MatchU: Matching Unseen Objects for 6D Pose Estimation from RGB-D Images

Supplementary Material

In this Appendix, we first introduce the detailed network architecture of each module for our method as well as training and inference parameters in Section 1. We then provide the formal definition of the coarse-level loss function in Section 2. More visualization results are shown in Section 4. We further conduct an ablation study on the verification and refinement method in Section 3. Finally, more discussions about the failure cases and limitations are provided in Section 5.

1. Implementation Details

1.1. Network Architecture

As shown in Figure 1, our network consists of two 3D backbones that perform encoding and decoding for the CAD points and depth points respectively, one 2D backbone that embeds textural information from RGB images, and one Latent Fusion Attention Module proposed to fuse the 2D and 3D cues in the coarse-level latent space.

3D Backbone. We adjust RoITr [12] network as our 3D backbone to extract rotation-invariant point descriptors for both the CAD and depth point clouds. For each observed depth image, we sample at most 2048 points as the target point cloud and the same number is applied for sampling points from the CAD model surface as the source point cloud. The 3D encoder-decoder architecture is based on the Point Pair Feature Transformer (PPFTrans). Our point cloud encoder consists of 4 blocks, as shown in Table 1, each block consists of an Attentional Abstraction Layer (AAL) for downsampling and abstraction and a PPF Attention Layers (PALs) for local geometry encoding and context aggregation. Both layers are in the same design with RoITr. We use the stride of 1,4, 2, and 2 for the encoder blocks respectively. The encoder blocks finally down-sample the point clouds into 128 superpoints with 256 feature dimensions. The decoder is also built by 4 blocks, as shown in Table 2, each block is made of a Transition Up Layer (TUL) that performs upsampling and context aggregation, as well as a PAL which enhances the highly-representative learned context.

2D Backbone. For learning the textural information from the RGB images, we follow the implementation of LoFTR [8] and use a modified ResNet18 [2] as the 2D backbone network. It consists of 3 convolutional blocks which down-sample the input image into $\frac{W}{2} \times \frac{H}{2}$, $\frac{W}{4} \times \frac{H}{4}$, $\frac{W}{8} \times \frac{H}{8}$ feature maps respectively. The feature map with resolution of $\frac{W}{8} \times \frac{H}{8} \times 256$ is then flattened and encoded with the positional embeddings. Same as LoFTR, the 2D extension of the absolute sinusoidal positional encoding is adopted. Finally, the superpixels from 2D branch are further fused with the obtained 3D superpoint features by the proposed Latent Fusion Attention Module.

Stage	Block	Operation
Input		$P \in \mathbb{R}^{n\times 1}$
Encoder	$\text{Block}_1^e(P) \rightarrow P_1$	$\text{AAL}(n \times 1) \rightarrow n \times 64$ $\text{PAL}(n \times 64) \rightarrow n \times 64$
	$\text{Block}_2^e(P_1) \rightarrow P_2$	$\text{AAL}(n \times 64) \rightarrow n/4 \times 128$ $\text{PAL}(n/4 \times 128) \rightarrow n/4 \times 128$
	$\text{Block}_3^e(P_2) \rightarrow P_3$	$\text{AAL}(n/4 \times 128) \rightarrow n/8 \times 256$ $\text{PAL}(n/8 \times 256) \rightarrow n/8 \times 256$
	$\text{Block}_4^e(P_3) \rightarrow P'$	$\text{AAL}(n/8 \times 256) \rightarrow n/16 \times 256$ $\text{PAL}(n/16 \times 256) \rightarrow n/16 \times 256$
Output		P' , $\phi^{p'}$

Table 1. Detailed architecture of our 3D encoder.

Stage	Block	Operation
Input		$P' \in \mathbb{R}^{n/16\times 256}$
Decoder	$\text{Block}_4^d(P') \rightarrow \hat{P}_4$	TUL $(n/16 \times 256) \rightarrow n/16 \times 256$ PAL: $n/16 \times 256 \rightarrow n/16 \times 256$
	$\text{Block}_3^d(\hat{P}_4, P_3) \rightarrow \hat{\mathcal{P}}_3$	$\text{TUL}(n/16 \times 256,\ n/8 \times 256) \rightarrow n/8 \times 256$ $\text{PAL}(n/8 \times 256) \rightarrow n/8 \times 256$
	$\text{Block}_2^d(\hat{P}_3, P_2) \rightarrow \hat{\mathcal{P}}_2$	$\text{TUL}(n/8 \times 256,\ n/4 \times 128) \rightarrow n/4 \times 128$ $\text{PAL}(n/4 \times 128) \rightarrow n/4 \times 128$
	$\text{Block}_1^d(\hat{\mathcal{P}}_2, P_1) \rightarrow \hat{P}$	$\text{TUL}(n/4 \times 128,\ n \times 64) \rightarrow n \times 64$ $\text{PAL}(n \times 64) \rightarrow n \times 64$
Output		$\hat{P}$, ϕ^p

Table 2. Detailed architecture of our 3D decoder.

Global Transformer. We aggregate the global context with a Global Transformer proposed in [**?**]. It consists of 3 blocks, each block is a sequence of a geometry-aware self-attention module (GSM) and a position-aware cross-attention module (PCM). The global transformer encodes the spatial cues from both CAD and depth point clouds and enhances the geometric features for both sides.

Latent Fusion Attention Module. As introduced in the main text, our Latent Fusion Attention Module consists of two types of transformers, *i.e.* Fusion Transformer and Global Transformer. The detailed network structure of this module is shown in the right part of Figure 1. For the global transformers, we follow the design of [**?**]. It consists of 3 blocks, each block is a sequence of a geometry-aware self-attention module (GSM) and a position-aware cross-

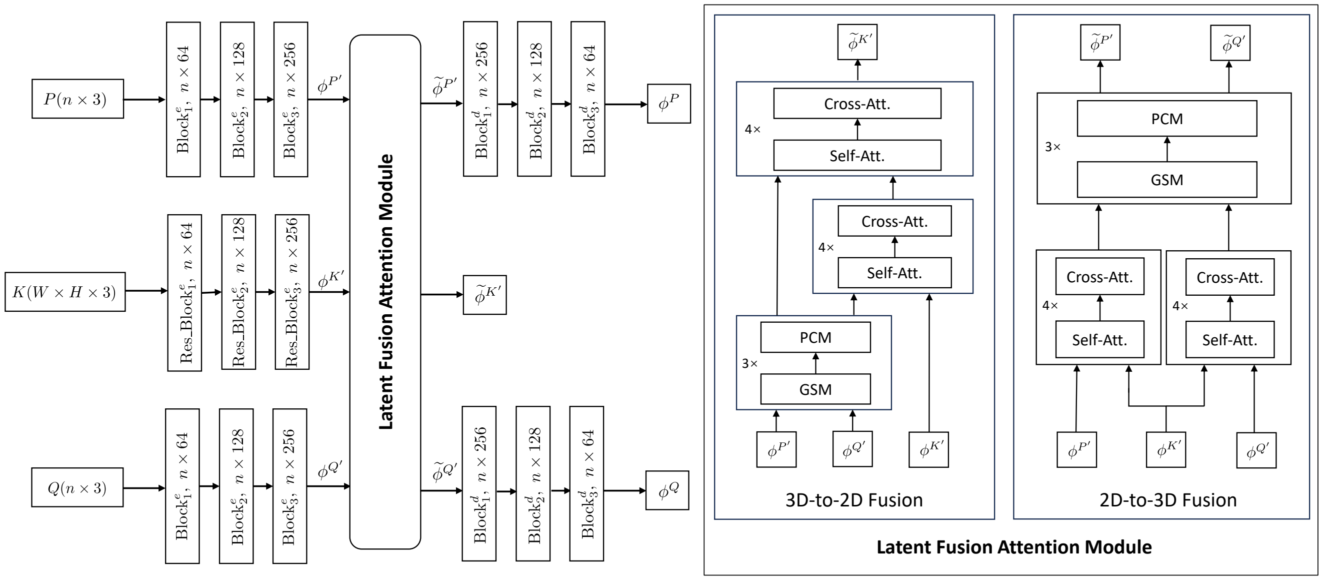
$P(n \times 3)$
$\mathrm{Block}^e_1,\ n \times 64$
$\mathrm{Block}^e_2,\ n \times 128$
$\mathrm{Block}^e_3,\ n \times 256$
$\phi^{P'}$
$\widetilde{\phi}^{P'}$
$\mathrm{Block}^d_1,\ n \times 256$
$\mathrm{Block}^d_2,\ n \times 128$
$\mathrm{Block}^d_3,\ n \times 64$
ϕ^P
$K(W \times H \times 3)$
$\mathrm{Res\_Block}^e_1,\ n \times 64$
$\mathrm{Res\_Block}^e_2,\ n \times 128$
$\mathrm{Res\_Block}^e_3,\ n \times 256$
$\phi^{K'}$
$\widetilde{\phi}^{K'}$
$Q(n \times 3)$
$\mathrm{Block}^e_1,\ n \times 64$
$\mathrm{Block}^e_2,\ n \times 128$
$\mathrm{Block}^e_3,\ n \times 256$
$\phi^{Q'}$
$\widetilde{\phi}^{Q'}$
$\mathrm{Block}^d_1,\ n \times 256$
$\mathrm{Block}^d_2,\ n \times 128$
$\mathrm{Block}^d_3,\ n \times 64$
ϕ^Q
Latent Fusion Attention Module
$\widetilde{\phi}^{K'}$
Cross-Att.
4×
Self-Att.
Cross-Att.
4×
Self-Att.
PCM
3×
GSM
$\phi^{P'}$
$\phi^{Q'}$
$\phi^{K'}$
3D-to-2D Fusion
$\widetilde{\phi}^{P'}$
$\widetilde{\phi}^{Q'}$
PCM
3×
GSM
Cross-Att.
4×
Self-Att.
Cross-Att.
4×
Self-Att.
$\phi^{P'}$
$\phi^{K'}$
$\phi^{Q'}$
2D-to-3D Fusion
Latent Fusion Attention Module

Figure 1. The detailed network architecture of our model.

attention module (PCM). The global transformer encodes the spatial cues from both CAD and depth point clouds and enhances the geometric features for both sides. For the fusion transformers, We follow [8] to design the attention layers and use 4 self- and cross-attention layers in each transformer block. In the Latent Fusion Attention Module, we obtain the features of 2D and 3D in different fusion branches. For 3D-to-2D fusion branch, we first use a global transformer to aggregate the cross-frame context between CAD and depth superpoints, then we fuse the obtained global-aware depth and CAD superpoint features with the 2D local features respectively, from which we obtain the fused 2D superpixel features $\widetilde{\phi}^{K'}$. For 2D-to-3D fusion branch, we first encode 2D features into CAD and depth superpoint features respectively, then we aggregate the fused CAD and depth features with a global transformer and obtain the global-aware 3D superpoint features $\widetilde{\phi}^{P'}$ and $\widetilde{\phi}^{Q'}$.

1.2. Training and Inference Parameters

During training, all the objects are normalized into a sphere with a radius of 0.1m, and we let λ_r to be 0.01 and 0.005 for coarse and fine matching respectively. We set λ_b as 0.3 and γ_c as 0.5 by default, and adopt an Adam [3] optimizer with an initial learning rate of 1e-4, which is exponentially decayed by 0.05 after each epoch. We train our model around 1M iterations with a batch size of 2 on two RTX4090 GPUs, while the model is tested without CPU parallel and with a batch size of 1. During inference, we set $\kappa = 128$, $s = 64$ and $\eta = 20$ by default. For the testing under the condition of unseen object localization, we adopt a CAD-based novel object segmentation method [6] to localize the objects in our input RGB-D images. For the evaluation of 6D pose with instance-level localization, we take the existing detection results from the methods [1, 5, 10]. To boost the performance, we scale up $\eta = 64$ to achieve more accurate results in comparison to some baselines[4, 13]. In all the experiments, we select 128 superpoint correspondences with the highest confidence scores as the coarse-level superpoint matching.

2. Loss Function

We explain the detailed Circle Loss [9] for the coarse-level matching between the superpoints and superpixels. The Circle Loss re-weight each similarity score under supervision, and is compatible with both class-level labels and pair-wise labels. Following [7], we use the overlaps between the corresponding superpoints as the similarity re-weighting score. For each superpoint $p'_i \in P'$, and $q'_j \in Q'$, we can calculate the overlap $\mathcal{V}$ between p'_i and q'_j as:

$$\mathcal{V}(p'_i, q'_j) = \frac{|\{\hat{p'}_u \in \hat{P'}_i \mid \exists\, \hat{q'}_v \in \hat{Q'}_j : \hat{p'}_u \leftrightarrow \hat{q'}_v\}|}{|\{\hat{p'}_u \in \hat{P'}_i\}|}, \quad (1)$$

where $\leftrightarrow$ denotes the correspondence relationship. $\hat{P'}_i$ is the group of points from P' assigned to p'_i by Point-to-Node grouping strategy [11], and $\hat{Q'}_j$ means the same for Q'.

A pair of superpoints p'_i and q'_j are considered as a positive pair if and only if $\mathcal{V}(p'_i, q'_j) > \tau_r$, where τ_r is the threshold of the overlap. We sample a positive set of superpoints from Q', denoted as $\mathcal{E}^P_i = \{q'_j \in Q' \mid \mathcal{V}(p'_i, q'_j) > \tau_r\}$, and a negative set of superpoints $\mathcal{F}^P_i = \{q'_j \in Q' \mid \mathcal{V}(p'_i, q'_j) =$

$0\}$. Then for P', the coarse-level superpoint matching loss is computed as:

$$\mathcal{L}_c^{P'} = \frac{1}{n'} \sum_{i=1}^{n'} \log[1+ \\ \sum_{q'_j \in \mathcal{E}_i^P} \exp(v_i^j \beta_e^{i,j} (d_i^j - \Delta_e)) \sum_{q'_k \in \mathcal{F}_i^P} \exp(\beta_f^{i,k} (\Delta_f - d_i^k))], \tag{2}$$

with $v_i^j = \mathcal{V}(p'_i, q'_j)$ and $d_i^j = \|\widetilde{\phi_{p'_i}} - \widetilde{\phi_{q'_j}}\|_2$. Δ_e and Δ_f are the positive and negative margins. $\beta_e^{i,j} = \gamma(d_i^j - \Delta_e)$ and $\beta_f^{i,k} = \gamma(\Delta_f - d_i^k)$ are the weights individually determined for different samples, with γ as a hyper-parameter. This Circle Loss minimizes the corresponding latent features and maximizes the incorresponding features among the superpoints and superpixels, establishing the cross-modality matches on coarse-level latent space.

3. Verification and Refinement

As introduced in the main text, our model is able to produce multiple pose hypotheses from the correspondences. We consider the availability of 2D RGB images and 3D point clouds information and employ both of them to perform verification for our pose hypotheses. As for 3D, we calculate the one-directional chamfer distance between the transformed CAD model and the lifted depth map as the score of 6D pose estimation. Specifically, we first transform the CAD point cloud with our predicted pose hypotheses respectively. For each point in the CAD points, we find the nearest point in the depth points and then measure the Euclidean distance between the CAD point and its nearest reference point. Finally, the pose hypothesis with the smallest mean distance over the CAD points is considered to be the final prediction. However, this measurement is not accurate because a low 3D point-distance-based score does not guarantee an accurate pose. As a complementary option, we also consider using 2D information as verification. We adopt a pre-trained scoring network the same as [4]. It takes a pair of RGB image crop and its coarse 6D pose prediction as input and estimates a confidence score for each input pair. Training with a large amount of image-pose pairs, this simple deep network is able to reject the outlier pose predictions. In addition to the 2D and 3D verification, we further boost the performance with a standard ICP procedure. In Table 3, we evaluate the effect of RGB verification and ICP refinement for our method on LM-O dataset. We observe that the RGB verification can help reject the outlier hypotheses, and ICP can refine our estimated poses even further. By adopting both the RGB verification and ICP refinement, our method achieves the best performance.

RGB Verification	ICP	AR
		56.2
✓		61.1
	✓	63.5
✓	✓	64.4

Table 3. **Ablation study on verification and refinement of our method on LM-O dataset.**

4. More Qualitative Results

We show more qualitative results on TUD-L, IC-BIN and YCB-V datasets in Figure 3, 4, and 5 respectively.

5. Limitations

Failure Cases. We show some of our failure cases in Figure 2. Our method fails to predict the objects in the first two columns because of different levels of external occlusion, which are reasonable since the observed part is limited by the viewpoint in both texture and depth. In the third column, the orientation of the banana is wrong although the predicted partial pose is well-aligned. This is because the distinctive part is unseen and causes ambiguity when matching the partial observation. We expect this ambiguity can be solved by increasing the resolution of our descriptors. Another failure case is shown in the fourth column, our method predicts the pose of the milk box upside down without external occlusion, which happens because of the highly symmetric texture and geometry presented in the observed part, introducing ambiguity when matching the descriptors with high similarity.

Further Discussions. Our method produces generic descriptors by fusing the 2D textural and 3D geometric information in the latent space. It captures the symmetry distribution automatically without external annotation. However, our method still needs further improvement in the verification of the correspondences and poses, especially when matching those highly symmetric objects that present similar features. Feasible solutions could be introducing a spatial consistency mechanism to filter out ambiguous correspondences or some learning-based and graph search methods. By puring the correspondences, higher quality of the pose hypotheses can be produced. On the other hand, instance mask prediction can also be involved in our pipeline by adding an additional output channel in our 2D branch, while multi-task training may also cause potential risks for representation learning. Finally, directly matching the objects to a raw scene without detection or segmentation is also a challenging and interesting direction for future research.

Object 6 DoF pose estimation is a crucial aspect in the

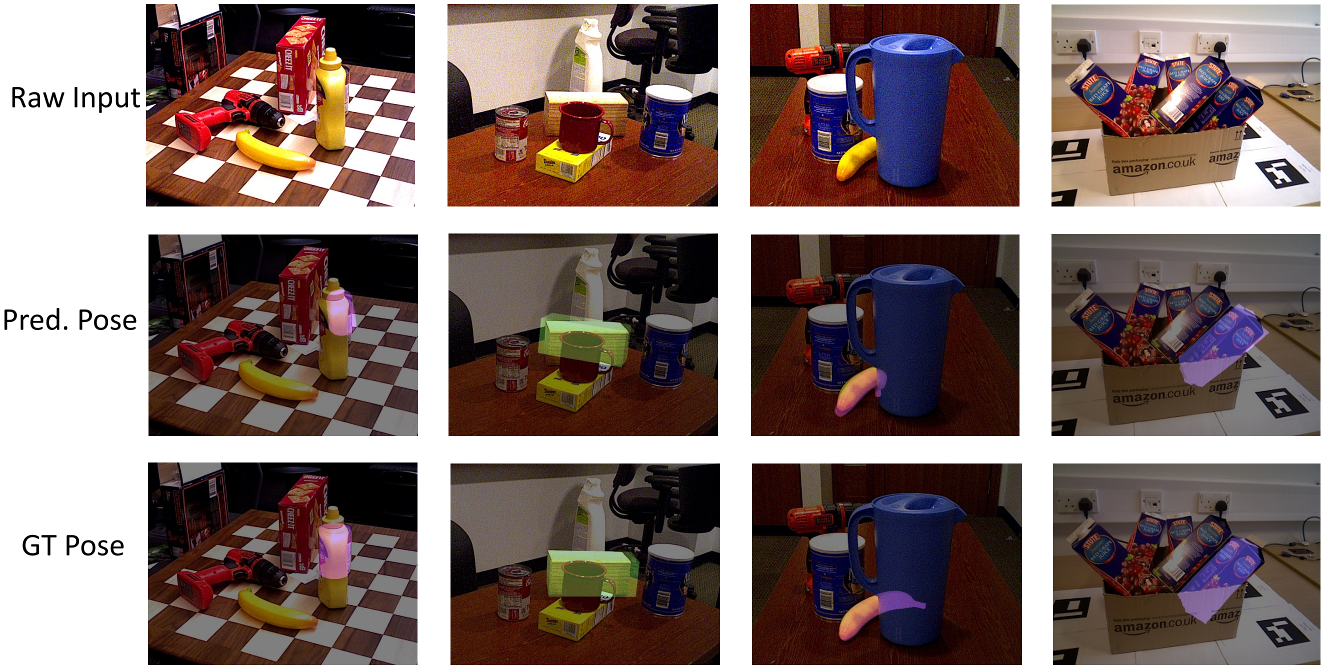

Figure 2. Failure case visualization of our method.

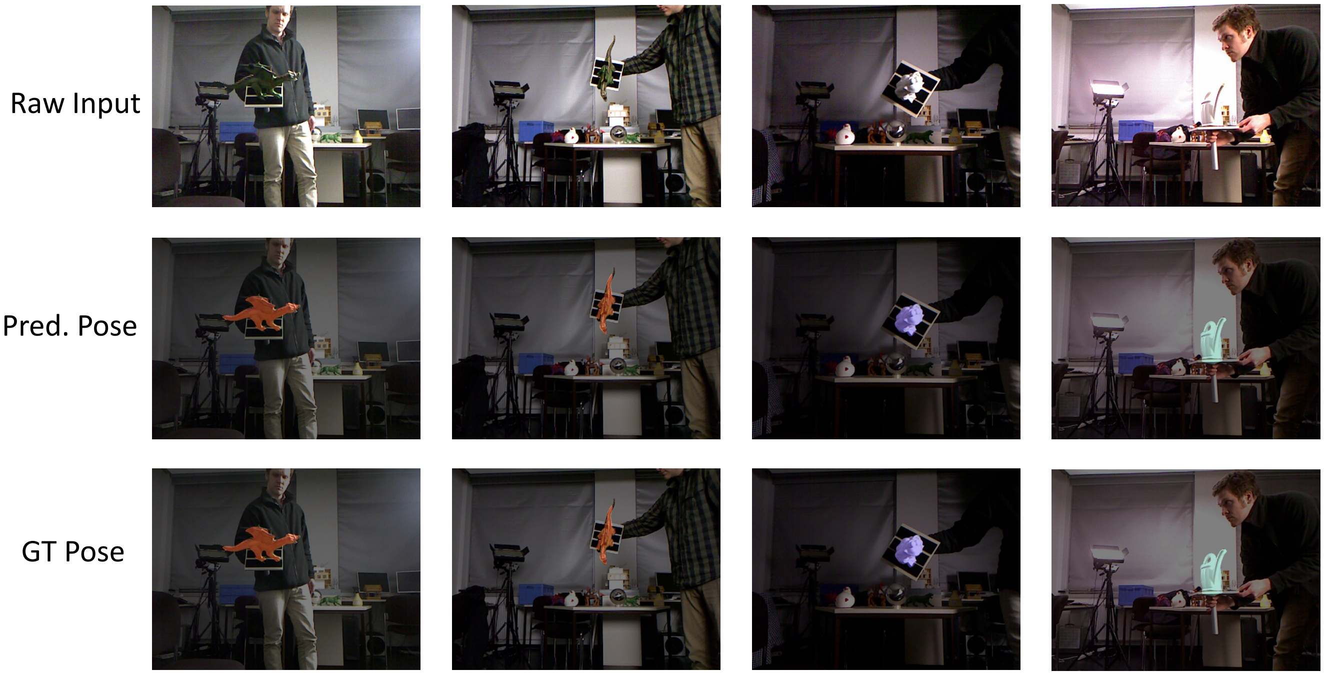

Figure 3. Visualization results of our method on TUD-L dataset.

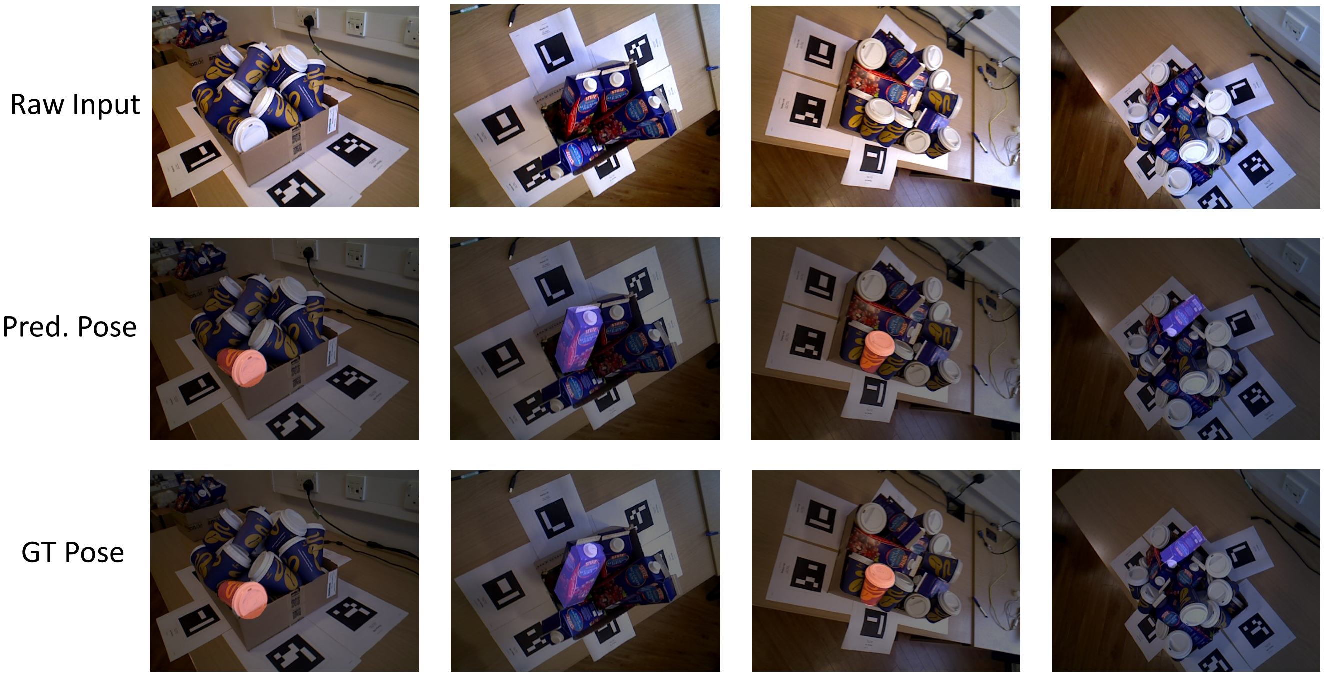

Figure 4. Visualization results of our method on IC-BIN dataset.

realms of computer vision and robotics, which has witnessed considerable attention, particularly with the advent of deep learning methodologies. Recent years have seen the emergence of diverse techniques, encompassing 2D-3D or 3D-3D descriptor-matching approaches, template-based methods, direct-regression methods, etc. However, challenges persist in scenarios involving textureless objects, occlusion, and cluttered backgrounds. This thesis introduces a novel pose estimation method based on diffusion models, which have demonstrated notable success in image synthesis tasks. The objective is to leverage the strengths of diffusion models, such as their scalability to large-scale data and the diversity of generation, to address the complexities inherent in 6 DoF pose estimation. A comprehensive diffusion framework is proposed, utilizing a single RGB-D image to generate multiple hypotheses for the 6 DoF pose of an object. The proposed method is rigorously evaluated on real-world datasets featuring noise and occlusion. Results indicate that our approach achieves comparable performance with state-of-the-art methods, highlighting its effectiveness in challenging scenarios. This study not only contributes a novel perspective to pose estimation but also underscores the potential of diffusion models in advancing the state of the art in this critical domain.

References

[1] Jianqiu Chen, Mingshan Sun, Tianpeng Bao, Rui Zhao, Liwei Wu, and Zhenyu He. Zeropose: Cad-model-based zero-shot pose estimation, 2023. 2

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1

[3] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2

[4] Yann Labbé, Lucas Manuelli, Arsalan Mousavian, Stephen Tyree, Stan Birchfield, Jonathan Tremblay, Justin Carpentier, Mathieu Aubry, Dieter Fox, and Josef Sivic. Megapose: 6d pose estimation of novel objects via render & compare. In *CoRL*, 2022. 2, 3

[5] 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. 2

[6] Van Nguyen Nguyen, Thibault Groueix, Georgy Ponimatkin, Vincent Lepetit, and Tomas Hodan. Cnos: A strong baseline for cad-based novel object segmentation. In *ICCV*, 2023. 2

[7] Zheng Qin, Hao Yu, Changjian Wang, Yulan Guo, Yuxing Peng, and Kai Xu. Geometric transformer for fast and robust point cloud registration. In *CVPR*. 2

[8] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *CVPR*, 2021. 1, 2

[9] Yifan Sun, Changmao Cheng, Yuhan Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. Circle loss: A unified perspective of pair similarity optimization. In *CVPR*, 2020. 2

[10] Martin Sundermeyer, Maximilian Durner, En Yen Puang, Zoltan-Csaba Marton, Narunas Vaskevicius, Kai O. Arras,

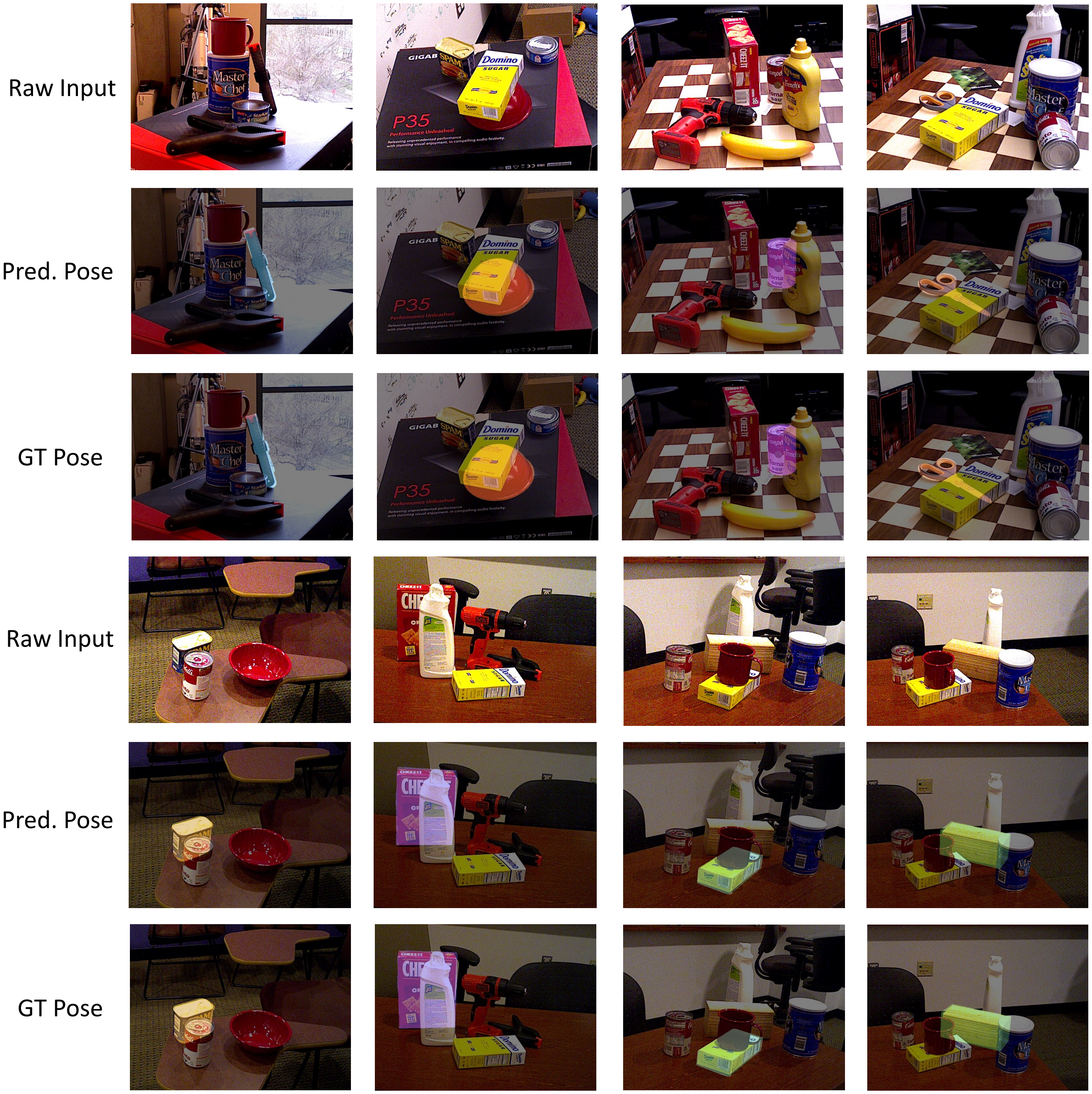

Figure 5. Visualization results of our method on YCB-V dataset.

and Rudolph Triebel. Multi-path learning for object pose estimation across domains. In *CVPR*, 2020. 2

[11] Hao Yu, Fu Li, Mahdi Saleh, Benjamin Busam, and Slobodan Ilic. Cofinet: Reliable coarse-to-fine correspondences for robust pointcloud registration. In *NeurIPS*, 2021. 2

[12] Hao Yu, Zheng Qin, Ji Hou, Mahdi Saleh, Dongsheng Li, Benjamin Busam, and Slobodan Ilic. Rotation-invariant transformer for point cloud matching. In *CVPR*, 2023. 1

[13] Heng Zhao, Shenxing Wei, Dahu Shi, Wenming Tan, Zheyang Li, Ye Ren, Xing Wei, Yi Yang, and Shiliang Pu. Learning symmetry-aware geometry correspondences for 6d object pose estimation. In *ICCV*, 2023. 2