

# Object Pose Estimation via the Aggregation of Diffusion Features

Tianfu Wang<sup>1,2,3</sup>, Guosheng Hu<sup>4</sup>, Hongguang Wang<sup>1,2</sup>

<sup>1</sup>State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences

<sup>2</sup>Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences

<sup>3</sup>University of Chinese Academy of Sciences, Beijing, 100049, China

<sup>4</sup>Oosto, Belfast, U.K.

wangtianfu100@gmail.com, huguosheng100@gmail.com, hgwang@sia.cn

## Abstract

*Estimating the pose of objects from images is a crucial task of 3D scene understanding, and recent approaches have shown promising results on very large benchmarks. However, these methods experience a significant performance drop when dealing with unseen objects. We believe that it results from the limited generalizability of image features. To address this problem, we have an in-depth analysis on the features of diffusion models, e.g. Stable Diffusion, which hold substantial potential for modeling unseen objects. Based on this analysis, we then innovatively introduce these diffusion features for object pose estimation. To achieve this, we propose three distinct architectures that can effectively capture and aggregate diffusion features of different granularity, greatly improving the generalizability of object pose estimation. Our approach outperforms the state-of-the-art methods by a considerable margin on three popular benchmark datasets, LM, O-LM, and T-LESS. In particular, our method achieves higher accuracy than the previous best arts on unseen objects: 98.2% vs. 93.5% on Unseen LM, 85.9% vs. 76.3% on Unseen O-LM, showing the strong generalizability of our method. Our code is released at <https://github.com/Tianfu18/diff-feats-pose>.*

## 1. Introduction

Estimating the pose of objects from images is crucial for various real-world applications, including robot manipulation [4, 38, 47], augmented reality [23], and many others. Thanks to the advancements in deep learning, the performance of object pose estimation has significantly improved.

The most popular methods for object pose estimation can be categorized into three approaches: indirect methods [7, 13, 19, 27, 29, 43], direct methods [3, 5, 15, 22, 39, 41], and template-based methods [1, 16, 18, 35, 40]. Template-based methods involve matching the input image of an ob-

ject to a set of template images with various poses rendered by 3D models to achieve the object’s pose. In this work, we focus on template-based methods due to their simplicity, increasing popularity, and promising accuracy.

Despite the success of the aforementioned methods, image-based object pose estimation methods have not yet become widely adopted in real-world applications. In particular, the ability to handle unseen objects without the need for retraining is crucial for quick and convenient adaptation to new scenarios. Recent works [24, 36, 46] have begun to tackle this challenge. The method [36] introduces a novel architecture with multiple decoders to adapt to different objects. The results indeed demonstrate the interesting generalization to unseen objects, however, these objects must share great similarity with the training objects. Methods [24, 46] propose the use of local object representations for template matching. Their methods have achieved interesting results, but a noticeable performance gap remains between seen and unseen objects. For example, the accuracy of one state-of-the-art method [24] is 99.1% on Seen LM dataset and 94.4% on Unseen LM dataset, resulting in a performance gap of approximately 4.7%.

The performance gap in the existing methods [24, 36, 46] is empirically found to mainly result from the inadequacy of their discriminative features. This finding, which will be detailed in Sec. 3.2, inspires us to learn image representations which can generalize well to unseen objects. On the other hand, we realize that text-to-image diffusion models have already proven their ability to generate high-quality images with robust features which can generalize well to different scenarios [20, 21, 37, 42, 44]. This promising generalizability can be attributed to four main factors: 1) Text supervision with rich semantic content can lead to highly discriminative features. 2) Diffusion models encode the information at various timesteps, capturing a spectrum of granularity and diverse attributes. 3) Diffusion models [44] are found capable of encoding 3D characteristics, including scene geometry, depth, etc. 4) Diffusion models, like

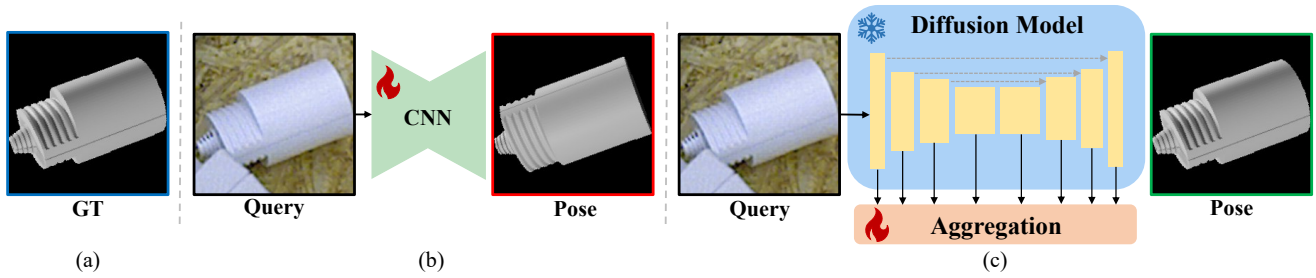


Figure 1. Unseen object pose estimation of one state-of-the-art method [24] and our method. (a) We render an image using the ground truth pose of an unseen object. (b) Template-pose [24] learns image features by fine-tuning a self-supervised learning [9] pre-trained model. (c) Our method aggregates features of different granularity from a diffusion model to achieve better pose estimation than template-pose [24] on the unseen object. The snowflake and flame symbols represent ‘parameters frozen’ and ‘fine-tune’, respectively.

Stable Diffusion [32], benefit from extensive training data, which enables them to learn discriminative features across a wide range of scenarios. Based on these advantages, we are inspired to explore the aggregation of diffusion features to generate discriminative features for object pose estimation as illustrated in Fig. 1.

To achieve this, we propose three architectures, namely, Arch. (a), Arch. (b) and Arch. (c), to aggregate diffusion features into an optimal feature for object pose estimation. First, we design a vanilla aggregation network, Arch. (a), which aligns the features to the same dimension through linear mapping and then aggregates them via element-wise addition. However, the limited capability of this linear network constrains its performance. To enhance the network’s fitting capability, we introduce Arch. (b), which replaces the linear mapping with a bottleneck module consisting of three convolution layers and ReLU functions. Finally, to obtain the optimal weights for different features, we design a context-aware weight aggregation network, Arch. (c), which learns the weights based on the context. To verify the efficacy of our aggregation methods, we conducted experiments on three popular benchmarks: LINEMOD (LM) [10], Occlusion-LINEMOD (O-LM) [2], and T-LESS [12]. On these benchmarks, our network achieves superior results over state-of-the-art methods and significantly reduces the performance gap between seen and *unseen* objects.

In summary, this work has the following contributions:

- We have an in-depth analysis on the diffusion features, which exhibit great potential for modeling *unseen* objects, and creatively incorporate diffusion features into object pose estimation. To our knowledge, we are the first to innovatively and systematically investigate the aggregation of diffusion features for object pose estimation.
- We propose three aggregation networks which can effectively capture different dynamics of the diffusion features, leading to the promising generalizability of object pose estimation.

- Our approach greatly outperforms the state-of-the-art methods on three benchmark datasets, LINEMOD (LM) [10], Occlusion-LINEMOD (O-LM) [2], and T-LESS [12]. In particular, our method performs significantly better than other methods on *unseen* datasets, 98.2% vs. 93.5% [24] on Unseen LM, 85.9% vs. 76.3% [24] on Unseen O-LM, showing the strong generalizability of our methods.

## 2. Related work

Estimating the pose of objects is a fundamental task in computer vision. In recent years, object pose estimation has been dominated by learning-based approaches which achieve great progress. Thus, in this section, we review those commonly employed learning-based approaches.

**Indirect methods** Many learning-based approaches are based on establishing 2D-3D correspondences [7, 13, 19, 27, 29, 43], followed by PnP and RANSAC [6, 17] to estimate the pose. These methods primarily focus on the ways of obtaining accurate 2D-3D correspondences. BB8 [29] regresses the 2D coordinates of projected 3D bounding box corners. PVNet [28] regresses pixel-wise unit vectors pointing to 2D projections of a set of 3D keypoints. Methods [19, 27, 43] employ an encoder-decoder network to regress pixel-level dense correspondences, which are the 2D coordinates of the object surface.

**Direct methods** Direct methods [5, 22, 41] treat pose estimation as a regression task, directly outputting the object’s pose. In addition, SSD-6D [15] divides the pose space into categories, transforming it into a classification problem. Some recent methods [3, 39] make the indirect method’s PnP process differentiable and use the 2D-3D correspondence from the indirect approach as a surrogate task.

**Template-Based Methods** Template-based methods involve determining the object’s pose by matching a query

image of the object with one of a set of templates, which are rendered images of the 3D model in various poses. One line of methods focuses on obtaining discriminative representations. Method [40] trains a network to acquire discriminative features of objects and use them for matching during testing. Building on this, method [1] considers the exact pose differences between training samples to obtain more discriminative representations. AAE [35] proposes learning representations by using a denoising autoencoder to recover the original image from an augmented image. Another line of methods includes DeepIM [18], which proposes iterative refinement by regressing a pose difference between a render of the pose hypothesis and the input image. CosyPose [16] is built on DeepIM, introducing improvements like a better architecture, a continuous rotation parametrization, etc.

**Unseen Object Pose Estimation** Recently, unseen object pose estimation attracted a growing interest. Some prior works [1, 18, 40] demonstrated that template-based methods could exhibit a certain degree of generalization to unseen objects. Therefore, recent studies [24, 36, 46] have explored template-based methods specifically tailored for unseen objects. Due to the inherent challenges in estimating the pose of unseen objects, these studies simplify the problem by assuming that the object is already localized in 2D and only focus on estimating the 3D pose (3D orientation). Despite the success achieved by these methods, they still exhibit noticeable performance gap between seen and unseen objects. In our research, we also concentrate on 3D pose estimation of objects, aiming at minimizing the performance gap between seen and unseen objects.

### 3. Methodology

In this section, we first formulate the task of object pose estimation (Sec. 3.1); We then explore the significance of features and identify the suitability of the intermediate features generated by the text-to-image diffusion model for object pose estimation (Sec. 3.2). Finally, we propose feature aggregation methods that aggregate intermediate features from the diffusion model to achieve the optimal feature for object pose estimation (Sec. 3.3).

#### 3.1. Object pose estimation

Given an input object image, the task of object pose estimation is to predict the class of the object and estimate the rigid transformation from the object coordinate system to the camera coordinate system. Currently, the most popular methods for object pose estimation are mainly categorized into three approaches: indirect methods [7, 13, 19, 27, 29, 43], direct methods [3, 5, 15, 22, 39, 41], and template-based methods [1, 16, 18, 24, 35, 36, 40, 46]. In this work, we focus on template-based methods due to their simplicity and promising generalizability.

Template-based methods usually use 3D models to render a set of template images with the annotations of object classes and poses, which can be naturally achieved during the rendering process. The input image  $I$  and these template images are mapped to the feature space via  $\Phi_{encoder}$  in order to compute their similarity:

$$\mathcal{F} = \Phi_{encoder}(I) \quad (1)$$

The class  $\hat{c}$  and the pose  $\hat{P}$  of the most similar template image is then assigned to the given real image to achieve the object pose estimation.

Clearly, we can observe that acquiring an effective feature  $\mathcal{F}$  through Eq. (1) is pivotal for computing the similarity between one real and template image, therefore, is important for the successful estimation of object pose.

#### 3.2. Motivation: feature matters

In recent years, object pose estimation has achieved significant success, e.g. template-based methods [1, 36, 40], on seen objects. However, these methods tend to struggle when dealing with *unseen* objects. To address this issue, recent methods [24, 46] propose to encode images into local features, resulting in interesting performance for unseen objects. In spite of the considerable progress, their methods still exhibit a noticeable performance gap between seen and unseen objects.

Despite the advancements in feature engineering, generalizing these improvements to *unseen* objects remains a significant challenge. Figure 2 illustrates this challenge by selecting two images (query) from the LINEMOD dataset. One image contains a *seen* object, the lamp, while the other showcases an *unseen* cat. Subsequently, we render the corresponding template images (e.g. the leftmost one in the 2nd row) using the same pose as the query. We then visualize the features of query and template in the feature space of the state-of-the-art method [24]. We project these features to a PCA space and assign RGB channels for visualization. The more similar color channels indicate more similar features. Remarkably, for the *seen* object lamp, the features of the query and template exhibit significant similarity. However, for the *unseen* object cat, notable differences emerge. This discovery underscores the primary focus of this paper: the pursuit of a generalized object features to accurately estimate the pose of unseen objects.

To obtain the discriminative features with strong generalization capability, we realize the recent text-to-image diffusion models have demonstrated strong features to generate high-quality images. As discussed in Sec. 1, a number of recent studies [20, 21, 37, 42, 44] have demonstrated that diffusion features generalize well to different scenarios.

Since image features are important for object pose estimation and the diffusion features can potentially offer discriminative features, we are inspired to investigate the fea-

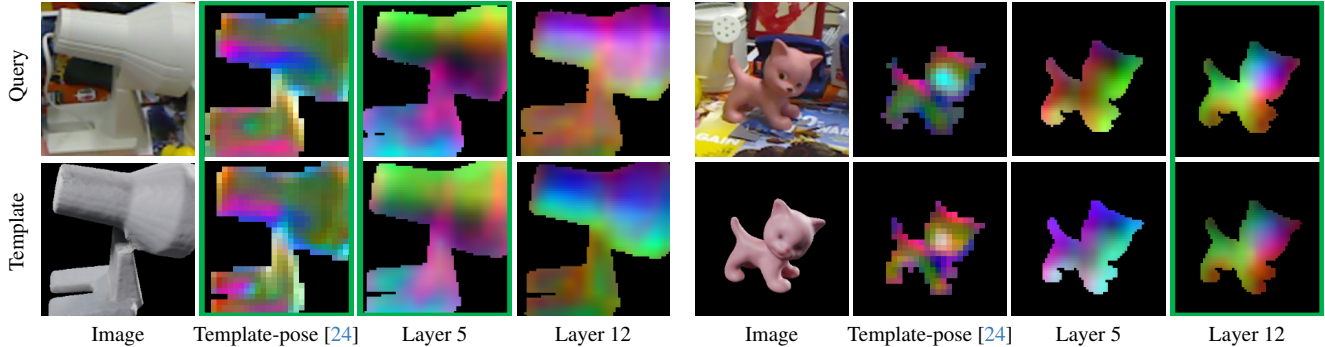


Figure 2. Feature visualization of LINEMOD. For query and template images, we visualize their 3 features from template-pose [24], Layer 5 and Layer 12 of a diffusion model. These features are projected to a PCA space, and the values of top 3 principal components are assigned to RGB values respectively for visualization. The more similar the colors of two feature images in one column, the more similar in feature space. The two features in one green box are very similar.

sibility of using diffusion features for object pose estimation. To verify this assumption, we employ a well-known diffusion model, Stable Diffusion [32], for this investigation. When an image is fed into Stable Diffusion, it generates a set of intermediate features from a UNet. As illustrated in Fig. 2, we visualize the intermediate features from both Layer 5 and Layer 12 of the UNet at timestep  $t = 0$ . Interestingly, the features from Layer 5 can well measure the similarity of query and template for the object lamp; in contrast, Layer 12 is the best for the object cat.

**Summarization** (1) The features of the state-of-the-art method [24] cannot well measure the similarity of unseen objects; (2) The features of Stable Diffusion show great potential to model *unseen* objects. For example, we do not use the lamp and cat images from LINEMOD dataset to finetune the model, however, Stable Diffusion performs quite well; (3) The features of Stable Diffusion from different layers can capture different dynamics, e.g. Layer 5 and 12 are the best for the lamp and cat respectively. It inspires us to investigate the aggregation of features from different layers to achieve strong generalization capability on unseen objects.

### 3.3. Diffusion features

In this section, we first introduce the diffusion process in Sec. 3.3.1, based on which we propose the solutions of diffusion feature aggregations in Sec. 3.3.2.

#### 3.3.1 Diffusion process

Text-to-image diffusion models involve both forward and reverse processes. One widely used sampling procedure for diffusion models is DDIM [34], as defined:

$$x_t = \sqrt{\alpha_t}x_0 + \sqrt{1 - \alpha_t}\epsilon_t, \text{ where } \epsilon_t \sim \mathcal{N}(0, 1) \quad (2)$$

where  $x_0$  denotes the initial clean sample,  $\alpha_t$  denotes the pre-defined noise schedule, and  $\epsilon_t$  denotes the randomly generated noise. During the forward process, Gaussian noise is gradually added to the clean sample over  $T$  steps, resulting in a sequence of noisy samples:  $x_1, x_2, \dots, x_T$ . As  $T$  approaches infinity,  $x_T$  converges to an isotropic Gaussian distribution. In the reverse process, the diffusion model is sampled by iteratively denoising  $x_T \sim \mathcal{N}(0, 1)$  conditioned on a text prompt  $y$ . Specifically, at each denoising step for  $t = 1, \dots, T$ ,  $x_{t-1}$  is determined using both  $x_t$  and the text prompt  $y$ . After the final denoising step,  $x_0$  is transformed back to the clean sample.

In practice, our objective is to obtain features directly from clean images without relying on conditional prompts. Specifically, we achieve this by employing an unconditioned text embedding and running Stable Diffusion only once with a very small timestep, such as  $t = 0$ . With such a small timestep, from the perspective of the diffusion model, clean images are predominantly treated as denoised images.

#### 3.3.2 Diffusion features aggregation

In Sec. 3.3.1, we introduce the feature extraction from a diffusion model. Motivated by the findings in Fig. 2, the aggregation of features extracted from different layers can potentially generalize well to different objects and scenarios. Thus, in this section, we investigate the ways of aggregation.

Given a set of raw diffusion features  $\{f_1, f_2, \dots, f_n\}$  from  $n$  layers in a diffusion model, we need to employ trainable architectures to aggregate these features into a fused one  $\hat{\mathcal{F}}$ . The aggregation involves two essential components, an extractor  $\Phi_{\text{ext}}$  and an aggregator  $\Phi_{\text{aggr}}$ . Specifically,  $\Phi_{\text{ext}}$  is employed to align the raw diffusion features to the same dimension and to learn task-specific features adapting to object pose estimation; while  $\Phi_{\text{aggr}}$  is employed to aggregate

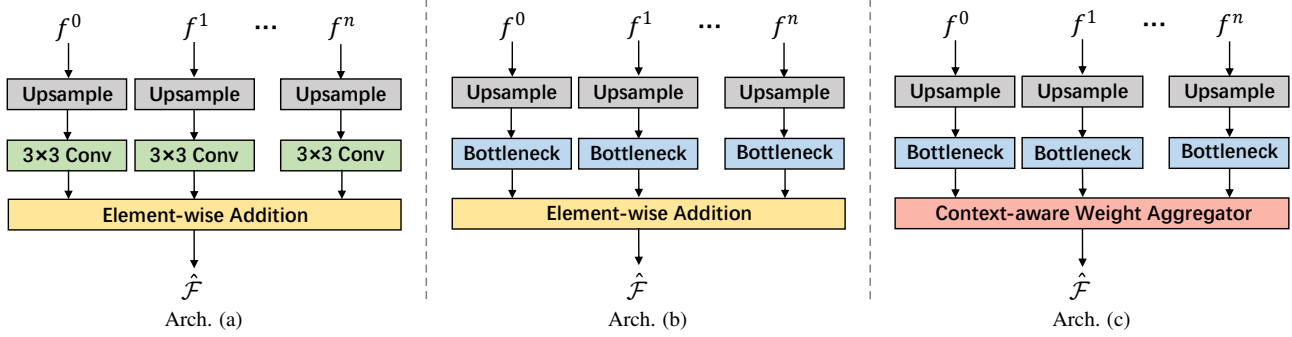


Figure 3. Diffusion aggregation methods. Arch. (a) is a vanilla aggregation, Arch. (b) is nonlinear aggregation, and Arch. (c) is context-aware weight aggregation.

the extracted features. This process can be standardized as follows:

$$\hat{\mathcal{F}} = \Phi_{\text{aggr}}(\{\Phi_{\text{ext}}^1(f_1), \Phi_{\text{ext}}^2(f_2), \dots, \Phi_{\text{ext}}^n(f_n)\}) \quad (3)$$

In this section we discuss several aggregation architectures for diffusion features.

**Arch. (a)** First, we design a vanilla aggregation network, illustrated in Fig. 3a. Given a set of diffusion features, we first upsample them to a standard resolution. Next, we utilize  $3 \times 3$  convolution layers to learn features specialized for object pose estimation and project them to the same channel count. Finally, we aggregate them directly through element-wise addition as:

$$\hat{\mathcal{F}} = \sum_{i=1}^n \Phi_{\text{ext}}^i(f_i) \quad (4)$$

**Arch. (b)** Since the vanilla aggregation network only performs linear mapping on the original features and lacks any nonlinearity, it falls short in capturing complex data patterns and nonlinearities. Hence, we design a nonlinear aggregation network, illustrated in Fig. 3b. In this architecture, we substituted  $\Phi_{\text{ext}}$  from the vanilla aggregation network with a bottleneck layer [8] to introduce nonlinearity. Specifically, the bottleneck layer consists of three convolutions and ReLU functions with a skip connection. However, as the number of parameters increases, the limited training data adversely affects the generalization of the aggregated features. In response to this challenge, we take inspiration from ControlNet [45] and initialize the last convolution in the bottleneck with zero values.

**Arch. (c)** During aggregation, Arch. (a) and (b) implicitly set the weights of each feature as 1 for element-wise addition. However, the optimal weights can be learned to lead to better aggregation. Thus, we design a so-called context-aware weight aggregation network that learns the weights

based on the context, as illustrated in Fig. 3c. Specifically, we start by upsampling all features and then passing them through an extractor composed of bottleneck layers same as Arch. (b):

$$h_i = \Phi_{\text{ext}}^i(f_i) \quad (5)$$

Subsequently, we apply an average pooling layer to transform each  $h_i$  into one-dimensional features  $l_i$ . Following this, we concatenate all these one-dimensional features to represent the entire context. Finally, we employ a multilayer perceptron (MLP) and a softmax to determine the weights associated with each feature:

$$\{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_n\} = \text{softmax}(\text{MLP}([l_1, l_2, \dots, l_n])) \quad (6)$$

Finally, these weights are used to perform a weighted sum of all the features to achieve the aggregation:

$$\hat{\mathcal{F}} = \sum_{i=1}^n \mathbf{w}_i \cdot h_i \quad (7)$$

## 4. Experiment

In this section, we first introduce the implementation details, training and test, the datasets, and metrics used for the evaluation (Sec. 4.1). Subsequently, we present an ablation study for timestep, aggregation methods and other pre-trained models (Sec. 4.2). Finally, we compare our method with the state-of-the-art methods on LM, O-LM and T-LESS datasets (Sec. 4.3).

### 4.1. Experimental settings

#### 4.1.1 Implementation details

In our experiments, we extract features from Stable Diffusion (SD) v1-5 [32], a generative latent diffusion model trained on LAION-5B [33]. We feed images with a resolution of 512 into this SD model and aggregate features from all layers of its UNet into output features with a dimensionality of  $64 \times 64$ . We train our network for 20 epochs with a

learning rate of 1e-3 for LM dataset and 1e-4 for T-LESS dataset. Throughout all experiments, we directly feed images into the SD model without adding any noise.

### 4.1.2 Training and test

To adapt our method to object pose estimation, we freeze the weights of SD model and only train our aggregation networks with pose estimation supervision. During training, following template-pose [24], we create 1 positive pair template and  $M - 1$  negative pair templates for each real image. We train our model to maximize the agreement between the representations of samples in positive pairs while minimizing that of negative pairs using the InfoNCE loss [25]. In testing, we find the template that is most similar to the input image and assign the identity and pose information annotated in the template to the input image to complete pose estimation. We use the same similarity measure as template-pose [24]. We first calculate the cosine similarity between features, then apply a threshold and template mask for filtering, and finally, we take the average of the remaining values as the final similarity measurement.

### 4.1.3 Dataset

Following existing works in object pose estimation [24], we use three popular object pose estimation benchmarks: LINEMOD (LM) [10], Occlusion-LINEMOD (O-LM) [2], and T-LESS [12]. LM and O-LM are standard benchmarks for evaluating object pose estimation methods.

The LM dataset consists of 13 sequences, each containing ground truth poses for a single object. CAD models for all the objects are provided. The O-LM dataset is an extension of the LM dataset, with objects in the dataset being heavily occluded, making pose estimation highly challenging. We follow the approach outlined in template-pose [24] to split the cropped data into three non-overlapping groups based on the depicted objects, reserving 10% of the data for testing on seen objects. Our model is trained on LM and tested on both LM and O-LM. Following the protocol of Wohlhart *et al.* [40], we render 301 templates per object. The input image is cropped at the center of objects with the ground-truth pose, following the approach used in previous works [1, 24, 40].

T-LESS [12] comprises 30 objects characterized by a lack of distinct textures or distinguishable colors. These objects are symmetric and similar in shape and size, presenting challenging scenarios with substantial clutter and occlusion. For T-LESS, we adopt the testing methodology established in previous works [24, 36]. We split the 30 objects into two groups: objects 1-18 are considered as seen objects, and objects 19-30 as unseen objects. Our model is trained on the dataset that includes only the seen objects, and testing is

performed on the complete test dataset. Following the protocol of template-pose [24], we render 21,672 templates per object. Similar to previous works [24, 35, 36], we use the ground-truth bounding box to crop the input image.

### 4.1.4 Evaluation metrics

For the LM and O-LM datasets, we calculate pose error by computing the geodesic distance [40] between two rotations:

$$d(\hat{\mathbf{R}}, \mathbf{R}) = \arccos\left(\frac{\text{tr}(\mathbf{R}^T \hat{\mathbf{R}}) - 1}{2}\right) / \pi \quad (8)$$

Here,  $\mathbf{R}$  denotes the ground truth 3D orientation,  $\hat{\mathbf{R}}$  denotes the predicted 3D orientation, and  $\text{tr}$  denotes the trace. In the case of unseen object, the object class is unknown during testing. The accuracy is defined as the percentage of test images for which the best angle error is below a specific threshold and the predicted object class is correct. It can be computed as follows:

$$\text{Acc.} = \begin{cases} 1 & \text{if } d(\hat{\mathbf{R}}, \mathbf{R}) < \lambda \text{ and } \hat{c} = c \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

Here,  $c$  denotes the ground truth class. Following template-pose [24], we use the Acc15 metric and the  $\lambda$  is set to 15.

Because most of the objects in T-LESS are symmetric, we follow the protocol of previous works [11, 24, 35, 36] and use the recall under the Visible Surface Discrepancy (VSD) metric with a threshold of 0.3. Additionally, we predict the translation using the projective distance estimation of SSD-6D [15], following the same approach as previous works [24, 35, 36].

## 4.2. Ablation study

### 4.2.1 Timestep

As mentioned in Sec. 3.2, the diffusion model encodes more specific information at various timesteps. To understand the impact of timestep, we conduct a performance comparison by extracting features from the SD model at different timesteps. We employ our vanilla aggregation network, Arch. (a), to conduct experiments on objects included in split #1 of both Seen LM and Seen O-LM to quickly identify the best timestep. We sampled eleven timesteps from 0 to 1000, evenly spaced. The results on seen objects of split 1 of the LM and O-LM datasets are presented in Fig. 4. The results show that the accuracies on LM and O-LM decrease as the timestep increases. This is because the diffusion model assumes less noise in the input when the timestep is small, which matches the scenarios in our images. Therefore, we set the timestep to 0 for the remaining experiments.

To evaluate the performance of the proposed aggregation networks, Tab. 1 and Tab. 2 compare these networks.

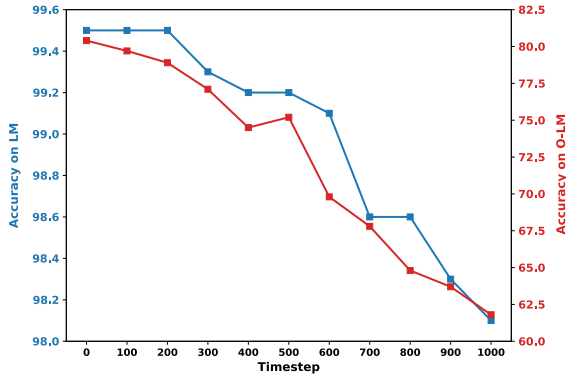


Figure 4. Ablation on timestep  $t$ . The accuracy is measured for the model with extracting features from Stable Diffusion at different timesteps on LM and O-LM datasets.

Method	Backbone	Split #1		Split #2		Split #3		Avg.	
		Seen	Unseen	Seen	Unseen	Seen	Unseen	Seen	Unseen
Balntas <i>et al.</i> [1]	Wohlhart <i>et al.</i> [40]	87.0	13.2	83.1	15.5	85.1	18.2	85.0	15.2
Wohlhart <i>et al.</i> [40]	Wohlhart <i>et al.</i> [40]	89.2	14.1	85.4	16.3	83.3	16.7	86.3	16.7
MPL [36]	MPL [36]	91.5	36.0	87.9	39.3	87.2	39.7	88.9	38.3
Zhao <i>et al.</i> [46]	ResNet [8]	96.9	87.5	94.4	76.2	93.1	73.7	93.8	79.2
Template-pose [24]	ResNet [8]	99.3	94.4	99.0	97.4	99.2	88.7	99.1	93.5
Ours (VA)	SD-V1-5 [32]	99.4	97.4	99.5	99.3	99.4	95.6	99.4	98.3
Ours (NA)	SD-V1-5 [32]	99.5	97.5	99.6	<b>99.6</b>	99.7	96.0	99.6	98.2
Ours (CWA)	SD-V1-5 [32]	99.5	<b>98.3</b>	99.6	98.9	<b>99.8</b>	<b>96.3</b>	<b>99.7</b>	98.2
Ours (CWA)	SD-V2-0 [32]	99.6	98.1	99.4	99.5	99.6	97.5	99.5	<b>98.4</b>
Ours (CWA)	OpenCLIP [31]	99.4	97.0	<b>99.7</b>	98.5	99.7	95.9	99.6	97.1
Ours (CWA)	DINOv2 [26]	<b>99.7</b>	96.6	99.6	99.1	99.5	<b>97.8</b>	99.6	97.8

Table 1. Results on LM [10]. VA represents vanilla aggregation, NA represents nonlinear aggregation, and CWA represents context-aware weight aggregation.

Method	Backbone	Split #1		Split #2		Split #3		Avg.	
		Seen	Unseen	Seen	Unseen	Seen	Unseen	Seen	Unseen
Balntas <i>et al.</i> [1]	Wohlhart <i>et al.</i> [40]	19.2	9.3	23.1	5.1	15.0	5.1	19.1	6.5
Wohlhart <i>et al.</i> [40]	Wohlhart <i>et al.</i> [40]	18.3	8.2	21.9	7.5	17.6	7.6	19.5	7.8
MPL [36]	MPL [36]	31.3	18.6	34.5	15.9	29.2	17.7	31.7	17.4
Zhao <i>et al.</i> [46]	ResNet [8]	54.9	40.1	63.4	32.7	49.9	37.5	56.1	36.8
Template-pose [24]	ResNet [8]	77.3	71.4	84.1	72.7	76.8	85.3	79.4	76.3
Ours (VA)	SD-V1-5 [32]	80.2	79.4	87.2	78.0	80.2	95.6	82.5	84.6
Ours (NA)	SD-V1-5 [32]	81.9	81.4	<b>88.5</b>	79.0	79.5	96.0	83.3	85.5
Ours (CWA)	SD-V1-5 [32]	81.5	<b>81.6</b>	86.0	<b>79.8</b>	<b>80.8</b>	<b>96.3</b>	<b>82.8</b>	<b>85.9</b>
Ours (CWA)	SD-V2-0 [32]	82.9	81.0	88.1	79.1	79.2	95.5	<b>83.4</b>	85.2
Ours (CWA)	OpenCLIP [31]	79.8	76.5	85.7	73.9	80.1	91.9	81.9	81.1
Ours (CWA)	DINOv2 [26]	<b>84.1</b>	73.7	96.1	74.3	79.0	86.3	83.1	78.1

Table 2. Results on O-LM [2]. VA represents vanilla aggregation, NA represents nonlinear aggregation, and CWA represents context-aware weight aggregation.

Notably, nearly all datasets achieve better results with the nonlinear aggregation network than with the vanilla aggregation network. This is because vanilla aggregation is not well-suited for adapting to downstream tasks. By introducing nonlinearity, the aggregation network can better fit the downstream task, resulting in improved performance. Furthermore, we observe significant performance improvements, especially on unseen objects, when using the context-aware weight aggregator. This is due to the context-aware weight aggregator’s ability to adapt its ag-

gregation weights to different inputs, improving the aggregation network’s ability to generalize to unseen data. The results confirm that the proposed extractors and aggregators can significantly improve the performance of the aggregation network. Based on these experimental results, we select the context-aware weight aggregation network for the remaining experiments.

#### 4.2.2 Other models pre-trained at large scale

In this section, we compare the features of other *big* models. We first use another SD model, SDv2-0, which differs from SDv1-5 in its text encoder, i.e. SDv1-5 with CLIP [30] and SDv2-0 with OpenCLIP [14]. Same as SDv1-5, we set the timestep to 0 and extract features from all layers in the UNet. Furthermore, our method can be easily incorporated into other pre-trained large visual models. As shown in Tab. 1 and Tab. 2, we have also applied our aggregation network to other models, including DINOv2 [26] pre-trained on LVD-142M dataset and OpenCLIP [30] pre-trained on LAION [33] dataset. We use the best checkpoints available on their GitHub repositories, ViT-G for DINOv2 and OpenCLIP. Similar to Stable Diffusion, we extract features from all transformer blocks of these models. We aggregate the intermediate features using the context-aware aggregation network, which is the best-performing network we obtained. The results are shown in Tab. 1 and Tab. 2. We can see that Stable Diffusion v1-5 achieves the best results in most of the benchmarks. This indicates that the intermediate features of the diffusion model have superior capabilities compared to the discriminative models for object pose estimation tasks.

### 4.3. Comparison with the state-of-the-art

#### 4.3.1 LINEMOD and Occluded-LINEMOD

We evaluate our method in Tab. 1 and Tab. 2, our method achieves significantly better results compared to previous state-of-the-art methods [1, 24, 36, 40, 46]. In particular, on the unseen O-LM test set, our model achieves an average accuracy 85.9%, outperforming template-pose [24] by 10.9%. This shows the superior generalization capability of our proposed solutions. Wohlhart *et al.* [40] and Balntas *et al.* [1] use a carefully designed network to learn feature embeddings with pose and class discriminative capabilities through the careful design of triplets and pairs loss functions. These methods use global representations and are more likely to fail when objects are occluded. Template-pose [24], which uses InfoNCE [25] loss and local per-patch representations masked by template masks, achieves better performance in the presence of occlusion, but there is a large performance gap between seen and unseen objects, 99.1% on Seen LM and 93.5% Unseen LM. In contrast, our method significantly reduces this gap, achieving 99.7%

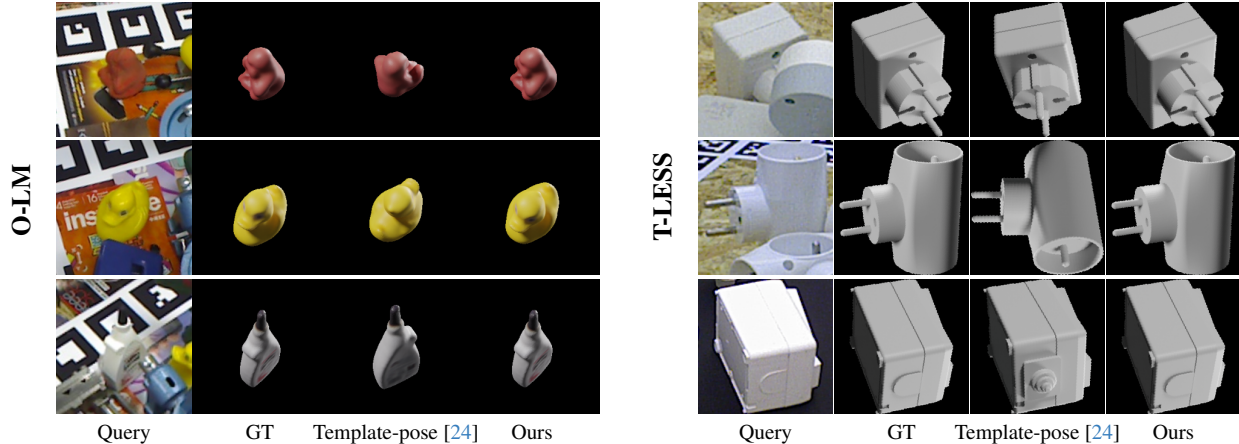


Figure 5. Qualitative results on unseen objects of O-LM (left) and T-LESS (right).

on Seen LM and 98.2% Unseen LM. This means that our method generalizes well to unseen objects. The strong generalizability shows the efficacy of our proposed solutions.

#### 4.3.2 T-LESS

The T-LESS dataset consists of particularly challenging texture-less rigid objects in highly cluttered scenes. In Tab. 3, we present the results of our method compared to state-of-the-art approaches [24, 35, 36]. Our method outperforms all the methods in this comparison. Notably, our method achieves higher accuracy with fewer templates compared to template-pose [24], 71.03% vs. 58.87%. When template-pose [24] uses the same number of templates as our method, we outperform it by 11.73% for seen objects and 14.35% for unseen objects. This demonstrates the enhanced discriminative power of the features obtained from the diffusion model. Notably, our method can even achieve superior results for unseen objects compared to seen objects, highlighting the strong generalizability of our method.

Method	Number templates	Recall VSD		
		Obj. 1-18	Obj. 19-30	Avg
Implicit [35]	92 K	35.60	42.45	38.34
MPL [36]	92 K	35.25	33.17	34.42
Template-pose [24]	92 K	59.62	57.75	58.87
Template-pose [24]	21 K	59.14	56.91	58.25
Ours (CWA)	21 K	<b>70.87</b>	<b>71.26</b>	<b>71.03</b>

Table 3. Results on T-LESS [12].

#### 4.4. Visualization

To better demonstrate the effectiveness of our aggregation method, we provide several qualitative results in Fig. 5. We

compare the estimated pose of our methods and the state-of-the-art method [24]. We find that our method performs better in challenging scenarios, e.g. objects with occlusions in T-LESS, showing the stronger generalizability and discriminative capability of our method.

#### 4.5. Efficiency

Table 4 displays the comparison of trainable parameter size with template-pose [24]. The proposed method outperforms template-pose with superior performance and approximately half the trainable parameters.

	Template-pose [24]	VA	NA	CWA
Params (M)	24.04	2.53	13.28	13.29

Table 4. Comparison of trainable parameter size.

### 5. Conclusion

In this study, we conduct an analysis of inaccurate object pose estimation, particularly for unseen objects. Our findings identify insufficient feature generalization as the primary culprit for these inaccuracies. To address this challenge, we propose three novel aggregation networks specifically designed to effectively aggregate diffusion features, exhibiting superior generalizability for object pose estimation. We evaluate our method on three standard benchmark datasets, demonstrating superior performance and improved generalization to unseen objects compared to existing methods. We hope that our findings and proposed method will serve as a catalyst for further advancements in this field.

**Acknowledgement** This work was funded in part by NSFC (No.52335003), and FRPSIA (Grant 2022JC3K06 and No.2023000479).



## References

- [1] Vassileios Balntas, Andreas Doumanoglou, Caner Sahin, Juil Sock, Rigas Kouskouridas, and Tae-Kyun Kim. Pose guided rgb-d feature learning for 3d object pose estimation. In *Proceedings of the IEEE international conference on computer vision*, pages 3856–3864, 2017. [1](#), [3](#), [6](#), [7](#)
- [2] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6d object pose estimation using 3d object coordinates. In *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part II 13*, pages 536–551. Springer, 2014. [2](#), [6](#), [7](#)
- [3] Hansheng Chen, Pichao Wang, Fan Wang, Wei Tian, Lu Xiong, and Hao Li. Epro-nnp: Generalized end-to-end probabilistic perspective-n-points for monocular object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2781–2790, 2022. [1](#), [2](#), [3](#)
- [4] Alvaro Collet, Manuel Martinez, and Siddhartha S Srinivasa. The moped framework: Object recognition and pose estimation for manipulation. *The international journal of robotics research*, 30(10):1284–1306, 2011. [1](#)
- [5] T Do, Trung Pham, Ming Cai, and Ian Reid. Real-time monocular object instance 6d pose estimation. 2019. [1](#), [2](#), [3](#)
- [6] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. [2](#)
- [7] 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. [1](#), [2](#), [3](#)
- [8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. [5](#), [7](#)
- [9] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. [2](#)
- [10] 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. [2](#), [6](#), [7](#)
- [11] Tomáš Hodaň, Jiří Matas, and Štěpán Obdržálek. On evaluation of 6d object pose estimation. In *Computer Vision—ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8–10 and 15–16, 2016, Proceedings, Part III 14*, pages 606–619. Springer, 2016. [6](#)
- [12] Tomáš Hodan, Pavel Haluza, Štěpán Obdržálek, Jiri Matas, Manolis Lourakis, and Xenophon Zabulis. T-less: An rgb-d dataset for 6d pose estimation of texture-less objects. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 880–888. IEEE, 2017. [2](#), [6](#), [8](#)
- [13] 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](#), [2](#), [3](#)
- [14] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, 2021. If you use this software, please cite it as below. [7](#)
- [15] 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](#), [2](#), [3](#), [6](#)
- [16] 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. [1](#), [3](#)
- [17] 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](#)
- [18] 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. [1](#), [3](#)
- [19] 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. [1](#), [2](#), [3](#)
- [20] Ziyi Li, Qinye Zhou, Xiaoyun Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie. Guiding text-to-image diffusion model towards grounded generation. *arXiv preprint arXiv:2301.05221*, 2023. [1](#), [3](#)
- [21] Grace Luo, Lisa Dunlap, Dong Huk Park, Aleksander Holynski, and Trevor Darrell. Diffusion hyperfeatures: Searching through time and space for semantic correspondence. *arXiv preprint arXiv:2305.14334*, 2023. [1](#), [3](#)
- [22] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6841–6850, 2019. [1](#), [2](#), [3](#)
- [23] 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](#)
- [24] Van Nguyen Nguyen, Yinlin Hu, Yang Xiao, Mathieu Salzmann, and Vincent Lepetit. Templates for 3d object pose es-

- timization revisited: Generalization to new objects and robustness to occlusions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6771–6780, 2022. [1](#), [2](#), [3](#), [4](#), [6](#), [7](#), [8](#)
- [25] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. [6](#), [7](#)
- [26] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. [7](#)
- [27] 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. [1](#), [2](#), [3](#)
- [28] 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. [2](#)
- [29] 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. [1](#), [2](#), [3](#)
- [30] Alec Radford, Jong Wook Kim, Chris Hallacy, A. Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. [7](#)
- [31] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. [7](#)
- [32] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. [2](#), [4](#), [5](#), [7](#)
- [33] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022. [5](#), [7](#)
- [34] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. [4](#)
- [35] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3d orientation learning for 6d object detection from rgb images. In *Proceedings of the european conference on computer vision (ECCV)*, pages 699–715, 2018. [1](#), [3](#), [6](#), [8](#)
- [36] Martin Sundermeyer, Maximilian Durner, En Yen Puang, Zoltan-Csaba Marton, Narunas Vaskevicius, Kai O Arras, and Rudolph Triebel. Multi-path learning for object pose estimation across domains. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13916–13925, 2020. [1](#), [3](#), [6](#), [7](#), [8](#)
- [37] Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent correspondence from image diffusion. *arXiv preprint arXiv:2306.03881*, 2023. [1](#), [3](#)
- [38] Jonathan Tremblay, Thang To, Balakumar Sundaralingam, Yu Xiang, Dieter Fox, and Stan Birchfield. Deep object pose estimation for semantic robotic grasping of household objects. *arXiv preprint arXiv:1809.10790*, 2018. [1](#)
- [39] 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](#), [2](#), [3](#)
- [40] Paul Wohlhart and Vincent Lepetit. Learning descriptors for object recognition and 3d pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3109–3118, 2015. [1](#), [3](#), [6](#), [7](#)
- [41] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017. [1](#), [2](#), [3](#)
- [42] Jiarui Xu, Sifei Liu, Arash Vahdat, Wonmin Byeon, Xiaolong Wang, and Shalini De Mello. Open-vocabulary panoptic segmentation with text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2955–2966, 2023. [1](#), [3](#)
- [43] Sergey Zakharov, Ivan Shugurov, and Slobodan Ilic. Dpod: 6d pose object detector and refiner. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1941–1950, 2019. [1](#), [2](#), [3](#)
- [44] Guanqi Zhan, Chuanxia Zheng, Weidi Xie, and Andrew Zisserman. What does stable diffusion know about the 3d scene? *arXiv preprint arXiv:2310.06836*, 2023. [1](#), [3](#)
- [45] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. [5](#)
- [46] Chen Zhao, Yinlin Hu, and Mathieu Salzmann. Fusing local similarities for retrieval-based 3d orientation estimation of unseen objects. In *European Conference on Computer Vision*, pages 106–122. Springer, 2022. [1](#), [3](#), [7](#)
- [47] Menglong Zhu, Konstantinos G Derpanis, Yinfei Yang, Samarth Brahmbhatt, Mabel Zhang, Cody Phillips, Matthieu Lecce, and Kostas Daniilidis. Single image 3d object detection and pose estimation for grasping. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3936–3943. IEEE, 2014. [1](#)