
Supplemental Materials

Wenzhe Ouyang¹, Jinghua Wang², Zenglin Xu^{3,4}, Jiming Chen¹, Qi Ye^{1*}

¹ Zhejiang University, ² Harbin Institute of Technology, Shenzhen,

³ Fudan University, ⁴ Shanghai Academy of AI for Science

1 More Implementation Details

In this section, we provide more implementation details of the proposed method and experiments.

1.1 Network Details

For a fair comparison, we employ identical point cloud and RGB feature encoders with GenPose++ [42], utilizing the parameter-efficient "ViT-S/14" variant of DINOv2 to maintain consistency in model architecture and parameter scale. Our pipeline processes instance-cropped RGB images from Omni6DPose [42], where object regions are extracted using ground-truth masks and resized to 224×224 resolution. These standardized image patches are subsequently fed into DINOv2 to generate a 16×16 feature grid, where each 384-dimensional vector corresponds to spatial features from a 14×14 patch in the original RGB input.

As illustrated in Fig.1, our conditional input integrates three components: concatenated visual features, sampled pose parameters $[R_t, T_t]$ and timestep t . The concatenated features undergo dimensionality expansion to 1024 through feature encoding, while parallel MLP branches separately process the pose parameters and timestep into 256-dimensional embeddings. The final velocity vector v_t is synthesized through a fusion MLP, decomposed into rotational ($v_t[:3] \in SO(3)$) and $v_t[3:] \in \mathbb{R}^3$) components, corresponding to the Lie algebra representation of SE(3) transformations.

1.2 Training and Inference Details

During the training phase, our framework undergoes optimization through 28 training epochs under identical experimental configurations as GenPose++ [42], employing the Adam optimizer with a batch size of 128 to ensure training stability. The learning rate scheduling follows a two-phase decay strategy as GenPose++ [42]: initialized at 0.001 for rapid gradient descent, then progressively reduced to 0.0001 through cosine annealing to facilitate fine-grained parameter adjustments near convergence regions. Notably, we do not utilize the any data augmentation strategy. In Omni6DPose [42], objects are classified into four types based on their symmetry properties: arbitrary symmetry, half-symmetry, quarter symmetry, and asymmetry, with three defined axes of symmetry. For simplicity, we only focus on objects with semi-symmetric properties in this paper, treating other symmetric cases as asymmetric.

In the inference phase, the threshold δ for discarding candidates with likelihoods below it is set to 40%. To eliminate the influence of scale estimation and ensure a fair comparison, we used the same scale estimation results as GenPose++ [42]. The ordinary differential equation (ODE) solver employs a fourth-order Runge-Kutta method with 20 discretization steps to balance computational efficiency and numerical precision. For uncertainty quantification in probability estimation, we configure the Hutchinson trace estimation procedure with $N = 10$ Monte Carlo samples, achieving an optimal trade-off between computational overhead and variance reduction in likelihood estimation.

*Corresponding author: Qi Ye (qi.ye@zju.edu.cn). This work was supported in part by NSFC under Grants (No.62233013, 62088101, 62293511, 62172285), Key Research and Development Program of Zhejiang Province (No.2025C01064) and Shenzhen Science and Technology Program (Project No.GXWD 20231130125451001).

Table 1: **Quantitative comparison of category-level object pose estimation on Omni6DPose IKEA test-set.** The results are averaged over all 149 categories.

Method	Input Modality	IoU			AUC			
		IoU ₂₅	IoU ₅₀	IoU ₇₅	5°2cm	5°5cm	10°2cm	10°5cm
Deterministic:								
- HS-Pose [45]	Point Clouds	40.2	18.9	4.9	3.6	7.9	8.1	12.8
- AG-Pose [20]	RGB-D	38.9	17.7	4.8	2.4	6.5	6.7	10.6
- SecondPose [10]	RGB-D	42.7	19.9	5.7	3.5	8.9	11.5	15.9
Probabilistic:								
- GenPose++ [42]	RGB-D	50.7	34.7	10.2	10.4	12.9	21.7	28.3
- Ours	RGB-D	54.4	38.4	11.9	12.4	16.7	23.5	32.0

Table 2: **Quantitative comparison of category-level object pose estimation on Omni6DPose Matterport3D test-set.** The results are averaged over all 149 categories.

Method	Input Modality	IoU			AUC			
		IoU ₂₅	IoU ₅₀	IoU ₇₅	5°2cm	5°5cm	10°2cm	10°5cm
Deterministic:								
- HS-Pose [45]	Point Clouds	39.3	18.0	5.0	3.7	7.9	8.3	13.1
- AG-Pose [20]	RGB-D	39.0	17.1	4.4	2.6	6.7	7.1	10.9
- SecondPose [10]	RGB-D	40.4	18.6	5.5	3.7	9.0	11.8	16.1
Probabilistic:								
- GenPose++ [42]	RGB-D	48.1	33.6	9.9	10.2	12.8	20.4	28.2
- Ours	RGB-D	51.7	35.7	11.1	12.3	16.7	22.7	31.6

This systematic configuration ensures both reproducibility and fair benchmarking against existing state-of-the-art approaches.

1.3 Evaluation Metrics

Following previous works, we report the 6D pose estimation accuracy, derived from the Volume Under Surface (VUS), across ranges of n° and m cm for 6D pose estimation, which denotes the percentage of prediction with rotation error less than n° and translation error less than m cm. Specifically, we use $5^\circ 2cm$, $5^\circ 5cm$, $10^\circ 2cm$, and $10^\circ 5cm$ as our evaluation metrics. We also report the Intersection over Union (IoU) for 3D bounding boxes with thresholds of $x\%$. As for object pose tracking evaluation, we utilize $5^\circ 5cm$, mean Intersection over Union (mIoU), average rotation error in degrees $R_{err}(\circ)$, and average translation error in centimeters $T_{err}(cm)$ as metrics.

2 More Quantitative and Visualizations Results

As shown in Table 1, we further report the quantitative comparison between our method and existing methods on Omni6DPose IKEA [1]. The proposed method still outperforms all other competing methods. Table 2 further shows the quantitative comparison of the proposed method on Omni6DPose Matterport3D [5].

Figure 1 presents detailed comparative visualization results of our model against GenPose++ [42] and the ground truths.

3 More Ablation Studies

In this section, we provide more ablation studies of the proposed method. We further focus on the following two aspects in our ablation studies: 1) the analysis of the methods for solving ODE and the impact of different step sizes; 2) the analysis of the Jacobian-Vector Product(JVP) Computation.

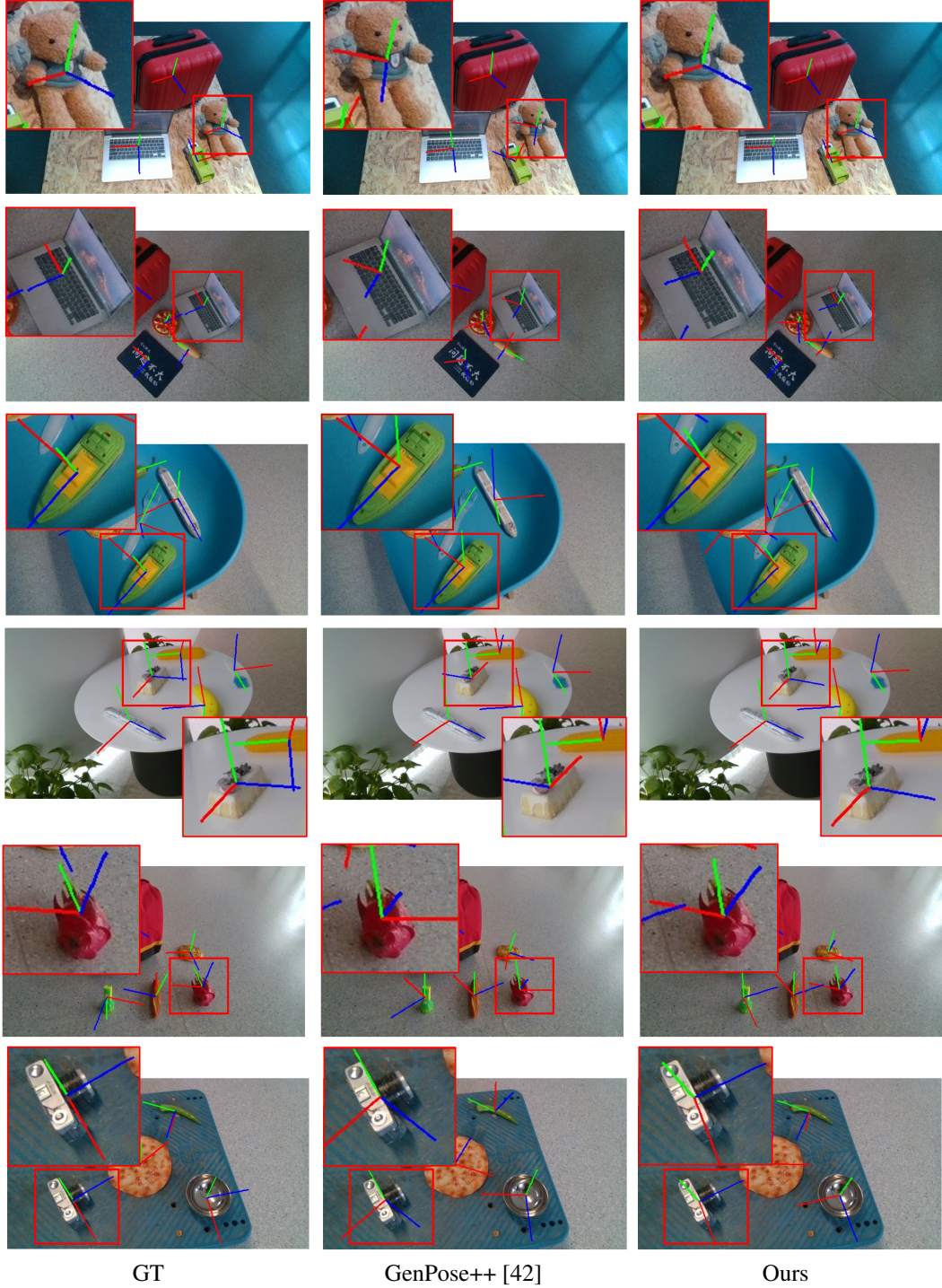


Figure 1: **Visualization comparison on Omni6DPose [42]**. The left column represents the ground truth pose, the middle column represents the results of GenPose++ [42], the right column represents the results of our approach. As shown in the zoomed area of the figure above, our approach has achieved better performance than GenPose++ [42]. In the last two rows, we also present some failure cases of our model on symmetrical objects (such as the stainless steel bowl in the last row).

Table 3: Ablation studies on the methods for solving ODE and the impact of different step sizes.

Ablation	IoU ₂₅ ↑	5°2cm↑	5°5cm↑	Latency↓
Euler’s method(20 Steps)	47.1	12.6	16.1	20ms
Runge-Kutta’s method(10 Steps)	46.7	12.5	15.8	17ms
Runge-Kutta’s method(20 Steps)	48.0	12.8	16.2	30ms
Runge-Kutta’s method(50 Steps)	47.4	12.6	16.0	66ms

Table 4: Ablation studies on the Jacobian-Vector Product(JVP) Computation.

Ablation	IoU ₂₅ ↑	5°2cm↑	5°5cm↑	Latency↓
Energy-Net in GenPose++ [42]	43.9	10.4	13.2	46ms
Ours($N = 1$)	45.6	11.7	14.9	14ms
Ours($N = 10$)	48.0	12.8	16.2	30ms
Ours($N = 20$)	47.9	12.8	16.2	43ms

The analysis of the methods for solving ODE and the impact of different step sizes. In this paper, we adopt a fourth-order Runge-Kutta method with 20 discretization steps as the ODE solver. To validate the reasonableness of this configuration, we compare different ODE solvers and step sizes. As demonstrated in Table 3, the Runge-Kutta method (20 steps) achieves a favorable balance between performance and computational efficiency.

The analysis of the Jacobian-Vector Product(JVP) Computation. In this paper, we employ JVP computation to estimate likelihood for each sample, which can be seen as a concern in some methods. By default, we compute the product between the Jacobian matrix and the tangent vector N times to obtain the expectation of JVP (Eq. 11). Notably, we exclusively calculate the latency of the ODE inference process and the flow matching network process, without considering the latency of data loading and feature extraction. As shown in Table 4, when $N=1$, a noticeable performance degradation occurs due to the inaccurate likelihood estimation at this point. When $N=10$, our model achieves a favorable balance between computational efficiency and performance, while still maintaining a significant latency advantage compared to GenPose++ [42]. When N exceeds 10, the performance does not improve, whereas computational resource consumption and latency increase significantly.