

A Additional Experimental Details

A.1 Robustness Experiments

To investigate the robustness of the learned representations, we conducted a set of experiments where we tested out our trained model with noisy object point clouds, partial point clouds and partial point clouds with noise added as well:

Noisy point clouds. For this experiment, we processed the 10 object point clouds of our evaluation set by adding Gaussian noise with standard deviation 0.001 and clipping to a one standard deviation interval, to each of the points. Our evaluation process was then repeated with these as input **zero-shot**, i.e. grasps were generated and evaluated with the same process.

Partial point clouds. For this experiment, we emulated a table top scenario where objects placed on a table would be missing the bottom of their surface. To achieve this, for each object point cloud, we defined a z-plane

$$z_{\text{thres}} = \frac{z_{\text{max}} - z_{\text{min}}}{6},$$

where z_{min} , z_{max} are the minimum and maximum z-value found in each object point cloud respectively. We then remove all points with $z < z_{\text{thres}}$ in order to emulate such a table effect. The resulting point clouds are again used **zero-shot** on our model to predict grasps.

Noisy partial point clouds. For this experiments, the table top emulating partial point clouds generated for the previous experiment are augmented with Gaussian noise of standard deviation 0.001 and clipping to a one standard deviation interval. Grasping generation occurs again **zero-shot** on our model, and the evaluation process remains the same as all other experiments.

Comparative results for all 3 experiments against noiseless inputs can be viewed in Tab. 3.

Augmentation	Success (%) \uparrow			Diversity (rad) \uparrow		
	ezgripper	barrett	shadowhand	ezgripper	barrett	shadowhand
noiseless	72.5	90.0	75.0	0.188	0.249	0.205
noisy	75	95.0	62.5	0.183	0.245	0.196
partial	67.5	67.5	65.0	0.181	0.207	0.197
noisy partial	65	75.0	62.5	0.143	0.227	0.212

Table 3: Comparisons between noiseless, noisy, partial, and noisy partial object point cloud inputs.

We observe that our model generally demonstrates robustness to noise with performance actually increasing in two out of three evaluated end-effectors. Partial point clouds cause the performance to drop as expected, however the model is still performing at a good level at multi-embodiments.

A.2 PointNet++ Ablation

Our choice of GCN as a geometry encoder is, of course, not the single architectural option available for representing 3D geometry features, with PointNet++ [20] being a popular choice in the literature. In this ablation, we investigate the efficacy of GCN in the multi-embodiment grasping setup compared to PointNet++ by replacing both our GCN object and end-effector encoders with a PointNet++ architecture¹.

Results in Tab. 4 show that the GCN encoder variant outperforms the PointNet++ one for the 3-finger and 5-finger gripper while performs on par with it for the 2-finger gripper. The GCN variant is also showing higher diversity of grasps for all 3 end-effectors.

A.3 Non-Shared Weights Ablation

For our main method, we assumed shared weights between the representations used in the autoregressive modules predicting each keypoint contact. However, it is of interest to investigate how

¹We used the implementation from https://github.com/yanx27/Pointnet_Pointnet2_pytorch

Encoder	Success (%) \uparrow			Diversity (rad) \uparrow		
	ezgripper	barrett	shadowhand	ezgripper	barrett	shadowhand
GCN [28]	75.0	90.0	72.5	0.188	0.249	0.205
PointNet++ [20]	75.0	70.0	65.0	0.154	0.223	0.151

Table 4: Comparison between GCN and PointNet++ encoder choices.

performance gets impacted if each autoregressive module is free to influence geometry representations for the keypoint it is responsible for. We thus, disentangled encoding weights for each of the autoregressive modules by passing in a separate end-effector encoder in each.

Ablation	Success (%) \uparrow			Diversity (rad) \uparrow		
	ezgripper	barrett	shadowhand	ezgripper	barrett	shadowhand
Shared weights	75.0	90.0	72.5	0.188	0.249	0.205
Non-shared weights	70.0	82.5	60.0	0.165	0.259	0.163

Table 5: Comparison between shared and non-shared weights of the end-effector encoder for autoregressive learning.

The comparison is provided in Tab. 5 and indicate that training end-to-end with a shared end-effector encoder for all keypoint predictions, is still a significantly better performant choice. The shared weights variant performs **5%-12.5% better** among the 3 sample embodiments than the non-shared weights ablation.

B Implementation Details

Implementation of all experiments was done using an Adam optimizer with learning rate of 1e-4 for 200 epochs. An assortment of GPU was used, namely RTX3090, V100, T4. Other hyperparameters used were provided in the main paper but for completeness, we include all hyperparameters here. The GNN used had 3 hidden layers of size 256. The output feature size of the GNN encoder was 512. The two parts of the loss were weighed by 0.5 each while the two positive weights used for the two BCE losses were 500 and 200 for the independent distributions and marginals respectively. The dataset used was the subset of MultiDex used by [12] to train the CMap-CVAE model of their approach, which contains 50,802 diverse grasping poses for 5 hands and 58 objects from YCB and ContactDB. The training set contained 38 objects and the validation set the remaining 10. The projection layer was a Linear layer without bias with an output dimension of 64 and each of the MLP autoregressive modules had 3 hidden layers of size 256.

For the IK, SciPy’s TRF algorithm was used where each resulting set of predicted keypoints was moved 5mm away from the surface of the object on the direction of the normal in order to form a pre-grasp pose. The initial pose guess provided, was a heuristic calculated by orienting the palm of the gripper to align with the negative of the normal on the object surface at the closest surface point. For evaluation, 4 grasps per object-gripper pair were sampled by selected the top-[0, 20, 50, 100] most likely keypoint 0.

The Isaac Gym based evaluation scripts from [12] were used as is, aside from the one Adam step of force closure where the step size used was 0.05 in order to make the force closure smoother and less abrupt.