SAM-6D: Segment Anything Model Meets Zero-Shot 6D Object Pose Estimation

Supplementary Material

CONTENT:

A. Supplementary Material for Instance Segmentation Model

A.1. Visible Ratio for Geometric Matching Score

In the Instance Segmentation Model (ISM) of our SAM-6D, we introduce a visible ratio r_{vis} to weight the reliability of the geometric matching score s_{geo}. Specifically, given an RGB crop $\mathcal{I}_m$ of a proposal m and the best-matched template $\mathcal{T}_{best}$ of the target object $\mathcal{O}$, along with their patch embeddings $\{\boldsymbol{f}^{patch}_{\mathcal{I}_m,j}\}_{j=1}^{N^{patch}_{\mathcal{I}_m}}$ and $\{\boldsymbol{f}^{patch}_{\mathcal{T}_{best},i}\}_{i=1}^{N^{patch}_{\mathcal{T}_{best}}}$, r_{vis} is calculated as the ratio of patches in $\mathcal{T}_{best}$ that can find a corresponding patch in $\mathcal{I}_m$, estimating the occlusion degree of $\mathcal{O}$ in $\mathcal{I}_m$. We can formulate the calculation of visible ratio r_{vis} as follows:

$$r_{vis} = \frac{1}{N^{patch}_{\mathcal{T}_{best}}} \sum_{i=1}^{N^{patch}_{\mathcal{T}_{best}}} r_{vis,i}, \tag{1}$$

where

$$r_{vis,i} = \begin{cases} 0 & \text{if } s_{vis,i} < \delta_{vis} \\ 1 & \text{if } s_{vis,i} \geq \delta_{vis} \end{cases},$$

and

$$s_{vis,i} = \max_{j=1,\dots,N^{patch}_{\mathcal{I}_m}} \frac{< \boldsymbol{f}^{patch}_{\mathcal{I}_m,j}, \boldsymbol{f}^{patch}_{\mathcal{T}_{best},i} >}{|\boldsymbol{f}^{patch}_{\mathcal{I}_m,j}| \cdot |\boldsymbol{f}^{patch}_{\mathcal{T}_{best},i}|}. \tag{2}$$

The constant threshold δ_{vis} is empirically set as 0.5 to determine whether the patches in $\mathcal{T}_{best}$ are occluded.

A.2. Template Selection for Object Matching

For each given target object, we follow [8] to first sample 42 well-distributed viewpoints defined by the icosphere primitive of Blender. Corresponding to these viewpoints, we select 42 fully visible object templates from the Physically-based Rendering (PBR) training images of the BOP benchmark [13] by cropping regions and masking backgrounds using the ground truth object bounding boxes and masks, respectively. These cropped and masked images then serve as the templates of the target object, which are used to calculate the object matching scores for all generated proposals. It's noted that these 42 templates can also be directly rendered using the pre-defined viewpoints.

A.3. Hyperparameter Settings

In the paper, we use SAM [6] based on ViT-H or FastSAM based on YOLOv8x as the segmentation model, and ViT-L of DINOv2 [10] as the description model. We utilize the publicly available codes for autonomous segmentation from SAM and FastSAM, with the hyperparameter settings displayed in Table 1.

A.4. More Quantitative Results

A.4.1 Detection Results

We compare our Instance Segmentation Model (ISM) with ZeroPose [2] and CNOS [8] in terms of 2D object detection in Table 2, where our ISM outperforms both methods owing to the meticulously crafted design of object matching score.

A.4.2 Effects of Model Sizes

We draw a comparison across different model sizes for both segmentation and description models on YCB-V dataset in Table 3, which indicates a positive correlation between larger model sizes and higher performance for both models.

Hyperparameter	Setting
(a) SAM [6]	
point_per_size	32
pred_iou_thresh	0.88
stability_score_thresh	0.85
stability_score_offset	1.0
box_nms_thresh	0.7
crop_n_layer	0
point_grids	None
min_mask_region_area	0
(b) FastSAM [14]	
iou	0.9
conf	0.05
max_det	200

Table 1. Hyperparameter Settings of (a) SAM [6] and (b) FastSAM [14] in their publicly available codes for autonomous segmentation.

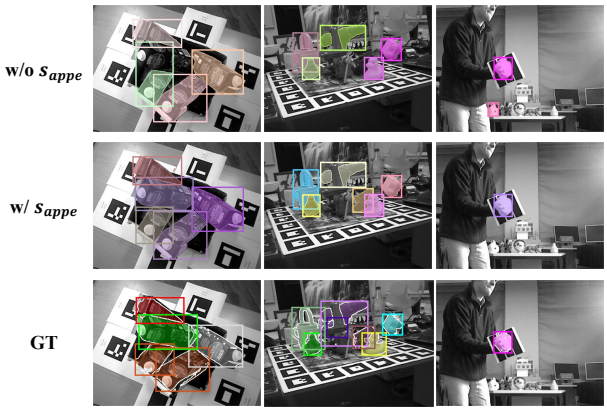

Figure 1. Qualitative results of our Instance Segmentation Model with or without the appearance matching score s_{appe}.

A.5. More Qualitative Results

A.5.1 Qualitative Comparisons on Appearance Matching Score

We visualize the qualitative comparisons of the appearance matching score s_{appe} in Fig. 1 to show its advantages in scoring the proposals w.r.t. a given object in terms of appearance.

A.5.2 Qualitative Comparisons on Geometric Matching Score

We visualize the qualitative comparisons of the geometric matching score s_{geo} in Fig. 2 to show its advantages in scoring the proposals w.r.t. a given object in terms of geometry, *e.g.*, object shapes and sizes.

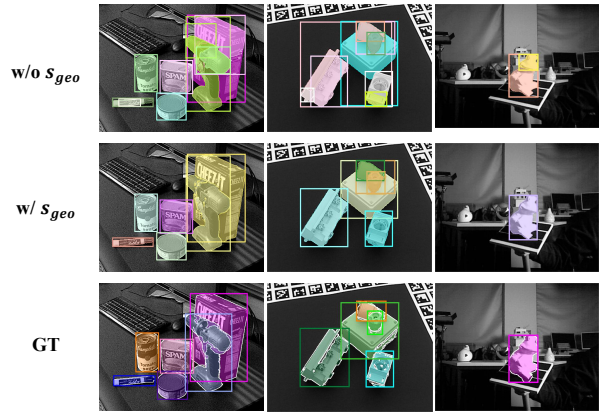

Figure 2. Qualitative results of our Instance Segmentation Model with or without the geometric matching score s_{geo}.

A.5.3 More Qualitative Comparisons with Existing Methods

To illustrate the advantages of our Instance Segmentation Model (ISM), we visualize in Fig. 3 the qualitative comparisons with CNOS [8] on all the seven core datasets of the BOP benchmark [13] for instance segmentation of novel objects. For reference, we also provide the ground truth masks, except for the ITODD and HB datasets, as their ground truths are not available.

B. Supplementary Material for Pose Estimation Model

B.1. Network Architectures and Specifics

B.1.1 Feature Extraction

In the Pose Estimation Model (PEM) of our SAM-6D, the Feature Extraction module utilizes the base version of the Visual Transformer (ViT) backbone [3], termed as ViT-Base, to process masked RGB image crops of observed object proposals or rendered object templates, yielding per-pixel feature maps.

Fig. 4 gives an illustration of the per-pixel feature learning process for an RGB image within the Feature Extraction module. More specifically, given an RGB image of the object, the initial step involves image processing, including masking the background, cropping the region of interest, and resizing it to a fixed resolution of 224×224. The object mask and bounding box utilized in the process can be sourced from the Instance Segmentation Model (ISM) for the observed scene image or from the renderer for the object template. The processed image is subsequently fed into ViT-Base to extract per-patch features using 12 attention blocks. The patch features from the third, sixth, ninth, and twelfth blocks are subsequently concatenated and passed through a fully-connected layer. They are then reshaped

Method	Segmentation Model	BOP Dataset							Mean
		LM-O	T-LESS	TUD-L	IC-BIN	ITODD	HB	YCB-V	
ZeroPose [2]	SAM [6]	36.7	30.0	43.1	22.8	25.0	39.8	41.6	34.1
CNOS [8]	FastSAM [14]	43.3	39.5	53.4	22.6	32.5	51.7	56.8	42.8
CNOS [8]	SAM [6]	39.5	33.0	36.8	20.7	31.3	42.3	49.0	36.1
SAM-6D	FastSAM [14]	46.3	**45.8**	**57.3**	24.5	**41.9**	**55.1**	**58.9**	**47.1**
SAM-6D	SAM [6]	**46.6**	43.7	53.7	**26.1**	39.3	53.1	51.9	44.9

Table 2. Object Detection results of different methods on the seven core datasets of the BOP benchmark [13]. We report the mean Average Precision (mAP) scores at different Intersection-over-Union (IoU) values ranging from 0.50 to 0.95 with a step size of 0.05.

Segmentation Model		Description Model		AP
Type	#Param	Type	#Param	
FastSAM-s	23 M	ViT-S	21 M	43.1
		ViT-L	300 M	54.0
FastSAM-x	138 M	ViT-S	21 M	48.9
		ViT-L	300 M	62.0
SAM-B	357 M	ViT-S	21 M	44.0
		ViT-L	300 M	55.8
SAM-L	1,188 M	ViT-S	21 M	47.2
		ViT-L	300 M	59.8
SAM-H	2,437 M	ViT-S	21 M	47.1
		ViT-L	300 M	60.5

Table 3. Quantitative comparisons on the model sizes of both segmentation and description models on YCB-V. We report the mean Average Precision (mAP) scores at different Intersection-over-Union (IoU) values ranging from 0.50 to 0.95 with a step size of 0.05.

Method	Segmentation Model	Server	Time (s)
CNOS [8]	FastSAM [14]	Tesla V100	0.22
CNOS [8]		DeForce RTX 3090	0.23
SAM-6D		DeForce RTX 3090	0.45
CNOS [8]	SAM [6]	Tesla V100	1.84
CNOS [8]		DeForce RTX 3090	2.35
SAM-6D		DeForce RTX 3090	2.80

Table 4. Runtime comparisons of different methods for instance segmentation of novel objects. The reported time is the average per-image processing time across the seven core datasets of the BOP benchmark [13].

and bilinearly interpolated to match the input resolution of 224×224 with 256 feature channels. Further specifics about the network can be found in Fig. 4.

For a cropped observed RGB image, the pixel features within the mask are ultimately chosen to correspond to the point set transformed from the masked depth image. For object templates, the pixels within the masks across views are finally aggregated, with the surface point of per pixel known from the renderer. Both point sets of the proposal and the target object are normalized to fit a unit sphere by dividing by the object scale, effectively addressing the variations in object scales.

We use two views of object templates for training, and 42 views for evaluation as CNOS [8], which is the standard setting for the results reported in this paper.

B.1.2 Coarse Point Matching

In the Coarse Point Matching module, we utilize T^c Geometric Transformers [12] to model the relationships between the sparse point set $\mathcal{P}_m^c \in \mathbb{R}^{N_m^c\times3}$ of the observed object proposal m and the set $\mathcal{P}_o^c \in \mathbb{R}^{N_o^c\times3}$ of the target object $\mathcal{O}$. Their respective features $\boldsymbol{F}_m^c$ and $\boldsymbol{F}_o^c$ are thus improved to their enhanced versions $\tilde{\boldsymbol{F}}_m^c$ and $\tilde{\boldsymbol{F}}_o^c$. Each of these enhanced feature maps also includes the background token. An additional fully-connected layer is applied to the features both before and after the transformers. In this paper, we use the upper script 'c' to indicate variables associated with the Coarse Point Matching module, and the lower scripts 'm' and 'o' to distinguish between the proposal and the object.

During inference, we compute the soft assignment matrix $\tilde{\mathcal{A}}^c \in \mathbb{R}^{(N_m^c+1)\times(N_o^c+1)}$, and obtain two binary-value matrices $\boldsymbol{M}_m^c \in \mathcal{R}^{N_m^c\times1}$ and $\boldsymbol{M}_o^c \in \mathcal{R}^{N_o^c\times1}$, denoting whether the points in $\mathcal{P}_m^c$ and $\mathcal{P}_o^c$ correspond to the background, owing to the design of background tokens; '0' indicates correspondence to the background, while '1' indicates otherwise. We then have the probabilities $\boldsymbol{P}^c \in \mathbb{R}^{N_m^c\times N_o^c}$ to indicate the matching degree of the $N_m^c \times N_o^c$ point pairs between $\mathcal{P}_m^c$ and $\mathcal{P}_o^c$, formulated as follows:

$$\boldsymbol{P}^c = \boldsymbol{M}_m^c \cdot (\tilde{\mathcal{A}}^c[1:,1:])^\gamma \cdot \boldsymbol{M}_o^{cT}, \tag{3}$$

where γ is used to sharpen the probabilities and set as 1.5. The probabilities of points that have no correspondence, whether in $\mathcal{P}_m^c$ or $\mathcal{P}_o^c$, are all set to 0. Following this, the probabilities $\boldsymbol{P}^c$ are normalized to ensure their sum equals 1, and act as weights used to randomly select 6,000 triplets of point pairs from the total pool of $N_m^c \times N_o^c$ pairs. Each

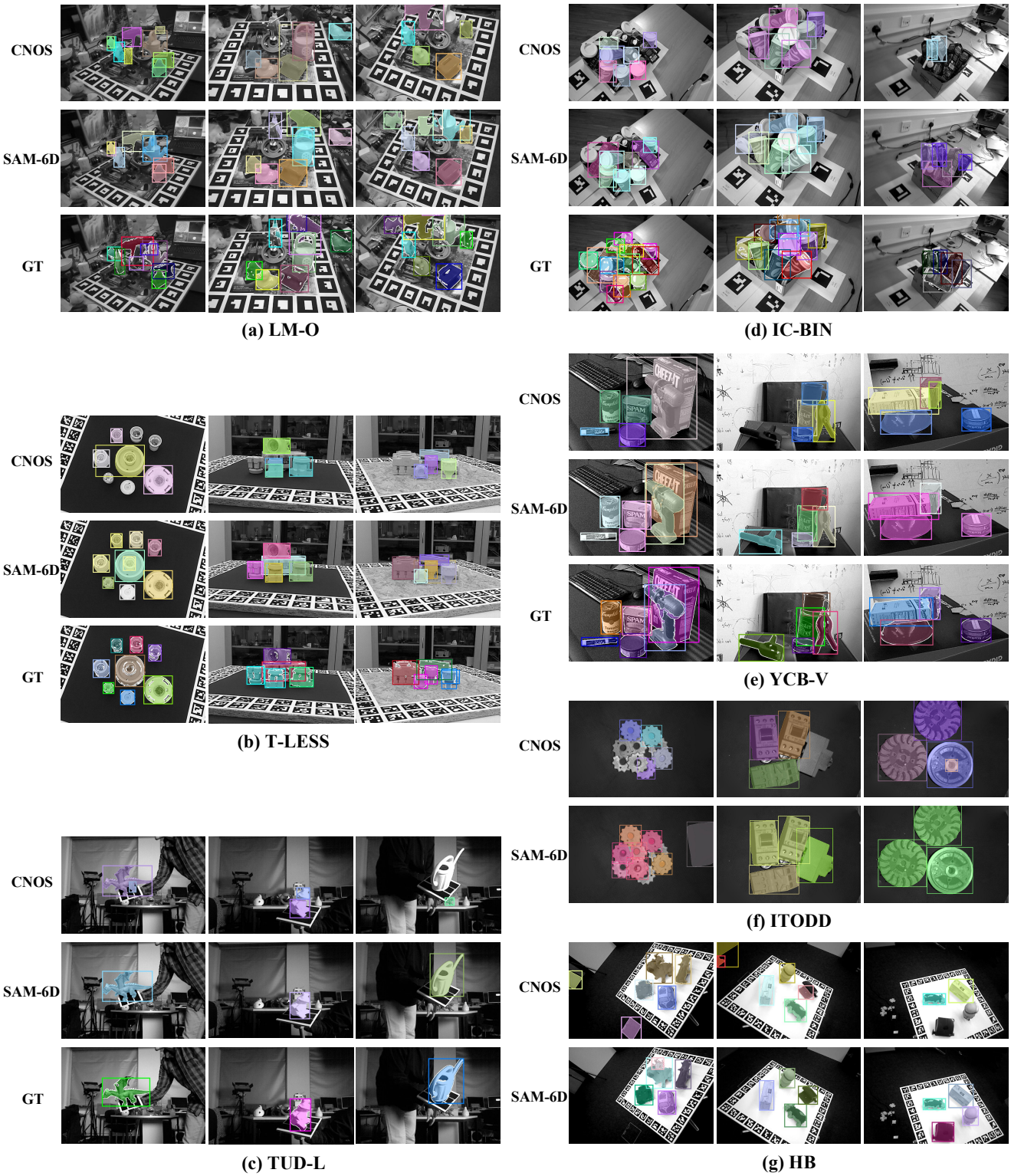

Figure 3. Qualitative results on the seven core datasets of the BOP benchmark [13] for instance segmentation of novel objects.

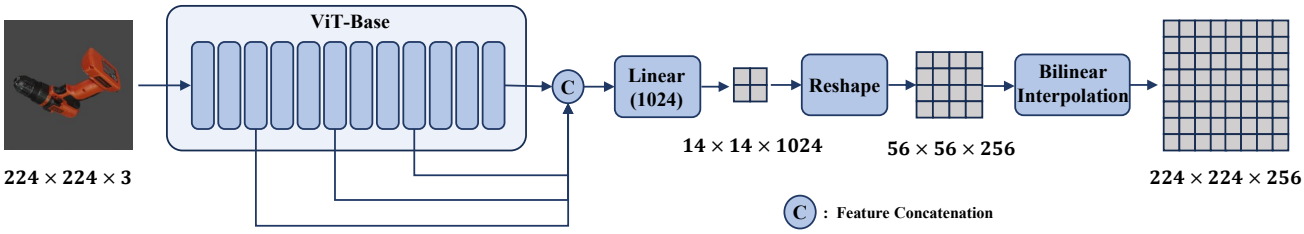

Figure 4. An illustration of the per-pixel feature learning process for an RGB image within the Feature Extraction module of the Pose Estimation Model.

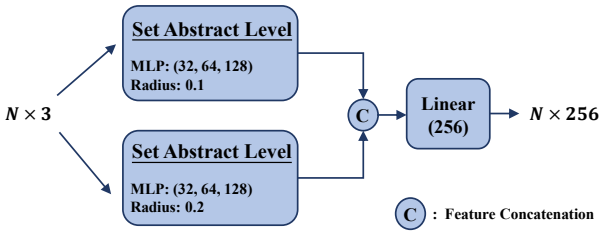

Figure 5. An illustration of the positional encoding for a point set with N points within the Fine Point Matching Module of the Pose Estimation Model.

triplet, which consists of three point pairs, is utilized to calculate a pose using SVD, along with a distance between the point pairs based on the computed pose. Through this procedure, a total of 6,000 pose hypotheses are generated, and to minimize computational cost, only the 300 poses with the smallest point pair distances are selected. Finally, the initial pose for the Fine Point Matching module is determined from these 300 poses, with the pose that has the highest pose matching score being selected.

In the Coarse Point Matching module, we set $T^c = 3$ and $N_m^c = N_o^c = 196$, with all the feature channels designated as 256. The configurations of the Geometric Transformers adhere to those used in [12].

B.1.3 Fine Point Matching

In the Fine Point Matching module, we utilize T^f Sparse-to-Dense Point Transformers to model the relationships between the dense point set $\mathcal{P}_m^f \in \mathbb{R}^{N_m^f \times 3}$ of the observed object proposal m and the set $\mathcal{P}_o^f \in \mathbb{R}^{N_o^f \times 3}$ of the target object $\mathcal{O}$. Their respective features $\boldsymbol{F}_m^f$ and $\boldsymbol{F}_o^f$ are thus improved to their enhanced versions $\tilde{\boldsymbol{F}}_m^f$ and $\tilde{\boldsymbol{F}}_o^f$. Each of these enhanced feature maps also includes the background token. An additional fully-connected layer is applied to the features both before and after the transformers. We use the upper script 'f' to indicate variables associated with the Fine Point Matching module, and the lower scripts 'm' and 'o' to distinguish between the proposal and the object.

Different from the coarse module, we condition both features $\boldsymbol{F}_m^f$ and $\boldsymbol{F}_o^f$ before applying them to the transformers by adding their respective positional encodings, which are learned via a multi-scale Set Abstract Level [11] from $\mathcal{P}_m^f$ transformed by the initial pose and $\mathcal{P}_o^f$ without transformation, respectively. The used architecture for positional encoding learning is illustrated in Fig. 5. For more details, one can refer to [11].

Another difference from the coarse module is the type of transformers used. To handle dense relationships, we design the Sparse-to-Dense Point Transformers, which utilize Geometric Transformers [12] to process sparse point sets and disseminate information to dense point sets via Linear Cross-attention layers [4, 5]. The configurations of the Geometric Transformers adhere to those used in [12]; the point numbers of the sampled sparse point sets are all set as 196. The Linear Cross-attention layer enables attention along the feature dimension, and details of its architecture can be found in Fig. 6; for more details, one can refer to [4, 5].

During inference, similar to the coarse module, we compute the soft assignment matrix $\tilde{\mathcal{A}}^f \in \mathbb{R}^{(N_m^f+1)\times(N_o^f+1)}$, and obtain two binary-value matrices $\boldsymbol{M}_m^f \in \mathcal{R}^{N_m^f \times 1}$ and $\boldsymbol{M}_o^f \in \mathcal{R}^{N_o^f \times 1}$. We then formulate the probabilities $\boldsymbol{P}^f \in \mathbb{R}^{N_m^f \times N_o^f}$ as follows:

$$\boldsymbol{P}^f = \boldsymbol{M}_m^f \cdot (\tilde{\mathcal{A}}^f[1:, 1:]) \cdot \boldsymbol{M}_o^{fT}. \tag{4}$$

Based on $\boldsymbol{P}^f$, we search for the best-matched point in $\mathcal{P}_o^f$ for each point in $\mathcal{P}_m^f$, assigned with the matching probability. The final object pose is then calculated using a weighted SVD, with the matching probabilities of the point pairs serving as the weights.

Besides, we set $T^f = 3$ and $N_m^f = N_o^f = 2,048$, with all the feature channels designated as 256. During training, we follow [7] to obtain the initial object poses by augmenting the ground truth ones with random noises.

B.2. Training Objectives

We use InfoNCE loss [9] to supervise the learning of attention matrices for both coarse and fine modules. Specifically, given two point sets $\mathcal{P}_m \in \mathbb{R}^{N_m \times 3}$ and $\mathcal{P}_o \in \mathbb{R}^{N_o \times 3}$, along with their enhanced features $\tilde{\boldsymbol{F}}_m$ and $\tilde{\boldsymbol{F}}_o$, which are

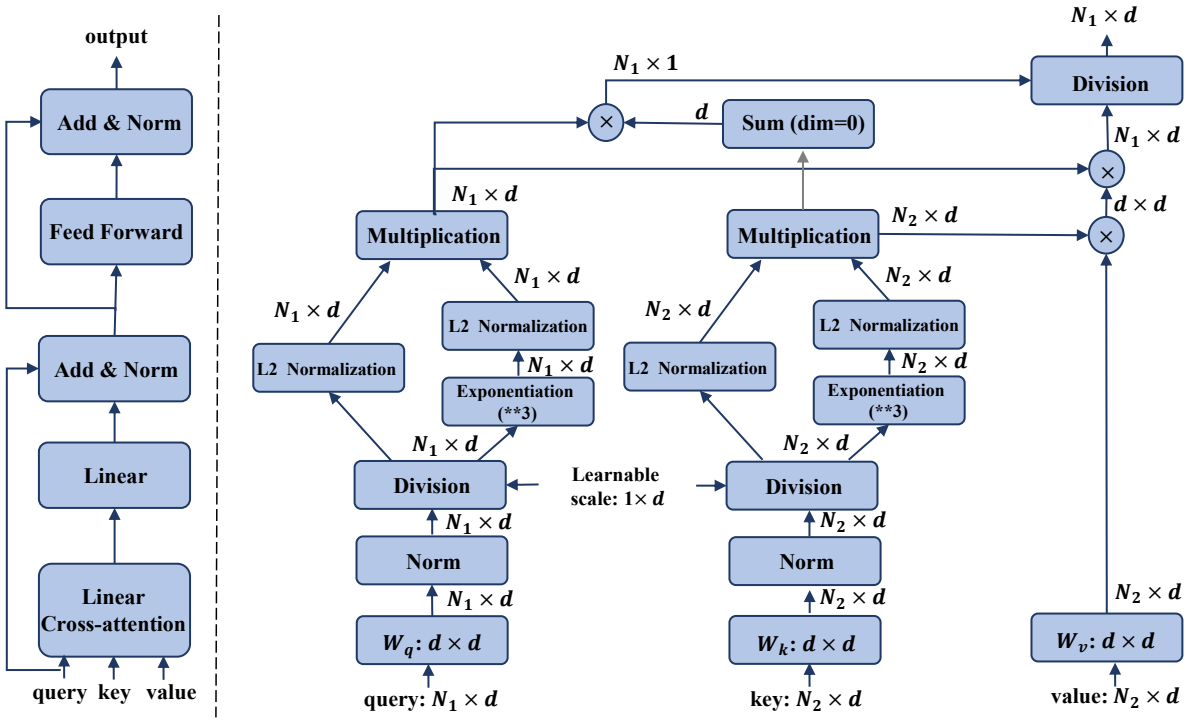

Figure 6. Left: The structure of Linear Cross-attention layer. Right: The structure of Linear Cross-attention.

learnt via the transformers and equipped with background tokens, we compute the attention matrix $\mathcal{A} = \tilde{\boldsymbol{F}}_m \times \tilde{\boldsymbol{F}}_o^T \in \mathbb{R}^{(N_m+1)\times(N_o+1)}$. Then $\mathcal{A}$ can be supervised by the following objective:

$$\mathcal{L} = \texttt{CE}(\mathcal{A}[1:,:], \hat{\mathcal{Y}}_m) + \texttt{CE}(\mathcal{A}[:,1:]^T, \hat{\mathcal{Y}}_o), \quad (5)$$

where $\texttt{CE}(\cdot,\cdot)$ denotes the cross-entropy loss function. $\hat{\mathcal{Y}}_m \in \mathbb{R}^{N_m}$ and $\hat{\mathcal{Y}}_o \in \mathbb{R}^{N_o}$ denote the ground truths for $\mathcal{P}_m$ and $\mathcal{P}_o$. Given the ground truth pose $\hat{\boldsymbol{R}}$ and $\hat{\boldsymbol{t}}$, each element y_m in $\hat{\mathcal{Y}}_m$, corresponding to the point $\boldsymbol{p}_m$ in $\mathcal{P}_m$, could be obtained as follows:

$$y_m = \begin{cases} 0 & \text{if } d_{k^*} \geq \delta_{dis} \\ k^* & \text{if } d_{k^*} < \delta_{dis} \end{cases}, \quad (6)$$

where

$$k^* = \text{Argmin}_{k=1,\dots,N_m} ||\hat{\boldsymbol{R}}(\boldsymbol{p}_m - \hat{\boldsymbol{t}}) - \boldsymbol{p}_{o,k}||_2,$$

and

$$d_{k^*} = ||\hat{\boldsymbol{R}}(\boldsymbol{p}_m - \hat{\boldsymbol{t}}) - \boldsymbol{p}_{o,k^*}||_2.$$

k^* is the index of the closest point $\boldsymbol{p}_{o,k^*}$ in $\mathcal{P}_o$ to $\boldsymbol{p}_m$, while d_{k^*} denotes the distance between $\boldsymbol{p}_m$ and $\boldsymbol{p}_{o,k^*}$ in the object coordinate system. δ_{dis} is a distance threshold determining whether the point $\boldsymbol{p}_m$ has the correspondence in $\mathcal{P}_o$; we set δ_{dis} as a constant 0.15, since both $\mathcal{P}_m$ and $\mathcal{P}_o$ are normalized to a unit sphere. The elements in $\hat{\mathcal{Y}}_o$ are also generated in a similar way.

We employ the objective (5) upon all the transformer blocks of both coarse and fine point matching modules, and thus optimize the Pose Estimation Model by solving the following problem:

$$\min \sum_{l=1,\dots,T_c} \mathcal{L}_l^c + \sum_{l=1,\dots,T_f} \mathcal{L}_l^f. \quad (7)$$

where for the loss $\mathcal{L}$ in Eq. (5), we use the upper scripts 'c' and 'f' to distinguish between the losses in the coarse and fine point matching modules, respectively, while the lower script 'l' denotes the sequence of the transformer blocks in each module.

B.3. More Quantitative Results

B.3.1 Effects of The View Number of Templates

We present a comparison of results using different views of object templates in Table 5. As shown in the table, results with only one template perform poorly as a single view cannot fully depict the entire object. With an increase in the number of views, performance improves. For consistency with our Instance Segmentation Model and CNOS [8], we utilize 42 views of templates as the default setting in the main paper.

# View	1	2	8	16	42
AR	21.8	62.7	83.9	84.1	84.5

Table 5. Pose estimation results with different view numbers of object templates on YCB-V. We report the mean Average Recall (AR) among VSD, MSSD and MSPD.

B.3.2 Comparisons with OVE6D

OVE6D [1] is a classical method for zero-shot pose estimation based on image matching, which first constructs a codebook from the object templates for viewpoint rotation retrieval and subsequently regresses the in-plane rotation. When comparing our SAM-6D with OVE6D using their provided segmentation masks (as shown in Table 6), SAM-6D outperforms OVE6D on LM-O dataset, without the need for using Iterative Closest Point (ICP) algorithm for post-optimization.

Method	LM-O
OVE6D [1]	56.1
OVE6D with ICP [1]	72.8
SAM-6D (Ours)	**74.7**

Table 6. Quantitative results of OVE6D [1] and our SAM-6D on LM-O dataset. The evaluation metric is the standard ADD(-S) for pose estimation. SAM-6D is evaluated with the same masks provided by [1].

B.4. More Qualitative Comparisons with Existing Methods

To illustrate the advantages of our Pose Estimation Model (ISM), we visualize in Fig. 7 the qualitative comparisons with MegaPose [7] on all the seven core datasets of the BOP benchmark [13] for pose estimation of novel objects. For reference, we also present the corresponding ground truths, barring those for the ITODD and HB datasets, as these are unavailable.

References

[1] Dingding Cai, Janne Heikkilä, and Esa Rahtu. Ove6d: Object viewpoint encoding for depth-based 6d object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6803–6813, 2022. 7

[2] Jianqiu Chen, Mingshan Sun, Tianpeng Bao, Rui Zhao, Liwei Wu, and Zhenyu He. 3d model-based zero-shot pose estimation pipeline. *arXiv preprint arXiv:2305.17934*, 2023. 1, 3

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2

[4] Dongchen Han, Xuran Pan, Yizeng Han, Shiji Song, and Gao Huang. Flatten transformer: Vision transformer using focused linear attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5961–5971, 2023. 5

[5] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pages 5156–5165. PMLR, 2020. 5

[6] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 1, 2, 3

[7] 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 *Proceedings of the 6th Conference on Robot Learning (CoRL)*, 2022. 5, 7

[8] Van Nguyen Nguyen, Thibault Groueix, Georgy Ponimatkin, Vincent Lepetit, and Tomas Hodan. Cnos: A strong baseline for cad-based novel object segmentation. *arXiv preprint arXiv:2307.11067*, 2023. 1, 2, 3, 6

[9] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 5

[10] 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. 1

[11] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 5

[12] Zheng Qin, Hao Yu, Changjian Wang, Yulan Guo, Yuxing Peng, and Kai Xu. Geometric transformer for fast and robust point cloud registration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11143–11152, 2022. 3, 5

[13] Martin Sundermeyer, Tomáš Hodaň, Yann Labbe, Gu Wang, Eric Brachmann, Bertram Drost, Carsten Rother, and Jiří Matas. Bop challenge 2022 on detection, segmentation and pose estimation of specific rigid objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2784–2793, 2023. 1, 2, 3, 4, 7, 8

[14] Xu Zhao, Wenchao Ding, Yongqi An, Yinglong Du, Tao Yu, Min Li, Ming Tang, and Jinqiao Wang. Fast segment anything. *arXiv preprint arXiv:2306.12156*, 2023. 2, 3

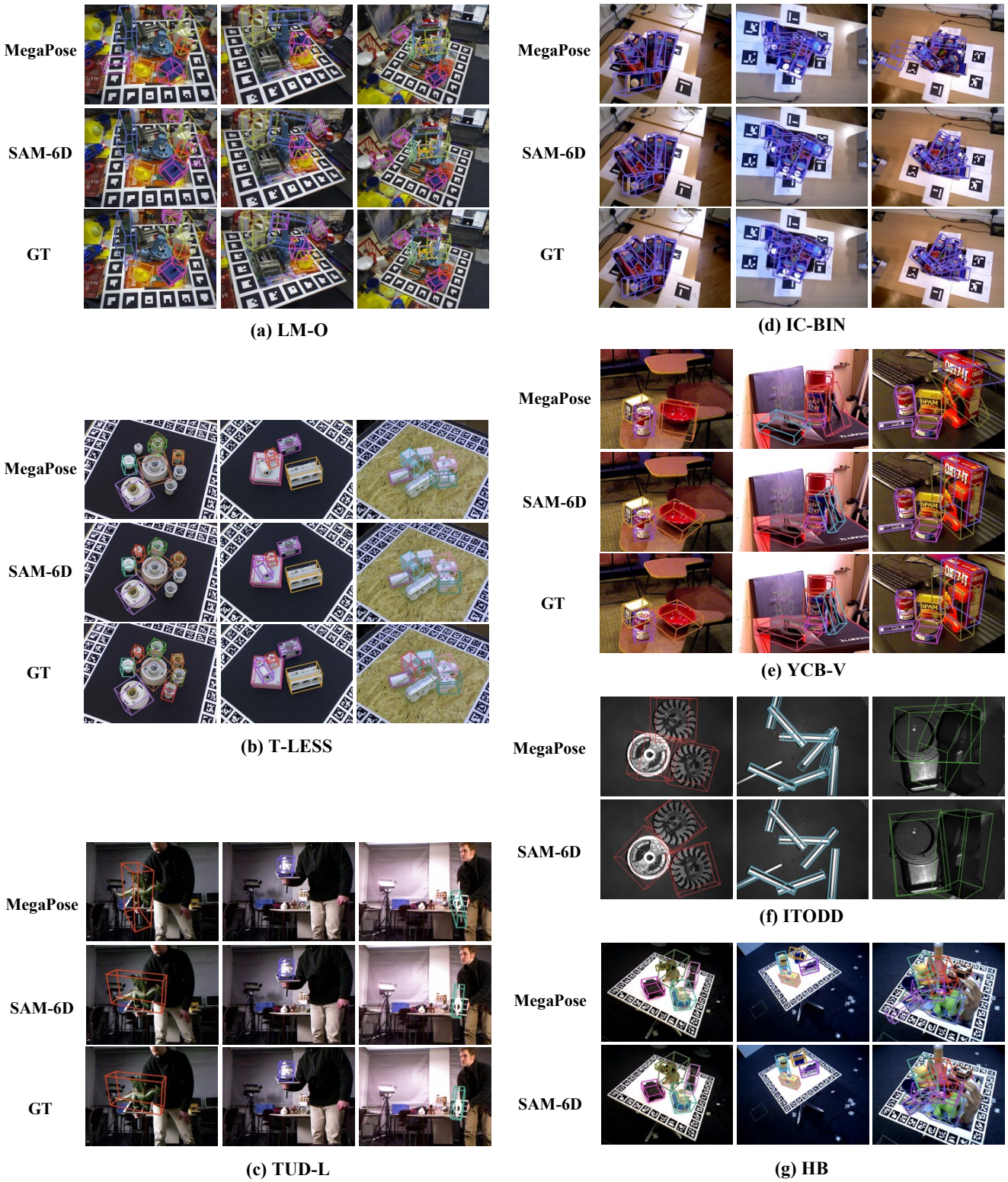

Figure 7. Qualitative results on the seven core datasets of the BOP benchmark [13] for pose estimation of novel objects.