$\mathcal{T}(\mathcal{R}, \mathcal{O})$ Grasp: Efficient Graph Diffusion of Robot-Object Spatial Transformation for Cross-Embodiment Dexterous Grasping

Anonymous Authors

APPENDIX

A. Real-World Experiment

1) Data Collection and Model Training: To perform real-world dexterous grasping, we collect LEAP Hand Dataset and XHand Dataset to train $\mathcal{T}(\mathcal{R}, \mathcal{O})$ Grasp models independently. We select 50 objects from ContactDB [1] and 28 objects from YCB dataset [2], applying DFC-based [3] grasp optimization to generate grasping demonstrations. After filtering, we obtain 7,800 grasp demonstrations for each hand as the training dataset. Following the training setup in Sec. IV-A, we train $\mathcal{T}(\mathcal{R}, \mathcal{O})$ Grasp on the collected dataset and evaluate its performance in real-world scenarios.

2) Real-world Deployment: First, we use AR Code [4] to scan 10 novel objects for each hand. After camera calibration, we employ FoundationPose [5] to estimate the object pose from monocular RGB-D input captured by an Intel RealSense D435 camera. The point cloud input for each object is then obtained by transforming the sampled point cloud from the scanned 3D model into the world frame. To avoid collision in the tabletop grasp setting, we randomly sample an initial hand pose from top-down to right-side orientations, while taking the sampled initial pose as guidance during $\mathcal{T}(\mathcal{R}, \mathcal{O})$ grasp synthesis. Then, we use MPLib [6] for xArm motion planning to reach the desired end-effector pose. For closed-loop grasping in dynamic environments, we place the object on a conveyor belt and employ FoundationPose tracking to continuously update its pose, repeating the above process in real time.

3) Experiment Results: As shown in Tab. I and Tab. II, $\mathcal{T}(\mathcal{R}, \mathcal{O})$ Grasp achieves an average success rate of **91%** and **90%** on XHand and LEAP Hand, respectively. Visualization in Fig. 1 demonstrates that our method performs robust and generalizable grasp synthesis on novel objects. Furthermore, Fig. 2 indicates that the high inference speed of $\mathcal{T}(\mathcal{R}, \mathcal{O})$ Grasp enables closed-loop grasp synthesis, allowing it to successfully capture moving objects on a conveyor belt. Complete videos of real-world experiments are available on the project website https://tro-grasp.github.io/.

Apple	**Bottle**	**Cola**	**Cylinder**	**Box**
9/10	10/10	9/10	8/10	9/10
Orange	**Sauce**	**Sponge**	**Toy**	**Spray Bottle**
10/10	8/10	9/10	10/10	9/10

TABLE I: Real-world experiment results on XHand.

Chip Box	**Bottle**	**Cola**	**Cylinder**	**Box**
9/10	10/10	10/10	7/10	9/10
Orange	**Sauce**	**Sponge**	**Spray Bottle**	**Toy**
8/10	8/10	9/10	10/10	10/10

TABLE II: Real-world experiment results on LEAP Hand.

B. Network Architecture

1) VQ-VAE Encoder: In $\mathcal{T}(\mathcal{R}, \mathcal{O})$ Graph, we leverage the pretrained VQ-VAE encoder from [7] to partition object patches and extract corresponding geometry tokens. The object point cloud is first normalized within a unit sphere, and then encoded by the pretrained encoder into $P = 25$ local geometry features $\{f_i^O\}_{i=1}^P$ along with corresponding patch center coordinates $\{c_i^O\}_{i=1}^P$.

2) BPS Encoder: Since the number of point cloud varies for each link, we employ Basis Point Set (BPS) [8] algorithm to encode each link point cloud into a fixed-length geometric feature. Point clouds of the dexterous hand with L links are defined as $\{P_i^R\}_{i=1}^L$ in their respective local frames, where each link point cloud is $P_i^R = \{p_{i1}, \ldots, p_{in_i}\} \in \mathbb{R}^{n_i \times 3}$. First, we normalize all points into a unit sphere:

$$p_{ij} = \frac{p_{ij} - \frac{1}{n_i}\sum_j p_{ij}}{\max_j ||p_{ij} - \frac{1}{n_i}\sum_j p_{ij}||}, \ \forall \ i, j. \tag{1}$$

Next, we randomly sample $B = 124$ points within the unit sphere as the basis point set for all link point clouds:

$$\mathbf{B} = [b_1, ..., b_B]^T, \ ||b_j|| \leq 1, \ \forall j. \tag{2}$$

Then, the BPS feature can be formulated as the minimum distance between the normalized link point cloud and basis point set to represent the link geometry, which is then encoded to link nodes as illustrated in Sec. III-A.

$$\text{BPS}(P_i^R) = [\min_j ||p_{ij} - b_1||, \ldots, \min_j ||p_{ij} - b_B||]. \tag{3}$$

3) Graph Denoising Layer: To predict noise on link node from the noisy $\mathcal{T}(\mathcal{R}, \mathcal{O})$ Graph, we employ a graph denoiser composed of $N = 6$ layers, each consisting of one OR-attention and one RR-attention block. Fig. 3 illustrates the structure details of both attention blocks, where attention mechanism aggregates information from graph nodes and edges to update their representations.

C. Cross-embodiment Zero-shot Generalization

Since $\mathcal{T}(\mathcal{R}, \mathcal{O})$ Grasp has only been trained on a limited set of dexterous hands, directly performing zero-shot experiments on a completely unseen hand is infeasible. Instead,

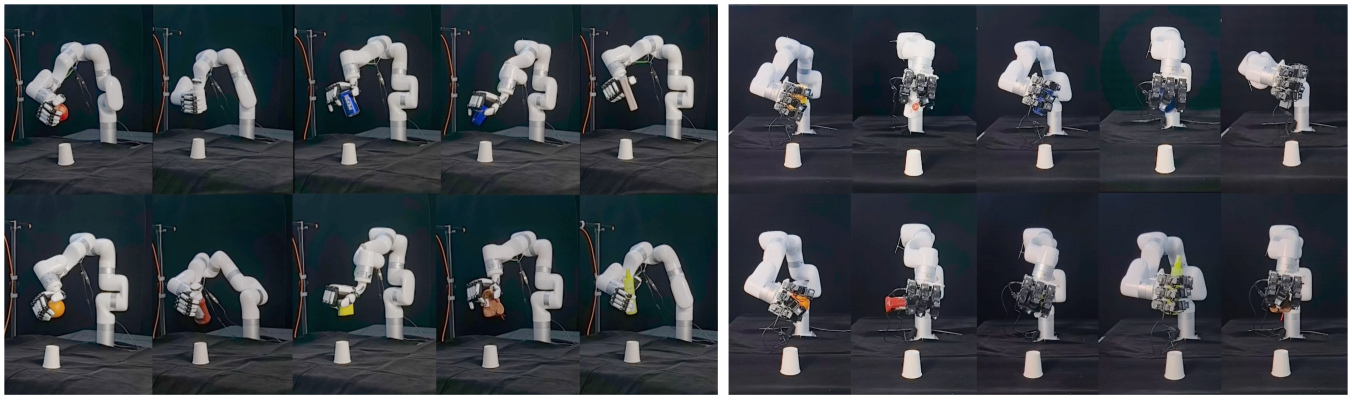

Fig. 1: Real-world grasp synthesis on XHand (left) and LEAP Hand (right).

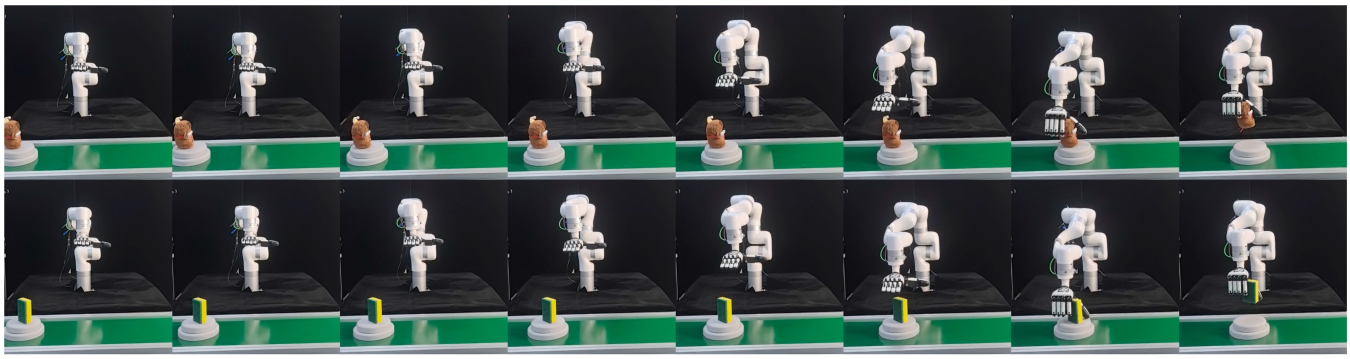

Fig. 2: Real-world closed-loop grasp synthesis in dynamic environments.

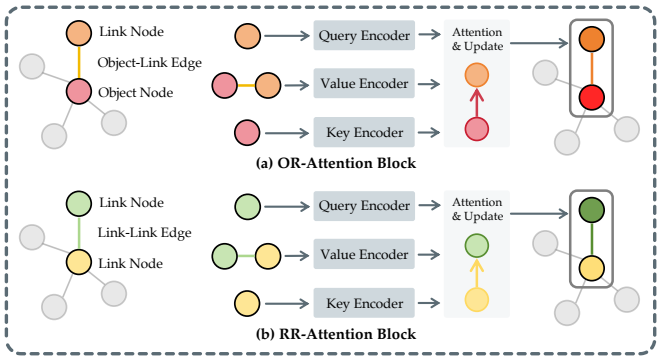

Fig. 3: OR and RR attention block.

creating derived embodiments from existing hands provides a controllable way to approximate cross-embodiment zero-shot generalization. Hence, we construct new hand embodiments by modifying the link length and joint limits of Allegro, Barrett, and ShadowHand. To assess the embodiment similarity, we define link alignment S_L and joint overlap S_J as:

$$S_L = 1 - \frac{1}{L}\sum_{i=1}^{L}\frac{|l_i' - l_i|}{l_i}, S_J = \frac{1}{L}\sum_{i=1}^{L}\frac{|j_i' \cap j_i|}{|j_i' \cup j_i|}. \quad (4)$$

where l_i and l_i' denote the original and modified link lengths, j_i and j_i' are the corresponding joint ranges. We train $\mathcal{T}(\mathcal{R}, \mathcal{O})$ Grasp on the original embodiment of Allegro, Barret and Shadowhand, and evaluate it on their derived embodiments. As illustrated in Fig. 4, $\mathcal{T}(\mathcal{R}, \mathcal{O})$ Grasp achieves over 70% success rate on test hands with similarity ≥ 0.5, highlighting the strong zero-shot capability of $\mathcal{T}(\mathcal{R}, \mathcal{O})$ across dexterous hand embodiments with comparable geometries. Notably, the current zero-shot performance is constrained by the limited training embodiments. This suggests that, when trained on a large-scale embodiment dataset, $\mathcal{T}(\mathcal{R}, \mathcal{O})$ has the potential to scale up to a foundation model for dexterous grasping with strong zero-shot generalization.

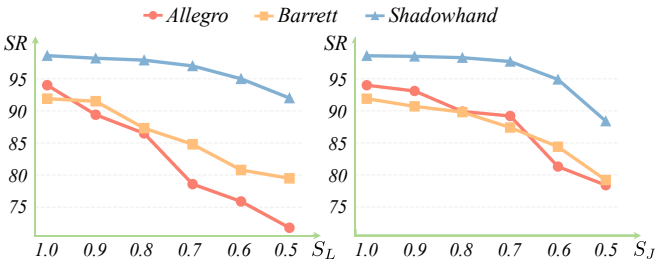

Fig. 4: Cross-embodiment zero-shot performance.

D. More Implementation Details

In this section, we provide more comprehensive details on network architecture, training and inference.

1) $\mathcal{T}(\mathcal{R}, \mathcal{O})$ Graph construction: For object nodes, the pretrained VQ-VAE [9] encodes object point cloud into $P = 25$ tokens with 64 dimensions, resulting in object nodes $N^O \in \mathbb{R}^{25\times(3+1+64)}$. For link nodes, BPS features along with link centers and scales are embeded to 128 dimensions. To allow parallel computing across embodiments with different number of links, link nodes are zero-padded to $N^R \in \mathbb{R}^{25\times(6+128)}$. Consequently, the object–link edges

Fig. 5: Visualization of unconditioned grasp synthesis.

Fig. 6: Visualization of conditioned grasp synthesis: red arrow denotes the direction of grasp guidance.

and link–link edges take the forms $E^{OR} \in \mathbb{R}^{6\times25\times25}$ and $E^{RR} \in \mathbb{R}^{6\times25\times24/2}$, respectively.

2) Training: During training, we adopt a linear scheduler for the noise variance ranging from $\beta_{\text{min}} = 1 \times 10^{-4}$ to $\beta_{\text{max}} = 0.02$ with $T = 1000$ diffusion steps in total. $\mathcal{T}(\mathcal{R}, \mathcal{O})$ Grasp model is trained for 300 epochs with Adam optimizer. The initial learning rate is set to 1×10^{-4} and decays by a factor of 0.8 every 20 epochs. Position and rotation noise loss weights are set to $\gamma_p = \gamma_r = 1.0$.

3) Inference: For both unconditioned and conditioned grasp synthesis, we follow DDIM [10] diffusion strategy to sample $M = 20$ steps for inference. To encourage grasp diversity, we set $\lambda = 0.2$ to inject random noise during inference. In conditioned grasp synthesis, the strength of orientation guidance is formulated as:

$$s(t) = 0.5 \sin(\frac{i\pi}{2M}),\ i = 1, \ldots, M. \tag{5}$$

where M is the total number of inference steps. The schedule $s(t)$ preserves diffusion diversity in the early steps, while encouraging the generated grasp to follow the orientation guidance in the later steps.

E. More Visualization

We provide more visualization on unconditioned and conditioned grasp synthesis of $\mathcal{T}(\mathcal{R}, \mathcal{O})$ Grasp in Fig. 5 and Fig. 6, respectively. Our model consistently produces accurate dexterous grasps on novel objects, while diverse orientation guidance enhances grasp diversity.

References

[1] S. Brahmbhatt, C. Ham, C. C. Kemp, and J. Hays, "Contactdb: Analyzing and predicting grasp contact via thermal imaging," in *CVPR*, 2019, pp. 8709–8719.

[2] B. Calli, A. Singh, J. Bruce, A. Walsman, K. Konolige, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Yale-cmu-berkeley dataset for robotic manipulation research," *The International Journal of Robotics Research*, vol. 36, no. 3, pp. 261–268, 2017.

[3] T. Liu, Z. Liu, Z. Jiao, Y. Zhu, and S.-C. Zhu, "Synthesizing diverse and physically stable grasps with arbitrary hand structures using differentiable force closure estimator," *IEEE Robotics and Automation Letters*, vol. 7, no. 1, pp. 470–477, 2021.

[4] AR Code, "Ar code," 2022, accessed: 2024-09-28. [Online]. Available: https://ar-code.com/

[5] B. Wen, W. Yang, J. Kautz, and S. Birchfield, "Foundationpose: Unified 6d pose estimation and tracking of novel objects," in *CVPR*, 2024, pp. 17 868–17 879.

[6] H. S. Lab, "Mplib: Motion planning library," 2023, accessed: 2024-09-28. [Online]. Available: https://github.com/haosulab/MPlib

[7] Z. Wang, J. Chen, and Y. Furukawa, "Puzzlefusion++: Auto-agglomerative 3d fracture assembly by denoise and verify," in *ICLR*, 2025.

[8] S. Prokudin, C. Lassner, and J. Romero, "Efficient learning on point clouds with basis point sets," in *ICCV*, 2019, pp. 4332–4341.

[9] A. Van Den Oord, O. Vinyals *et al.*, "Neural discrete representation learning," *NeurIPS*, vol. 30, 2017.

[10] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," *arXiv preprint arXiv:2010.02502*, 2020.