RayPose: Ray Bundling Diffusion for Template Views in Unseen 6D Object Pose Estimation

Supplementary Material

1. More Analysis

1.1. Effects of number of templates.

We train our model on the LM-O dataset to examine the impact of the number of templates on our method. Table 1 compares performance and training time across different template counts. Increasing the number of templates enhances performance, with AR improving from 55.79 using 2 templates to 70.03 with 32 templates. However, this comes at the cost of higher training time per epoch. To balance accuracy, computational efficiency, and memory usage in our case, we select 8 templates as our default setting. Nonetheless, the upward trend in performance suggests that our method could further benefit from increasing the number of templates.

No. template	LMO (AR)	Training Time (h/Epoch)
2	55.79	28
4	59.06	37
8	**65.58**	**48**
16	68.42	69
32	70.03	125

Table 1. Performance v.s. training time cost comparison with different numbers of templates on LM-O dataset. The bold is our default setting.

1.2. Effects of number diffusion time steps.

Table 2 compares the performance and inference time across different numbers of diffusion steps. As the number of steps decreases, inference time improves linearly while the performance increases marginally. We apply 20 diffusion steps, achieving a balance between accuracy and efficiency (714 ms/frame), making it a practical choice for real-world applications.

No. Steps	AR	Inference Time (ms/frame)
100	67.88	3571
30	65.73	1075
20	**65.58**	**714**
10	64.09	357
1	63.27	37

Table 2. Performance v.s. inference time comparison with different numbers of diffusion steps on LM-O dataset. The bold is our default setting.

2. Implementation Details.

We use 8 DiT-based self-attention blocks for the multiview fuser and 8 self and cross attention blocks in the diffusion transformer decoder. We train our model with around 480 GPU hours on RTX4090 GPUs. Learning rate $lr = 1e-4$ with step scheduler, we set $\gamma = 0.99$ and step size 10000. We train the model with a batch size of 16 with 8 templates each for each batch. During training, we balance the rotation and translation map with $\lambda_{\text{rot}} = \lambda_{\text{trans}} = 0.5$. Specifically, we set the reconstruction loss factor λ_{recon} as 1.0 for both rotation and translation maps. For the rest of the weights, we set $\lambda_t = 0.5$, $\lambda_{reg} = 0.2$ and $\lambda_{cos} = 0.5$. For both the rotation and translation maps, we use the same resolution of 16×16, and adopt the frozen DINOv2 (S/14) as image encoder.

To ensure realistic training, we sample posed template images from a physically based rendering dataset. We preprocess these images by cropping the object from the scene. To ensure sufficient visual overlap between the templates and the query image, we retain only those crops with a visibility ratio greater than 0.8. To achieve an even distribution of object poses across the sampled templates, we first randomly generate 256 candidate viewpoints by varying both the camera radius and viewing angles. We then select the object crops whose poses are closest to these sampled viewpoints. To enhance the model's robustness to bounding box inaccuracies, we apply random augmentations to the bounding boxes during training.