Supplementary Material of Generalizable Single-view Object Pose Estimation by Two-side Generating and Matching

Yujing Sun[1*], Caiyi Sun[2*], Yuan Liu[1*], Yuexin Ma[2], Siu Ming Yiu[1]
[1]The University of Hong Kong, [2]ShanghaiTech University
{yjsun@cs,smyiu@cs, yuanly@connect}.hku.hk, scy639@outlook.com, mayuexin@shanghaitech.edu.cn

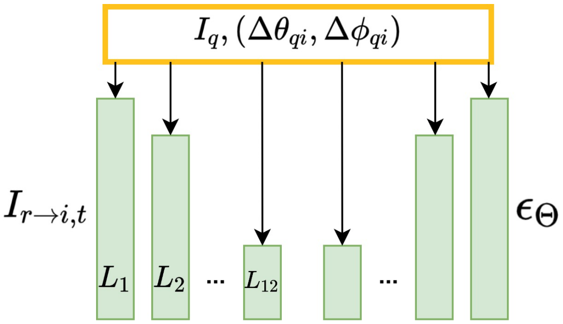


Figure 1. The structure of the UNet in Zero123. We denote the l-th layer of the UNet encoder as L_l, where $l \in \{1, 2, ..., 12\}$. Then the calculation of l-th layer can be formulated by $L_l(I_{r\to i,t}, I_q, \Delta\theta_{qi}, \Delta\phi_{qi})$.

1. Accelerating inference

In the light version method which is discussed in Section 4.4 of the main paper, we incorporate several techniques and strategies to accelerate the inference. The detailed description of these approaches is as follows.

1.1. Feature reuse

Previous works [7, 12] have shown that the early layers in the encoder of the UNet focus more on the feature extraction, and the encoder features change gently across different time steps. Similar to that, in our method, we observe that for the same conditioned image and noisy latent, the early layers of the UNet encoder share similarity across adjacent viewpoints. Thus, when calculating the matching loss of pose candidates, for each i, we propose to share the same encoder feature across adjacent poses. As illustrated in Figure 1, we denote the l-th layer of the UNet encoder as L_l, where $l \in \{1, 2, ..., 12\}$. Then the calculation of l-th layer can be formulated by $L_l(I_{r\to i,t}, I_q, \Delta\theta_{qi}, \Delta\phi_{qi})$. As our feature reuse is applied to every (I_r, I_q) and every intermediate viewpoint i independently, we omit the I_q and the terms containing i or r here, simplifying the expression to $L_l(I_q, \theta_q, \phi_q)$. We further use $\theta_q^{(k)}, \phi_q^{(k)}$ to represent the k-th pose candidate, so the output feature of l-th layer can be formulated as:

$$L_l(I_q, (\theta_q^{(k)}, \phi_q^{(k)})) \tag{1}$$

We divide the upper hemisphere along into 4 grids where each grid occupies δazimuth $= \frac{360^\circ}{4} = 90^\circ$ and δelevation $= 90^\circ$. Those poses that fall into the same grid are regarded as nearby viewpoints, and the pose candidate closest to the grid center is referred to as the center pose. When searching for θ_q, ϕ_q, we firstly compute the matching loss of each grid's center pose in the same way as usual, where the calculations are performed across the entire U-Net. During this process, the features of the early N_r layers will be cached for subsequent reuse, with N_r meaning the number of layers we apply feature reuse.

Once the feature cache is initialized, we reuse the cache in the calculation of the remaining pose candidates. In other words, we approximate the Expression 1 by

$$L_l(I_q, center\_pose(\theta_q^{(k)}, \phi_q^{(k)})), \tag{2}$$

where $center\_pose(\cdot)$ is a function that returns the center pose of the grid in which the input pose falls.

We find that $N_r = 6$ yields a good balance between accuracy and speed. By applying this technique, the computational time of our method decreases to 0.8 times the original duration, with only a slight reduction in $Racc$@15 and $Racc$@30 by 0.01 and 0.02, respectively.

1.2. Intermediate viewpoints pruning

As the inference time approximately scales linearly with the number of intermediate viewpoints, we propose a strategy to "prune" away some intermediate viewpoints.

[*] denotes joint first author

We hand-picked the strategy based on the trade-off between speed and accuracy. The final strategy is as follows: (1). We define $distance((\theta_1, \phi_1), (\theta_2, \phi_2)) = \sqrt{(\theta_r - \theta_i)^2 + (\phi_r - \phi_i)^2}$. We firstly discard the loss terms associated with viewpoints near I_r, specifically those for which $distance((\theta_r, \phi_r), (\theta_i, \phi_i)) < TH1$. Instead, we use a single loss term

$$w || \epsilon_\Theta(I_{r,t} | I_q, \Delta\theta_{qi}, \Delta\phi_{qi}) - \epsilon \,||_2^2 \tag{3}$$

to replace the discarded terms, where w represents the number of discarded viewpoints. Essentially, we approximate $||\epsilon_\Theta(I_{r\to i,t} | I_q, \Delta\theta_{qi}, \Delta\phi_{qi}) - \epsilon \,||_2^2$ by Expression 3 for each discarded i. (2). We further discard the viewpoints that are significantly distant from (θ_r, ϕ_r), i.e., those for which $distance((\theta_r, \phi_r), (\theta_i, \phi_i)) > TH2$. The rationale behind this is that the generated view $I_{r\to i}$ tends to have poor quality, which might contribute less to solving Eq. (5) in the main paper.

We find that the thresholds of $TH1 = 45°$ and $TH2 = 145°$ provide a good balance between accuracy and speed. Under this configuration, we reduce the number of intermediate viewpoints from 64 to 43. By applying this technique, the computational time of our method decreases to 0.69 times the original duration, with no reduction in $Racc@15$ and only a minor drop of 0.02 in $Racc@30$.

1.3. Difficulty-guided router

Given an image pair, we firstly identify the "difficulty" (which can also be understood as viewpoint change or similarity) for each pair. Then, for a pair with low difficulty, we perform naive matching proposed in E2VG, which can be completed in 150ms. For a pair with high difficulty, we route it to our two-side matching method. For a pair with moderate difficulty, we also apply our two-side matching but with a reduced M. We use the number of LoFTR matching points to determine the difficulty level.

1.4. Quantitative result of the light version method

After integrating all the aforementioned acceleration techniques into our method and adopting a light configuration where M is reduced to $\frac{1}{2}$, we obtain our light version method, discussed in Section 4.4 of the main paper. The quantitative results are shown in Table 1.The light version method can process a query image in 1.12 seconds while still outperforms other baseline methods in accuracy by a large margin. It should also be noted that our method can be easily parallelized by distributing the matching loss calculation of different candidates across multiple GPUs. This parallelization can reduce the total processing time approximately by a factor equal to the number of GPUs. With multiple GPUs, we can process query images in real-time.

Although our method is slower than some baseline methods with a single GPU, accuracy is much more important than efficiency especially in non-real-time applications like reconstruction from unposed views and offline AR. We showcase in the supplementary video that we can generate AR videos using accurate poses while the baseline method fails to estimate camera poses correctly.

Table 1. The comparison of inference time for each query image. We present results on the NAVI dataset.

Method	Inference time (s)	$Acc@15°$	$Acc@30°$
IDPose	27.52	10.09	36.66
E2VG(N=64)	0.13	42.69	64.21
Ours	7.84	55.32	82.14
Ours(Light)	1.12	54.28	78.93

Table 2. Ablation studies on the number of the Monte-Carlo sampling M in IDPose on a 30% subset of NAVI.

Method	$Acc@15°$	$Acc@30°$
IDPose($M = 16$)	15.11	38.82
IDPose($M = 32$)	15.07	37.95
IDPose($M = 64$)	15.43	38.57
Ours	56.84	83.16

2. More ablation analysis

The matching scheme adopted by IDPose can be considered a special case of our two-side matching where the reference set contains only I_r. In other words, they estimate the pose by minimizing:

$$\frac{1}{M} \sum_{j}^{M} || \epsilon_\Theta(I_{r,t} | I_q, \Delta\theta_{qi}, \Delta\phi_{qi}) - \epsilon^{(j)} \,||_2^2, \tag{4}$$

In our main paper we have present the result of IDPose where $M = 16$ which is their standard configuration. Here we additionally present the result with a larger M for IDPose. For computational reasons, we perform this experiment on a 30% subset of NAVI. As shown in Table 2, there is no obvious enhancement in performance as M increases from 16 to 64. This shows that simply increasing M in the naive matching can not lead to better performance.

3. Stability and robustness

3.1. Robustness to artifacts from diffusion models

For diffusion-based methods, the performance of the generative diffusion model can affect the accuracy of pose estimation. Typically, our method is robust to minor artifacts generated by the diffusion model, as shown in Figure 4. Due to the two-sided matching approach, our method exhibits greater robustness under such conditions compared to other diffusion-based methods.

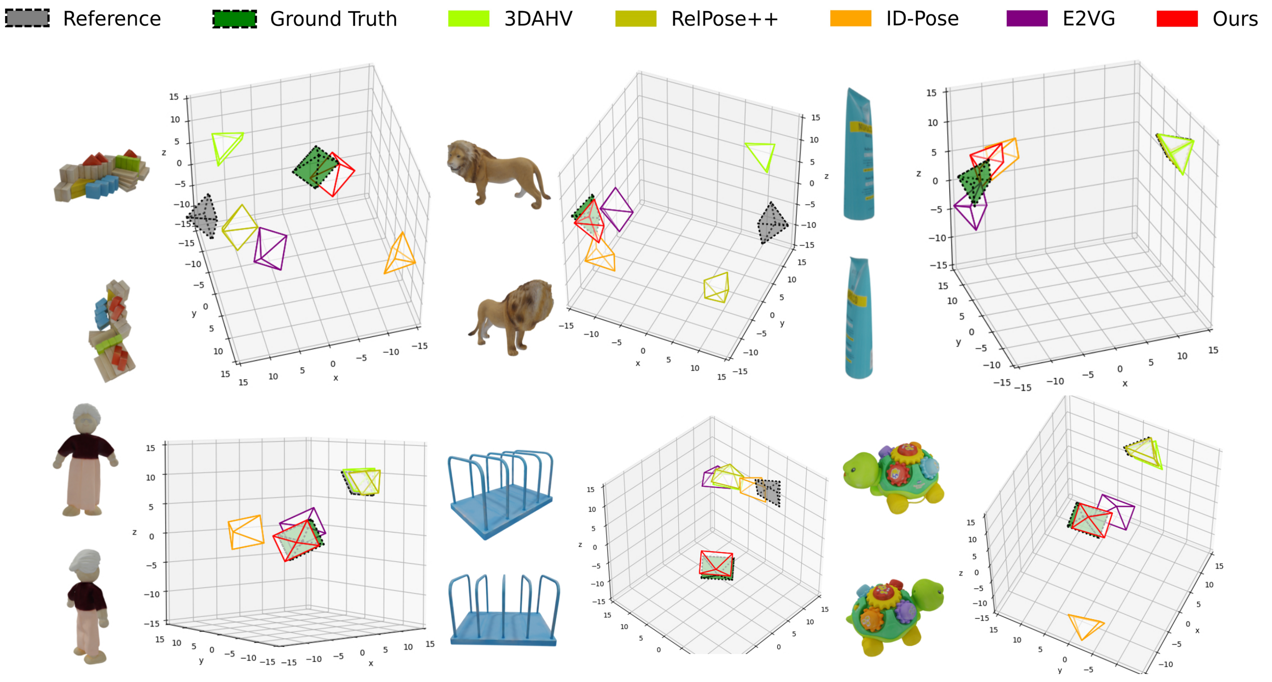


Figure 2. Visual Comparison on the GSO dataset [4].

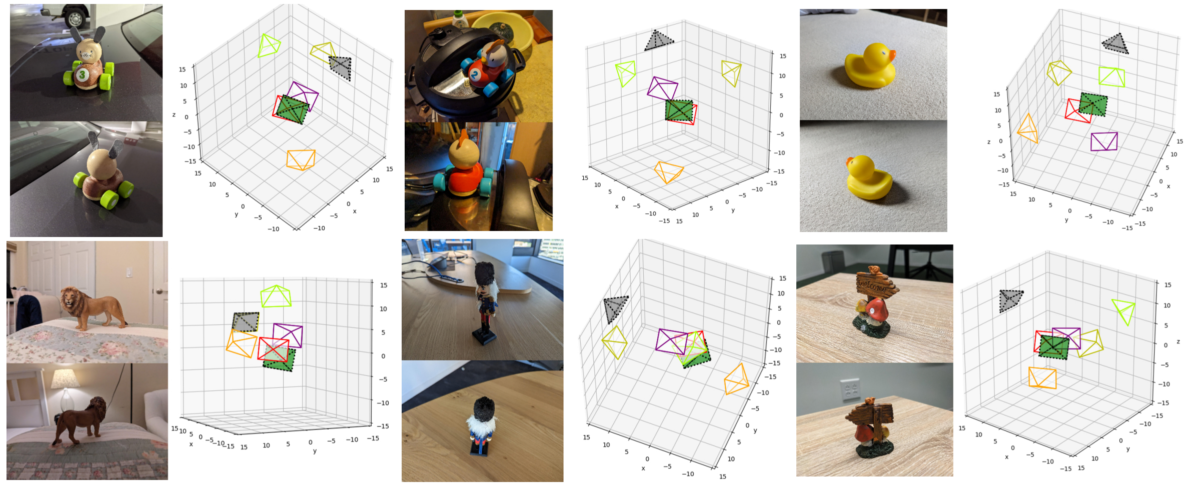


Figure 3. Visual Comparison on the NAVI dataset [6].

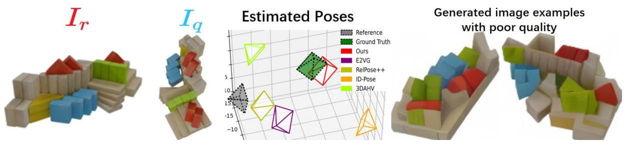


Figure 4. A visual example illustrating that our method is more robust to minor artifacts generated by the diffusion model compared to other diffusion-based methods.

To show that our method performs more stably on different objects than several key baseline methods, we report the minimum and variance of rotation accuracy at 30° across all objects of NAVI in Table 3. Our minimum accuracy is the highest with the lowest variance, indicating that our method is more stable than baselines.

Table 3. The minimum and variance of rotation accuracy at 30° across all objects of NAVI.

Method	Min of $Acc@30^\circ(\%)\uparrow$	Variance of $Acc@30^\circ\downarrow$
Relpose++	5.00	0.075
3DAHV	10.00	0.097
IDPose	0.00	0.089
E2VG	25.00	0.050
Ours	**38.89**	**0.033**

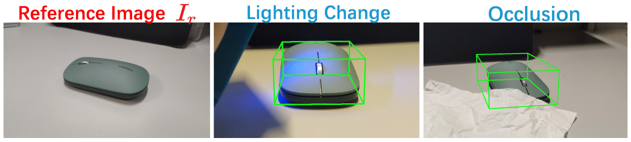


Figure 5. An example illustrating that our method can handle moderate lighting changes and occlusions.

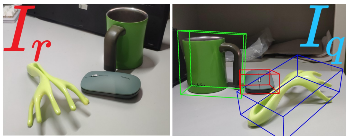


Figure 6. Dealing with multiple objects.

3.2. Robustness under complex scenes

Our method can handle moderate lighting changes and occlusions, as shown in the examples in Figure 5. However, extreme occlusions or lighting changes can affect our performance, which also affects other baseline methods. In this work, we focus on sparse view setting and is not specifically designed to address these challenges. There are works that can tackle such challenges such as pixel-wise voting. It is promising to combine these techniques with our methods to improve robustness in future works.

4. Scenes with multiple objects

For scenes with multiple objects, we can firstly segment all objects out and then apply our method separately. We show an example in Figure 6.

5. More visual comparisons

We present more visual comparisons on the GSO dataset in Figure 2 and the NAVI dataset in Figure 3.

6. Performance under multi-view setting

We adapt our method to a setting with 16 reference views. We adopt a simple strategy of selecting the nearest reference view as the conditioning image. This approach leads to a 19% accuracy increase on the filtered testing dataset that includes only objects with sufficient images.

Table 4. The quantitative comparison results under smaller thresholds on the synthesized dataset GSO [4], and the real dataset NAVI [6].

Methods	NAVI [6] Rotation Accuracy			GSO [4] Rotation Accuracy		
	1°	5°	10°	1°	5°	10°
3DAHV [16]	0.00	6.37	18.28	0.00	4.13	10.87
IDPose [3]	0.37	1.48	4.44	0.22	5.22	14.13
Relpose++ [8]	0.37	9.14	15.47	0.22	2.17	7.83
E2VG(N=128) [14]	0.19	**13.01**	30.38	0.22	11.30	28.26
Ours	**0.56**	10.82	**34.31**	0.22	**17.61**	**42.39**

Moreover, there are several ways to further enhance our method in a multi-view setting, such as stochastic multi-view conditioning [15] and fine-tuning the generative model using LoRA [5].

7. The Accuracy under smaller thresholds

The accuracy under smaller thresholds are shown in Table 4.

8. The comparison under different viewpoint changes

Following Table 2 in the main paper, here we present a more detailed result regarding how the accuracy gap between the proposed method and baselines changes as δ changes. The analysis is performed on the GSO dataset, which comprises images with viewpoints uniformly distributed across the upper hemisphere, thus providing more cases with large pose changes than NAVI.

In Figure 7 (a), we present the Acc30° for pairs where $\delta \geq D$, across various values of D. When $D = 0^\circ$, which includes all pairs, our accuracy is 1.26 times that of E2VG. As D increases to 135°, the ratio increases to 1.65. In Figure 7 (b), we display Acc30° for pairs where $D' - 5^\circ \leq \delta < D' + 5^\circ$, with $D' = \{5^\circ, 15^\circ, ..., 175^\circ\}$. It is evident that for pairs with small viewpoint changes, both our proposed method and E2VG perform well. However, as D' increases, the accuracy of E2VG declines more rapidly compared to our method.

Interestingly, in Figure 7 (a), when $\delta > 165^\circ$ the performance of all three diffusion-based methods improves compared to moderate δ. This might be due to the following two reasons:

- A δ closer to 180° means that Ir and Iq are nearly opposite. Additionally, since almost all images are in the upper hemisphere of the object, this further suggests that the elevation of Ir and Iq is near zero, while the azimuth change approaches 180°. As all three diffusion-based methods use an elevation estimation module [10] to estimate the elevation of Ir, and we

Table 5. The translation accuracy on the two testing datasets. We compute the translation error as the angle between the normalized ground-truth translation $\mathbf{t}_{\text{gt}}$ and the normalized predicted translation $\mathbf{t}_{\text{pr}}$ by $\arccos \mathbf{t}_{\text{gt}}^{\intercal}\mathbf{t}_{\text{pr}}$.

Methods	NAVI [6]		GSO [4]	
	Translation Accuracy		Translation Accuracy	
	15°	30°	15°	30°
SIFT [11]+ZoeDepth [2]+PnP	12.47	25.25	5.65	15.65
SIFT [11]+ZoeDepth [2]+Procrustes	9.60	24.31	3.48	14.57
Map-free-loc RPR [1]	13.81	35.92	3.91	16.52
LoFTR [13]	20.74	30.38	29.57	36.52
Relpose++ [8]	24.84	42.71	20.22	32.61
E2VG(N=128) [14]	50.64	72.41	42.39	60.65
Ours	**62.70**	**82.57**	**59.78**	**75.43**

Table 6. The quantitative comparison results on the rotated NAVI dataset.

Methods	Rotated NAVI			
	Rotation Accuracy		Translation Accuracy	
	15°	30°	15°	30°
SIFT [9]+ZoeDepth [2]+PnP	18.09	23.90	13.68	22.08
SIFT [9]+ZoeDepth [2]+Procrustes	15.47	23.95	8.11	19.31
Map-free-loc RPR [1]	8.69	17.58	7.78	22.23
LoFTR [13]	13.14	21.28	17.38	28.97
IDPose [3]	11.38	25.14	-	-
3DAHV [16]	12.50	30.72	-	-
Relpose++ [8]	12.10	26.46	12.74	29.20
E2VG(N=128) [14]	**29.81**	49.34	35.36	60.27
Ours	27.68	**59.57**	**35.72**	**67.12**

empirically found that this module works slightly better when the input image has a low elevation, a possible explanation for this phenomenon is that when δ is closer to 180°, all Ir have a low elevation, allowing the elevation estimator to give a better estimation compared to pairs under other δ, which finally leads to better Acc30°. To validate this explanation, we conducted an experiment where the ground-truth elevation I_r is provided. We found that the phenomenon was reduced by half (when δ is closer to 180°, the increase in Acc30° was halved), partially supporting this explanation.

- There are only six pairs where $\delta \geq 175°$ and only 21 pairs where $\delta \geq 170°$, making the results more susceptible to random variation. It is possible that Zero123 happened to perform better on these few pairs compared to others.

9. The translation accuracy

In the main paper, we focus on the 3D rotation because the 3D translation can be easily derived from the rotation and the given 2D object bounding box [14, 16], meaning the translation accuracy is highly related to the rotation accuracy.

Here we present the translation accuracy in Table 5. The translation contains a scale ambiguity, so we follow previous work [14] to compute the translation error as the angle between the normalized ground-truth translation $\mathbf{t}_{\text{gt}}$ and the normalized predicted translation $\mathbf{t}_{\text{pr}}$ by $\arccos \mathbf{t}_{\text{gt}}^{\intercal}\mathbf{t}_{\text{pr}}$. The results of ID-Pose and 3DAHV are not reported because ID-Pose and 3DAHV do not estimate the object translation in their works.

10. Performance on symmetric objects

There is one symmetric object in the testset, as visualized in Figure 8. For this object, the proposed method reaches 60% for Acc15° and 65% for Acc30° while all baselines achieve less than 40% at Acc15° and less than 55% at Acc30°. A visual comparison can be found in Figure 2. This case indicates that, despite the object's symmetric shape, the proposed method can effectively capture details like the shadow on this object.

To quantitatively show how the proposed method works on objects with axial symmetry, we select six objects from the GSO [4] dataset as shown in Figure 9, and render multiple images to create a new testing dataset of axially symmetric objects. Using the difference in elevation angle as the error metric, the proposed method achieves an average accuracy of 78% at Acc15° and 93% at Acc30° across the six objects.

References

[1] Eduardo Arnold, Jamie Wynn, Sara Vicente, Guillermo Garcia-Hernando, Aron Monszpart, Victor Prisacariu, Daniyar Turmukhambetov, and Eric Brachmann. Map-free visual relocalization: Metric pose relative to a single image. In *European Conference on Computer Vision*, pages 690–708. Springer, 2022. 5

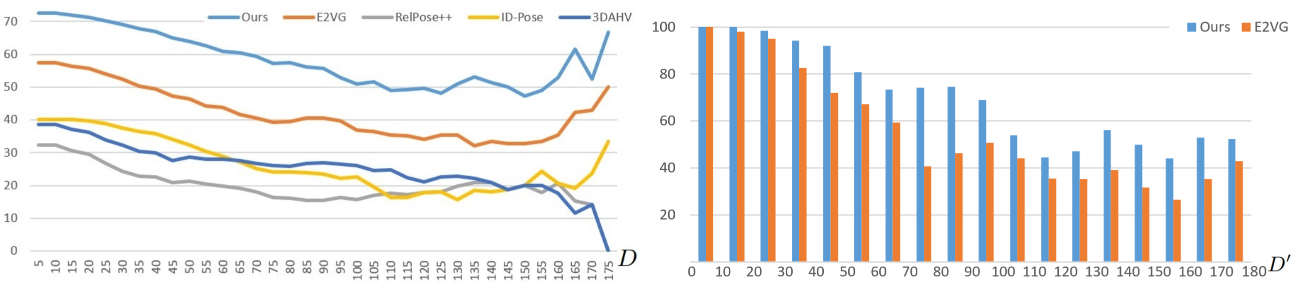
[2] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023. 5

[3] Weihao Cheng, Yan-Pei Cao, and Ying Shan. Id-pose: Sparse-view camera pose estimation by inverting diffusion models. *arXiv preprint arXiv:2306.17140*, 2023. 4, 5

[4] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2553–2560. IEEE, 2022. 3, 4, 5

[5] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. 4

[6] Varun Jampani, Kevis-Kokitsi Maninis, Andreas Engelhardt, Arjun Karpur, Karen Truong, Kyle Sargent, Stefan



(a)

(b)

Figure 7. The comparison results under different viewpoint changes. We report the proportion of angular errors within 30° on GSO dataset. (a) Acc30° for $\delta \geq D$. (b) Acc30° for $D' - 5^\circ \leq \delta < D' + 5^\circ$ with $D' = \{5^\circ, 15^\circ, ..., 175^\circ\}$.

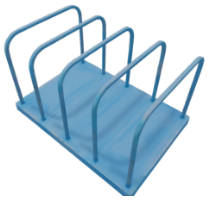


Figure 8. A symmetric object in the testsets.

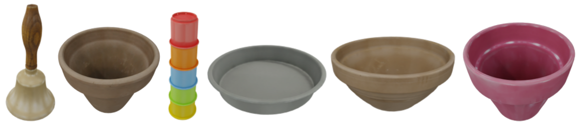


Figure 9. Six axially symmetric objects are selected from GSO dataset to create a new testset.

Popov, Andre Araujo, Ricardo Martin-Brualla, Kaushal Patel, Daniel Vlasic, Vittorio Ferrari, Ameesh Makadia, Ce Liu, Yuanzhen Li, and Howard Zhou. NAVI: Category-agnostic image collections with high-quality 3d shape and pose annotations. In *NeurIPS*, 2023. 3, 4, 5

[7] Senmao Li, Taihang Hu, Fahad Shahbaz Khan, Linxuan Li, Shiqi Yang, Yaxing Wang, Ming-Ming Cheng, and Jian Yang. Faster diffusion: Rethinking the role of unet encoder in diffusion models. *arXiv preprint arXiv:2312.09608*, 2023. 1

[8] Amy Lin, Jason Y Zhang, Deva Ramanan, and Shubham Tulsiani. Relpose++: Recovering 6d poses from sparse-view observations. *arXiv preprint arXiv:2305.04926*, 2023. 4, 5

[9] Ce Liu, Jenny Yuen, and Antonio Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):978–994, 2010. 5

[10] Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Zexiang Xu, Hao Su, et al. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. *arXiv preprint arXiv:2306.16928*, 2023. 4

[11] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110, 2004. 5

[12] Xinyin Ma, Gongfan Fang, and Xinchao Wang. Deepcache: Accelerating diffusion models for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15762–15772, 2024. 1

[13] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8922–8931, 2021. 5

[14] Yujing Sun, Caiyi Sun, Yuan Liu, Yuexin Ma, and Siu Ming Yiu. Extreme Two-View Geometry From Object Poses with Diffusion Models. *arXiv e-prints*, page arXiv:2402.02800, Feb. 2024. 4, 5

[15] Daniel Watson, William Chan, Ricardo Martin-Brualla, Jonathan Ho, Andrea Tagliasacchi, and Mohammad Norouzi. Novel view synthesis with diffusion models. *arXiv preprint arXiv:2210.04628*, 2022. 4

[16] Chen Zhao, Tong Zhang, and Mathieu Salzmann. 3d-aware hypothesis & verification for generalizable relative object pose estimation. *Proceedings of the International Conference on Learning Representations*, 2024. 4, 5