GeoMatch++: Morphology Conditioned Geometry Matching for Multi-Embodiment Grasping

Yunze Wei[1], **Maria Attarian**[1,2], **Igor Gilitschenski**[1]

[1]University of Toronto, [2]Google DeepMind

Abstract: Despite recent progress on multi-finger dexterous grasping, current methods focus on single grippers and unseen objects, and even the ones that explore cross-embodiment, often fail to generalize well to unseen end-effectors. This work addresses the problem of dexterous grasping generalization to unseen end-effectors via a unified policy that learns correlation between gripper morphology and object geometry. Robot morphology contains rich information representing how joints and links connect and move with respect to each other and thus, we leverage it through attention to learn better end-effector geometry features. Our experiments show an average of **9.64%** increase in grasp success rate across 3 out-of-domain end-effectors compared to previous methods.

Keywords: Robot Morphology, Dexterous Grasping, Multi-Embodiment

1 Introduction

As we aspire to solve more dexterous tasks in robotics, multi-finger grasping becomes of increasing importance. However, the varying degrees of freedom (DoF) of end-effectors and high multimodality of grasping modes depending on both end-effectors and objects, still pose open challenges. Previous works in grasping focus on parallel grippers [1, 2, 3], a single multi-finger gripper [4, 5, 6, 7], or a shared policy for multiple dexterous grippers [8, 9, 10, 11]. However, even methods that explore cross-embodiment mostly focus on generalization to unseen objects, and still show limited zero-shot generalization to unseen grippers.

In this work, we propose GeoMatch++, a multi-embodiment grasping method which improves out-of-domain generalization on unseen grippers by leveraging robot morphology. Intuitively, robot morphology is essential to grasping – various end-effectors may have a different number of fingers, but fingertips and palm tend to be the most frequent contact regions. Thus, we hypothesize that learning good morphology embeddings can lead to a transferable grasping policy between different robots. Our main contribution is learning geometry correlation features between objects and end-effector morphology, which improve out-of-domain grasp success by 9.64% compared to previous methods, and our method showcases a minimal decrease in performance compared to in-domain evaluation.

2 Related Work

Dexterous Grasping: Works focused on grasping for multi-finger grippers either train an end-to-end model to predict gripper pose directly [4, 6, 7] or learn a contact map distribution before computing the final grasp [8, 9, 10, 11]. Many of these methods are either constrained to one end-effector while others can generalize to unseen grippers, yet rarely incorporate gripper morphology explicitly, which better represents how complex multiple DoF grippers move during grasping. TAX-Pose [12] is a recent method that learns a task-specific pose relationship between target objects to address manipulation tasks that involve multiple objects. The authors are inspired by Deep Closest Point (DCP) [13], which proposes using transformers [14] to learn a matching between point clouds, and show that attention is also beneficial to the grasping problem. Instead of capturing the attention between point clouds, we propose using self-attention and cross-attention between object point cloud and end-effector morphology to learn a transferable grasping policy. Our work extends GeoMatch [8], which uses Graph Convolutional Networks (GCN) [15] to learn object and robot geometries then performs autoregressive matching to predict object-robot contact points, via incorporating such morphology self and cross-attention.

Correspondence emails: lulu.wei@mail.utoronto.ca, jmattarian@google.com

8th Conference on Robot Learning (CoRL 2024), Munich, Germany.

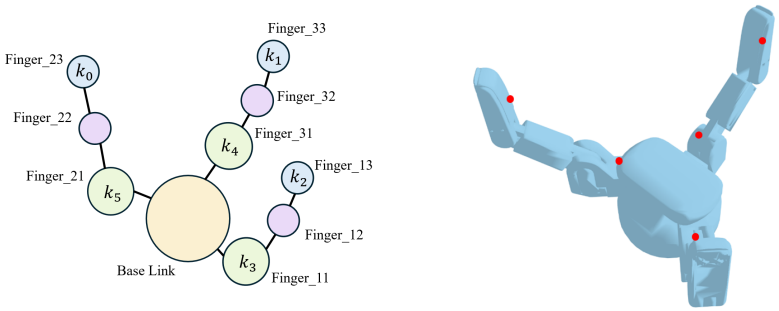


Figure 1: Sample morphology graph for Barrett hand with labelled keypoints.

Robot Morphology: Robot morphology has been explored in other robotics control tasks in policy learning and imitation learning to generalize zero-shot to new tasks and agents [16, 17, 18, 19, 20]. A notable example is NerveNet [17] which explicitly models the structure of a modular agent as a graph, and propagates messages between nodes of the agent to train a reinforcement learning (RL) policy. Prior work has also explored robot structure as an inductive bias for transformers: MetaMorph [16] conditions a transformer on morphology and learns a universal controller while Body Transformer (BoT) [20] considers agent sensors and actuators as graph nodes and modifies attention masking to leverage morphology of the agent's structure. We show that morphology similarly leads to an improvement in generalization for cross-embodiment dexterous grasping.

3 Method

Our model (Fig. 2) learns a multi-embodiment policy that generates diverse grasps for dexterous grippers for both unseen objects and end-effectors. Operating under the same problem formulation as [8], we match $N=6$ pre-defined keypoints on the end-effector $k_0,...,k_{N-1}$ to predicted contact points on the object $c_0,...,c_{N-1}$. Our model encodes graph features for the object point cloud $\mathcal{G}_O$, gripper point cloud $\mathcal{G}_G$, and the graph representing the morphology of the gripper $\mathcal{G}_M$ using GCNs. Transformer modules perform self-attention and cross-attention to capture global correspondence between the object and end-effector. Finally, the model autoregressively predicts contact points using the latent embeddings.

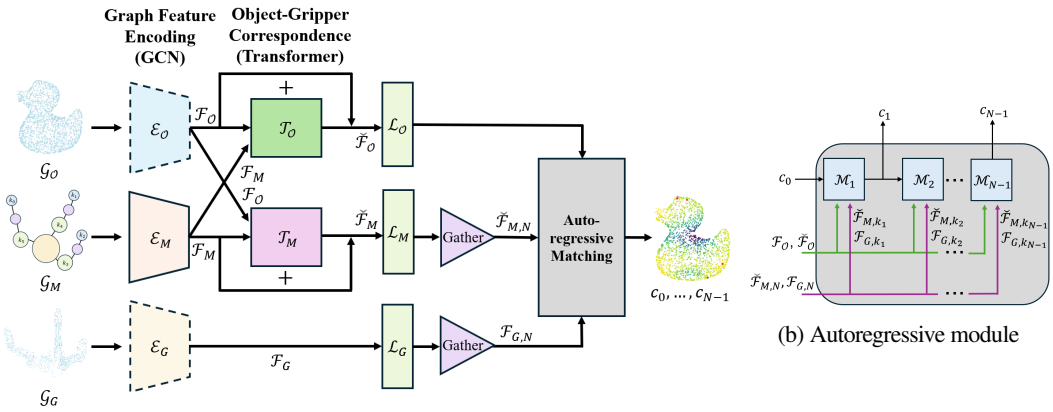
3.1 Dataset

We use a subset of the MultiDex dataset synthesized by [9] using force closure optimization [21]. The dataset contains 5 high-DoF multi-finger grippers, EZGripper, Barrett, Robotiq-3F, Allegro, and ShadowHand, and 58 household objects from the ContactDB [22] and YCB [23] datasets. We train on 50,802 grasps, represented by poses consisting of translation, rotation, and joint angles of the gripper.

3.2 Graph Representation

Object and End-effector Point Clouds: Object and end-effector point clouds are represented as graphs $\mathcal{G}_O=(\mathcal{V}_O,\mathcal{E}_O)$, and $\mathcal{G}_G=(\mathcal{V}_G,\mathcal{E}_G)$. Each point is represented as its 3D coordinates $\mathbf{p_i}=(x_i,y_i,z_i)\in\mathcal{R}^3$. The graph is constructed by sampling $S_O=2048$ points for the object mesh and $S_G=1000$ points from the end-effector mesh. Prior to sampling, the end-effector is set to a canonical rest pose that has zero root translation, zero root rotation, and all joints set to the middle of their joint limits.

End-effector Morphology Representation: The end-effector's kinematic chain, which contains information about link-joint connections and parameters, is obtained from the Universal Robot Description Format (URDF) and constructed as a graph $\mathcal{G}_M=(\mathcal{V}_M,\mathcal{E}_M)$. In our setup, nodes $\mathcal{V}_M$ are links and edges $\mathcal{E}_M$ are joints (Fig. 1). The graph features consist of offset, link centre of mass, and link size. Offset represents the translation between the coordinate frames of two connected links. Link centre of mass is estimated via computing the least volume rectangular bounding boxes around the link mesh and finding its mean coordinate on each axis. Finally, link size is the length, width, and height of the bounding box. The coordinate frame of the centres of mass and the scale of the link sizes are all geometrically consistent with object and end-effector point clouds. Only the offset is encoded relative to two connected nodes. Due to varied DoFs of end-effectors, $\mathcal{G}_M$ is zero-padded to $S_M=32$ to enable batch processing. More details are given in Appendix C.



(a) Model architecture

Figure 2: Architecture for GeoMatch++. GCNs learn latent features for object and gripper point clouds, and end-effector morphology. Features are passed into transformer modules to learn the object-gripper correspondence. Autoregressive matching predicts final contact points using MLP layers.

3.3 Architecture

Graph Feature Encoding: The model uses three separate GCNs to generate latent embeddings of dimension $n=512$ for $\mathcal{G}_O$, $\mathcal{G}_G$, and $\mathcal{G}_M$. We use $\mathcal{F}_O$, $\mathcal{F}_G$, and $\mathcal{F}_M$ to represent the latent embeddings. We use pretrained weights from GeoMatch [8] for $\mathcal{F}_O$ and $\mathcal{F}_G$ and freeze them during training, as empirically this shows the best performance. $\mathcal{G}_M$ is novel to our model and trained from scratch. $\mathcal{G}_M$ is zero-padded to account for different DoFs in the end-effectors which does not pose an issue given GCN's property of only aggregating features of a node's direct neighbourhood.

Object-Gripper Correspondence: We use two transformer modules with self-attention and cross-attention to learn correspondence between the latent embeddings for object features $\mathcal{F}_O$ and morphology $\mathcal{F}_M$. Following Wang and Solomon [13], we consider the output of the transformer as a residual term and add it to the GCN encoding:

$$\hat{\mathcal{F}}_O = \mathcal{F}_O + \mathcal{T}_O(\mathcal{F}_O, \mathcal{F}_M) \ \in \mathcal{R}^{n \times S_O} \quad \hat{\mathcal{F}}_M = \mathcal{F}_M + \mathcal{T}_\mathcal{M}(\mathcal{F}_M, \mathcal{F}_O) \ \in \mathcal{R}^{n \times S_M} \tag{1}$$

This operation modifies features $\mathcal{F}_O$ and $\mathcal{F}_M$ such that they are aware of the correlation between object and morphology. Then, linear layers downsample the embeddings for further processing.

Autoregressive Matching: We modify the autoregressive module from [8] to incorporate morphology encodings. We gather $\mathcal{F}_G$ and $\hat{\mathcal{F}}_M$ to obtain only the embedding corresponding to the N keypoints. Each layer $\mathcal{M}_i$ in autoregressive matching is an MLP that predicts contact point c_i from the concatenation of the full object embedding $\hat{\mathcal{F}}_O$, gathered embeddings $\mathcal{F}_{G,N}$ and $\hat{\mathcal{F}}_{M,N}$ repeated $S_O = 2048$ times, and the contact points $c_0,...,c_{i-1}$ from the previous layers. c_0 is predicted from the unnormalized likelihood contact maps, further explained in Section 3.3.1. Although only the i-th feature of $\hat{\mathcal{F}}_{M,N}$ is used in layer M_i, $\hat{\mathcal{F}}_O$ contains information about the entire end-effector morphology through cross-attention.

3.3.1 Losses

We use the same loss functions as Attarian et al. [8], consisting of the Geometric Embedding Loss and Predicted Contact Loss, with modifications described below. For more details, we refer the reader to the paper.

Geometric Embedding Loss: We calculate the BCE loss between the predicted unnormalized likelihood contact maps for each pair of object vertex v_o and keypoint k_i, and the ground truth contact maps $C_O(v_o, k_i)$. Instead of learning the contact maps using GCN encodings as done in GeoMatch [8], we use the dot product between the object-gripper correspondence transformer output of the object point cloud and the GCN embeddings of the gripper point cloud.

Predicted Contact Loss: We use the same predicted contact loss as [8] to train autoregressive matching contact point predictions.

Method	Success (%) ↑				Diversity (rad) ↑		
	ezgripper	barrett	shadowhand	**Mean**	ezgripper	barrett	shadowhand
GeoMatch [8]	55.0	60.0	67.5	60.83	0.185	0.259	**0.235**
GenDexGrasp [9]	38.59	70.31	**77.19**	62.03	**0.248**	0.267	0.207
GeoMatch++ (ours)	**67.5**	**77.5**	70.0	**71.67**	0.208	**0.378**	0.184

Table 1: Out-of-domain success rate and diversity comparisons with GeoMatch and GenDexGrasp

4 Experiments and Discussion

We use the same evaluation setup as [8, 9] which leverages IsaacGym to measure grasp success rate and diversity. Grasp success rate is calculated over four grasps per object-gripper pairs, and diversity is measured as the standard deviation of the joint angles of successful grasps.

4.1 Out-of-domain evaluation

The model is evaluated on out-of-domain grippers by training on 4 out of 5 grippers, and testing using the unseen gripper with 10 unseen objects. We choose to compare results with two recent methods that focus on multi-gripper dexterous grasping, GeoMatch [8] and GenDexGrasp [9].

Our model shows significant improvement in out-of-domain generalization, having a mean success rate of **71.67%** and a mean grasp diversity of 0.257. GeoMatch++ outperforms the mean success rate of GeoMatch [8] by **10.84%** and GenDexGrasp [9] by **9.64%** (Table 1). Furthermore, our mean out-of-domain performance is only **3.33%** lower than in-domain (75.0%, Appendix A), demonstrating the method's strength in generalizing to new grippers. Sample grasps are rendered in Figure 3.

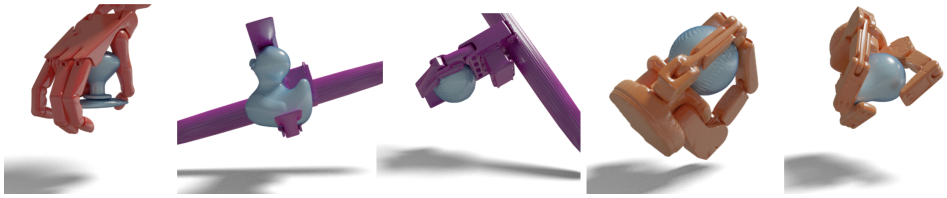


Figure 3: Qualitative grasp results on unseen grippers.

4.2 Ablations

Q1: What is the importance of starting training from good point cloud embeddings? We train ablations where weights of $\mathcal{G}_O$ and $\mathcal{G}_G$ are trained from scratch, pretrained and fintuned, or pretrained and frozen. Empirically, freezing the pretrained weights achieves the best success rate. In particular, we note that training from scratch suffers a large drop in success rate ($24.97\% \downarrow$) (Appendix B.1).

Q2: Does including robot morphology improve out-of-domain generalization? To examine the role of end-effector morphology in generalization, we remove morphology completely and add transformer modules between the object and robot point clouds instead. We find that mean success rate of our final method (including morphology) is 22.51% higher than without morphology (Appendix B.2).

Q3: What is the contribution of different morphology features? The relative importance of features of the robot morphology graph is examined through using different combinations of morphological features in $\mathcal{G}_M$. We run ablations for joints only features (relative offset, joint axis, joint limits) and links only features (absolute origin coordinates, centre of mass, size of bounding box). Our final selection of features, with a combination of relative offset and link coordinate information, achieves the best results (Appendix B.3).

5 Conclusion

In this paper we propose a novel method, GeoMatch++, that leverages robot morphology to improve out-of-domain generalization to unseen grippers. We demonstrate that learning robot link and joint features and the object-morphology correlation are important for achieving high grasp success rates out-of-domain, outperforming baseline by 9.64%. We hope this work is a step forward towards zero-shot generalization to unseen grippers in real robot settings.

References

[1] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox. Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13438–13444. IEEE, 2021.

[2] E. Chisari, N. Heppert, T. Welschehold, W. Burgard, and A. Valada. Centergrasp: Object-aware implicit representation learning for simultaneous shape reconstruction and 6-dof grasp estimation. *IEEE Robotics and Automation Letters*, 2024.

[3] H.-S. Fang, C. Wang, H. Fang, M. Gou, J. Liu, H. Yan, W. Liu, Y. Xie, and C. Lu. Anygrasp: Robust and efficient grasp perception in spatial and temporal domains. *IEEE Transactions on Robotics*, 2023.

[4] Z. Weng, H. Lu, D. Kragic, and J. Lundell. Dexdiffuser: Generating dexterous grasps with diffusion models, 2024. URL https://arxiv.org/abs/2402.02989.

[5] Y. Xu, W. Wan, J. Zhang, H. Liu, Z. Shan, H. Shen, R. Wang, H. Geng, Y. Weng, J. Chen, et al. Unidexgrasp: Universal robotic dexterous grasping via learning diverse proposal generation and goal-conditioned policy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4737–4746, 2023.

[6] G.-H. Xu, Y.-L. Wei, D. Zheng, X.-M. Wu, and W.-S. Zheng. Dexterous grasp transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17933–17942, June 2024.

[7] V. Mayer, Q. Feng, J. Deng, Y. Shi, Z. Chen, and A. Knoll. Ffhnet: Generating multi-fingered robotic grasps for unknown objects in real-time. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 762–769, 2022.

[8] M. Attarian, M. A. Asif, J. Liu, R. Hari, A. Garg, I. Gilitschenski, and J. Tompson. Geometry matching for multi-embodiment grasping. In *Conference on Robot Learning*, pages 1242–1256. PMLR, 2023.

[9] P. Li, T. Liu, Y. Li, Y. Geng, Y. Zhu, Y. Yang, and S. Huang. Gendexgrasp: Generalizable dexterous grasping. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8068–8074. IEEE, 2023.

[10] L. Shao, F. Ferreira, M. Jorda, V. Nambiar, J. Luo, E. Solowjow, J. A. Ojea, O. Khatib, and J. Bohg. Unigrasp: Learning a unified model to grasp with multifingered robotic hands. *IEEE Robotics and Automation Letters*, 5(2):2286–2293, 2020.

[11] K. Li, N. Baron, X. Zhang, and N. Rojas. Efficientgrasp: A unified data-efficient learning to grasp method for multi-fingered robot hands. *IEEE Robotics and Automation Letters*, 7(4):8619–8626, 2022.

[12] C. Pan, B. Okorn, H. Zhang, B. Eisner, and D. Held. Tax-pose: Task-specific cross-pose estimation for robot manipulation. In *Conference on Robot Learning*, pages 1783–1792. PMLR, 2023.

[13] Y. Wang and J. M. Solomon. Deep closest point: Learning representations for point cloud registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[14] A. Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[15] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.

[16] A. Gupta, L. Fan, S. Ganguli, and L. Fei-Fei. Metamorph: Learning universal controllers with transformers. In *International Conference on Learning Representations*, 2022.

[17] T. Wang, R. Liao, J. Ba, and S. Fidler. Nervenet: Learning structured policy with graph neural networks. In *International conference on learning representations*, 2018.

[18] V. Kurin, M. Igl, T. Rocktäschel, W. Boehmer, and S. Whiteson. My body is a cage: the role of morphology in graph-based incompatible control. In *International Conference on Learning Representations*, 2021.

[19] C. Blake, V. Kurin, M. Igl, and S. Whiteson. Snowflake: Scaling gnns to high-dimensional continuous control via parameter freezing. *Advances in Neural Information Processing Systems*, 34:23983–23992, 2021.

[20] C. Sferrazza, D.-M. Huang, F. Liu, J. Lee, and P. Abbeel. Body transformer: Leveraging robot embodiment for policy learning. *arXiv preprint arXiv:2408.06316*, 2024.

[21] 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*, 7(1):470–477, 2021.

[22] S. Brahmbhatt, C. Ham, C. C. Kemp, and J. Hays. Contactdb: Analyzing and predicting grasp contact via thermal imaging. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8709–8719, 2019.

[23] 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*, 36(3):261–268, 2017.

[24] Dawson-Haggerty et al. trimesh. URL https://trimesh.org/.

A In-domain Evaluation

In domain, our model's mean success rate across the 3 evaluated grippers is **75.0%**, outperforming GenDexGrasp by **10.89%** and being worse than GeoMatch by **4.19%** (Table 2). Despite the minor drop in performance compared to baseline, our model shows significant improvement in out-of-domain performance.

Method	Success (%) ↑				Diversity (rad) ↑		
	ezgripper	barrett	shadowhand	**Mean**	ezgripper	barrett	shadowhand
GeoMatch [8]	75.0	**90.0**	72.5	**79.17**	0.188	0.249	0.205
GenDexGrasp [9]	43.44	71.72	**77.03**	64.11	**0.238**	0.248	**0.211**
GeoMatch++ (ours)	**82.5**	72.5	70.0	75.0	0.175	**0.342**	0.206

Table 2: In-domain success rate and diversity comparisons with GeoMatch and GenDexGrasp

B Ablations

We include results from ablation studies used to support the discussion in Section 4.2.

B.1 What is the importance of starting training from good point cloud embeddings?

Method	Success (%) ↑				Diversity (rad) ↑		
	ezgripper	barrett	shadowhand	**Mean**	ezgripper	barrett	shadowhand
From Scratch	57.5	62.5	20.0	46.7	0.222	0.197	0.116
Pretrained (Finetune)	**72.5**	**77.5**	55.0	68.33	**0.255**	0.318	**0.221**
Pretrained (Freeze)	67.5	**77.5**	**70.0**	**71.67**	0.208	**0.378**	0.184

Table 3: Comparison of weights for point cloud GCN embeddings

B.2 Does including robot morphology improve out-of-domain generalization?

Method	Success (%) ↑				Diversity (rad) ↑		
	ezgripper	barrett	shadowhand	**Mean**	ezgripper	barrett	shadowhand
Point Cloud Only	27.5	70.0	50.0	49.16	**0.270**	**0.429**	0.141
PC and Morphology	**67.5**	**77.5**	**70.0**	**71.67**	0.208	0.378	**0.184**

Table 4: Comparison of using only point clouds vs. using point clouds and morphology

B.3 What is the contribution of different morphology features?

Method	Success (%) ↑				Diversity (rad) ↑		
	ezgripper	barrett	shadowhand	**Mean**	ezgripper	barrett	shadowhand
Joints Only	62.5	72.5	62.5	65.83	0.209	**0.390**	0.199
Links Only	57.5	67.5	57.5	60.83	**0.244**	0.271	**0.215**
Final	**67.5**	**77.5**	**70.0**	**71.67**	0.208	0.378	0.184

Table 5: Comparison of different morphology features

C Morphology Graph Representation

We formulate the morphology graph from the URDF description of each end-effector. Nodes of the graph are links and edges are joints. We consider both revolute and fixed joints as edges. Two nodes are connected if they are respectively the parent and child link of a joint. Self-connections are added in the graph. The

offset feature is obtained from the $< joint >< origin >< xyz >$ element of a joint. The feature is attributed to the child link of the joint. End-effectors may have a root link that is connected to a joint with multiple children links; in this case, the offset feature is attributed to the child link first listed in the kinematic chain. The least volume rectangular bounding boxes of links are estimated from the link meshes using the Trimesh library [24]. We determine the morphology features most useful for learning empirically.

D Implementation Details

We use $N = 6$ for the number of keypoint-contact pairs. The keypoints are chosen to lie on different links to capture diverse morphological information and to be semantically consistent across end-effectors, but otherwise satisfy no other constraint.

Experiments are conducted on a RTX3090 GPU. The model is trained using Adam with a learning rate of 0.00005 and betas of (0.9, 0.99), for 150 epochs with batch size 32. The parameters for the GCNs and autoregressive module are similar to GeoMatch [8]. GCNs have 3 hidden graph convolution layers of dimension 256, and a final output linear layer of dimension 512. Each autoregressive MLP contains 3 hidden layers of dimension 256 and outputs a contact likelihood map of size 2048. We use the same parameters for the object-gripper correspondence transformers as the transformer module in DCP [13], but with input dimensions of object point cloud size $S_O = 2048$ and morphology graph size $S_M = 32$.

We use the same inverse kinematics optimization and IsaacGym evaluation setup as [8].