45]		-	<u>79.6</u>	<u>91.5</u>	<u>87.0</u>
Self6D [39]	$S+R^-+D$	58.9	32.1	-	-
Self6D++ [40]		88.5	64.7	91.4	80.0
RKHSPose [43]		<u>95.9</u>	<u>68.7</u>	<u>92.6</u>	<u>83.0</u>
DSC-PoseNet [46]	$S+R^-$	54.7	21.9	-	-
Self6D++ [40]		76.0	59.8	-	-
SMOC-Net [37]		91.3	63.3	-	-
TexPose [5]		<u>91.7</u>	<u>66.7</u>	-	-
AAE [36]	$S$	31.4	-	-	-
MHP [26]		38.8	-	-	-
Self6D++ [40]		77.4	52.9	89.4	77.8
RKHSPose [43]		78.2	54.3	90.2	76.5
EA6D		<b>96.3</b>	<b>87.8</b>	<b>93.1</b>	<b>88.3</b>

methods and approaches trained with synthetic data instead. Self-supervised approaches have proven to be able to almost completely close the domain gap, yet typically require additional modalities (unannotated real RGB data, depth data) to achieve comparable results. The proposed EA6D achieves a better performance than all the comparative methods (regardless of whether these methods are trained in either fully-supervised or self-supervised manner), and only uses synthetic images for training. This is attributed to EA6D generating environment-independent object representations for pose estimation rather than estimating poses directly from input images, avoiding the impact of domain gap on network performance.

**Comparisons on Linemod-Occluded.** In the inference stage, each object in an image will be inferred independently. The corresponding results are reported in Table 1 which shows that EA6D outperforms all these methods, demonstrating strong robustness against occlusion. Note that EA6D only uses synthetic images as training data and still outperforms fully supervised methods (6D-diff by 8.3%, CheckerPose by 10.3%, ZebraPose by 10.9%).

**Comparisons on YCB-V.** As shown in Table 1, EA6D achieves the best performance on both the AUC of ADD-S and the AUC of ADD(-S) metrics, showing that EA6D is capable of handling a wide variety of object appearances and lighting conditions.

**Comparisons on Unseen Environments.** In this experi-

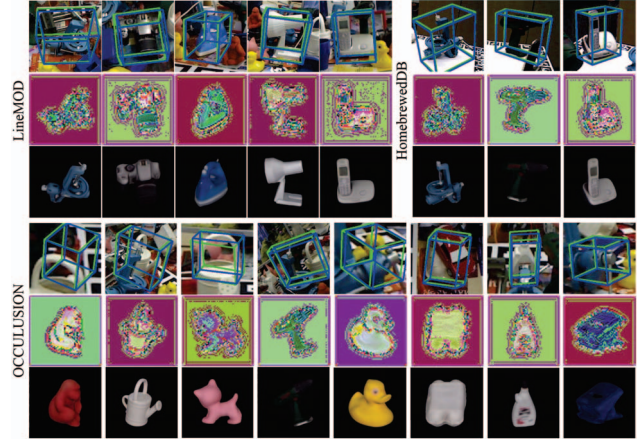


Figure 4. Qualitative results on Linemod (top-left), Linemod-Occluded (bottom) and HomebrewedDB (top-right). For each dataset, Top: input images and the visualizations of pose, green and blue bounding boxes represent GT poses and estimated poses, respectively. Middle: Visualization of generated environment-independent object representations. Bottom: The image after decoding the environment-independent object representations.

Table 2. Evaluation results on HomebrewedDB dataset. All results except ours are copied from TexPose [5]. The best pose method using the same kind of training data is underlined, and the overall best method is marked in bold.

Training data	$R^- + S$			$S$	
Method	Sock[32]	Self6D++[40]	Texpose[5]	GDR[41]	EA6D
Benchvise	57.3	75.7	<u>92.9</u>	88.8	<b>97.6</b>
Driller	46.6	<u>89.4</u>	<b>94.2</b>	92.8	<u>93.8</u>
Phone	41.5	76.8	<u>81.2</u>	78.7	<b>92.6</b>
Mean	52.0	80.6	<u>89.4</u>	86.8	<b>94.7</b>

Table 3. Evaluation results on DiverseScenes dataset. All cited methods are re-implemented using their open source codes.

Methods	Self6D++ [40]	GDR [41]	Texpose [5]	EA6D
Scene 1	56.5	64.1	82.0	<b>84.4</b>
Scene 2	62.2	70.7	76.4	<b>83.9</b>
Scene 3	48.6	68.9	66.5	<b>83.0</b>
Scene 4	69.5	54.3	64.3	<b>84.0</b>
Mean	59.2	64.5	72.3	<b>83.8</b>

ment, we want to study the generalizability of our method towards new scenes. First, we evaluate our method on HomebrewedDB. From Table 2, we observe that EA6D outperforms all other methods and achieves the same results as testing in Linemod (95.7% vs.94.7%). Self6D++ and TexPose undergo a clear performance drop when testing on Linemod. Self6D++ drops from 94.1% to 80.6% and TexPose drops from 91.8% to 89.4%. It indicates that the generalization performance of existing self-supervised methods is limited by the unannotated real images used for training. Further, Table 3 shows the results of the test on the DiverseScenes dataset. In environments with significant changes

in background and lighting, there is a certain decline in performance for all four methods, but our method is the least affected and outperforms the others by a huge leap.

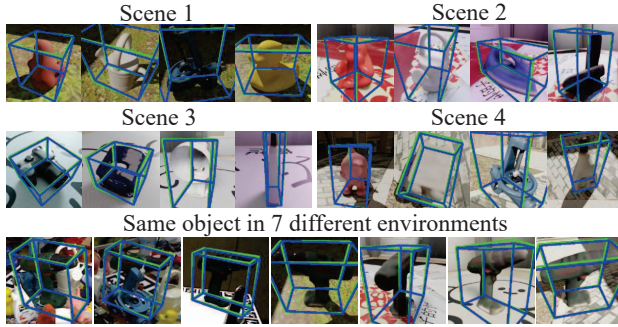


Figure 5. The top two rows are qualitative results on DiversifyScenes Dataset. Each scene has different backgrounds and lighting conditions. The bottom row is qualitative results of a same object in 7 different environments.

We would like to emphasise that since our method is only trained on Synthetic PBR Dataset [13], the aforementioned datasets are all unseen environments for our method. Our method maintains accurate predictions (average 90.6%) under different backgrounds, lighting conditions, and occlusions. It shows that the choice of adaptation domain (LM, HB, or other environments) has a marginal effect on our final performance. We attribute this to the generation of pure object features, eliminating the domain gap problem caused by environmental factors.

#### 4.5. Visualizations

In this section, we will analyze our work by visualization of the test results. Figure 4 illustrates the visualizations of environment-independent object representations and pose estimation. It clearly shows that the images generated by the diffusion model filter out all environmental factors while accurately preserving the object information. The visualizations of results on LM-O explain why EA6D can effectively handle occlusions. With the strong generative capabilities of diffusion models, even when facing severe occlusion, the model is still able to accurately generate occluded parts. Figure 5 shows the visualizations of the results on the DiverseScenes Dataset and the visualizations of the same object in 7 different environments. These provide a clear demonstration of its capabilities in handling various environments. We give more visualizations results in supplementary material.

#### 4.6. Ablation Studies

**The effect of diffusion models and diffusion steps.** We study the impact of different diffusion models on EA6D performance. We replace EDDM with traditional Denoising Diffusion Probabilistic Model (DDPM) [11] and Decoupled Diffusion Model (DDM) [15] and compare the results on Linemod. Also, the number of iterating steps is

another key parameter in diffusion models that determines the inference speed, which is essential for pose estimation. All the results are reported in Table 4. We can observe that the EA6D with EDDM achieves more accurate results with fewer iterating steps. This result verifies the effectiveness of mixing of input image latent representations and noise instead of pure noise in generation process, which provides prior information, thus speeding up the inference procedure and improving the generation quality.

Table 4. Mean results with ablation EA6D and the number of iterating steps on Linemod.

Method \ steps	1	5	10	20	100	1000
EA6D-DDPM [11]	0	0	3.6	14.3	52.7	78.4
EA6D-DDM [15]	93.1	96.1	95.6	96.3	96.2	96.3
EA6D w/o VAE	68.4	71.3	73.1	75.1	76.2	77.4
EA6D*	76.7	79.2	79.2	79.4	79.3	79.5
EA6D	96.3	96.3	96.4	96.5	96.5	96.5
Time(s) EA6D	0.23	0.51	0.87	1.60	6.98	92.1

**The effect of object representations.** In our framework, we generate the object representations via performing the denoising process to estimate object pose. The object representations are the latent code of pure object images encoded by VAE. To evaluate the efficacy of such representations, we first remove the VAE and directly generate the pure object images as object representations through EDDM. Then we use the framework of EA6D, but decode the generated object representations to pure object images using the decoder of VAE and input pose decoder to estimate pose. We denote this setting as EA6D\*. As shown in Table 4, the performance of these two types of object representations significantly drops, showing that the latent code of pure object images captures the rich semantic and structural information in the image and is more suitable for pose estimation than the generated images.

## 5. Conclusion

In this paper, we propose a novel generative paradigm for 6D pose estimation. Different from typical RGB-based pose estimation frameworks which directly extract features from input images for pose estimation and are therefore easily influenced by environmental factors, we employ an environment decoupling model that separates the object from environmental factors during the denoising process, generating environment-independent object representations which are robust to environmental changes. A pose decoder is designed to estimate poses using these representations. The experimental results on five datasets demonstrate that EA6D significantly outperforms current state-of-the-art self-supervised methods, particularly in its ability to generalize to previously unseen scenarios.

## 6. Acknowledgments

This research was supported by the National Key Research and Development Program of China (No.2024YFF0907600), the National Natural Science Foundation of China (Grant No.62273275), the China Postdoctoral Science Foundation (No.2022M722573), Shaanxi Postdoctoral Research Project (No.2023BSHEDZZ244), Shaanxi Natural Science Basic Research Program (No.2025JC-YBQN-928), Key Research and Development Plan of Shaanxi Province (2025GH-YBXM-018) and Xi'an Science and Technology Innovation and Qinchuangyuan Innovation Major Program (23ZDCYJSGG0009-2023).

## References

- [1] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *ECCV*, 2006. 2
- [2] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020. 5, 6
- [3] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother. Learning 6d object pose estimation using 3d object coordinates. In *ECCV*, 2014. 1, 2, 6
- [4] E. Brachmann, F. Michel, A. Krull, M.Y. Yang, S. Gumhold, and C. Rother. Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image. In *CVPR*, 2016. 2
- [5] Hanzhi Chen, Fabian Manhardt, Nassir Navab, and Benjamin Busam. Texpose: Neural texture learning for self-supervised 6d object pose estimation. In *CVPR*, 2023. 2, 5, 6, 7
- [6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 6
- [7] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 2, 3
- [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 *ICCV*, 2021. 6, 7
- [9] Jia Gong, Lin Geng Foo, Zhipeng Fan, QiuHong Ke, Hossein Rahmani, and Jun Liu. Diffpose: Toward more reliable 3d pose estimation. In *CVPR*, 2023. 3
- [10] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *ACCV*, 2012. 1, 6
- [11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2, 3, 8
- [12] Tomáš Hodan, P. Haluza, Štěpán Obdržálek, J. Matas, M. Lourakis, and X. Zabulis. T-less: An rgb-d dataset for 6d pose estimation of texture-less objects. In *WACV*, 2017. 1
- [13] Tomáš Hodaň, Martin Sundermeyer, Bertram Drost, Yann Labbé, Eric Brachmann, Frank Michel, Carsten Rother, and Jiří Matas. Bop challenge 2020 on 6d object localization. In *ECCV Workshops*, 2020. 6, 8
- [14] Yinlin Hu, Pascal Fua, Wei Wang, and Mathieu Salzmann. Single-stage 6d object pose estimation. In *CVPR*, 2020. 2
- [15] Yuhang Huang, Zheng Qin, Xinwang Liu, and Kai Xu. Decoupled diffusion models with explicit transition probability. *arXiv preprint arXiv:2306.13720*, 2023. 3, 4, 5, 8
- [16] Roman Kaskman, Sergey Zakharov, Ivan Shugurov, and Slobodan Ilic. Homebreweddb: Rgb-d dataset for 6d pose estimation of 3d objects. In *ICCV Workshops*, 2019. 1, 6
- [17] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab. Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In *ICCV*, 2017. 1, 2
- [18] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014. 3
- [19] V. Lepetit, F. Moreno-Noguer, and P. Fua. Epnnp: An accurate o(n) solution to the pnp problem. *IJCV*, 81(2):155–166, 2009. 2
- [20] Fu Li, Shishir Reddy Vutukur, Hao Yu, Ivan Shugurov, Benjamin Busam, Shaowu Yang, and Slobodan Ilic. Nerf-pose: A first-reconstruct-then-regress approach for weakly-supervised 6d object pose estimation. In *ICCV*, 2023. 1, 2
- [21] Hongyang Li, JieHong Lin, and Kui Jia. Dcl-net: Deep correspondence learning network for 6d pose estimation. In *ECCV*, 2022. 2
- [22] Zhigang Li, Gu Wang, and Xiangyang Ji. Cdpn: Coordinates-based disentangled pose network for real-time rgb-based 6-dof object pose estimation. In *ICCV*, 2019. 1, 2, 5
- [23] Ruyi Lian and Haibin Ling. Checkerpose: Progressive dense keypoint localization for object pose estimation with graph neural network. In *ICCV*, 2023. 1, 2, 6, 7
- [24] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 5
- [25] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *CVPR*, 2022. 3
- [26] Fabian Manhardt, Diego Martin Arroyo, Christian Rupprecht, Benjamin Busam, Tolga Birdal, Nassir Navab, and Federico Tombari. Explaining the ambiguity of object detection and 6d pose from visual data. In *ICCV*, 2019. 2, 6, 7
- [27] Kiru Park, Timothy Patten, and Markus Vincze. Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation. In *ICCV*, 2019. 2
- [28] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao. Pvnnet: Pixel-wise voting network for 6dof pose estimation. In *CVPR*, 2019. 1, 2, 6, 7
- [29] M. Rad and V. Lepetit. Bb8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth. In *ICCV*, 2017. 2

- [30] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bjorn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 4
- [31] Ivan Shugurov, Sergey Zakharov, and Slobodan Ilic. Dpodv2: Dense correspondence-based 6 dof pose estimation. *TPAMI*, 2021. 1
- [32] Juil Sock, Guillermo Garcia-Hernando, Anil Armagan, and Tae-Kyun Kim. Introducing pose consistency and warp-alignment for self-supervised 6d object pose estimation in color images. In *3DV*, 2020. 7
- [33] Chen Song, Jiaru Song, and Qixing Huang. Hybridpose: 6d object pose estimation under hybrid representations. In *CVPR*, 2020. 2
- [34] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 2, 3
- [35] 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 *CVPR*, 2022. 1, 2, 6, 7
- [36] M. Sundermeyer, Z. Marton, M. Durner, M. Brucker, and R. Triebel. Implicit 3d orientation learning for 6d object detection from rgb images. In *ECCV*, 2018. 2, 6, 7
- [37] Tao Tan and Qiulei Dong. Smoc-net: Leveraging camera pose for self-supervised monocular object pose estimation. In *CVPR*, 2023. 1, 2, 5, 6, 7
- [38] B. Tekin, S.N. Sinha, and P. Fua. Real-time seamless single shot 6d object pose prediction. In *CVPR*, 2018. 2
- [39] Gu Wang, Fabian Manhardt, Jianzhun Shao, Xiangyang Ji, Nassir Navab, and Federico Tombari. Self6d: Self-supervised monocular 6d object pose estimation. In *ECCV*, 2020. 1, 2, 6, 7
- [40] Gu Wang, Fabian Manhardt, Xingyu Liu, Xiangyang Ji, and Federico Tombari. Occlusion-aware self-supervised monocular 6d object pose estimation. *TPAMI*, 2021. 1, 2, 6, 7
- [41] Gu Wang, Fabian Manhardt, Federico Tombari, and Xiangyang Ji. Gdr-net: Geometry-guided direct regression network for monocular 6d object pose estimation. In *CVPR*, pages 16611–16621, 2021. 1, 2, 7
- [42] Tianfu Wang, Guosheng Hu, and Hongguang Wang. Object pose estimation via the aggregation of diffusion features. In *CVPR*, 2024. 3
- [43] Yangzheng Wu and Michael Greenspan. Pseudo-keypoint rkhs learning for self-supervised 6dof pose estimation. In *ECCV*, 2024. 2, 6, 7
- [44] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017. 1, 2, 6
- [45] Li Xu, Haoxuan Qu, Yujun Cai, and Jun Liu. 6d-diff: A keypoint diffusion framework for 6d object pose estimation. In *CVPR*, 2024. 2, 3, 6, 7
- [46] Zongxin Yang, Xin Yu, and Yi Yang. Dsc-posenet: Learning 6dof object pose estimation via dual-scale consistency. In *CVPR*, 2021. 1, 2, 6, 7
- [47] Yunfan Ye, Kai Xu, Yuhang Huang, Renjiao Yi, and Zhiping Cai. Diffusionedge: Diffusion probabilistic model for crisp edge detection. In *AAAI*, 2024. 3
- [48] S. Zakharov, I. Shugurov, and S. Ilic. Dpod: 6d pose object detector and refiner. In *ICCV*, 2019. 2, 6, 7
- [49] Shaobo Zhang, Wanqing Zhao, Ziyu Guan, Xianlin Peng, and Jinye Peng. Keypoint-graph-driven learning framework for object pose estimation. In *CVPR*, 2021. 2