

## 405 A Additional Experimental Details

### 406 A.1 Robustness Experiments

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

410 **Noisy point clouds.** For this experiment, we processed the 10 object point clouds of our evaluation  
411 set by adding Gaussian noise with standard deviation 0.001 and clipping to a one standard deviation  
412 interval, to each of the points. Our evaluation process was then repeated with these as input **zero-**  
413 **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

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

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

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

421 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	<b>75.0</b>	0.188	0.249	0.205
noisy	<b>75</b>	<b>95.0</b>	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.

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

### 425 A.2 PointNet++ Ablation

426 Our choice of GCN as a geometry encoder is, of course, not the single architectural option avail-  
427 able for representing 3D geometry features, with PointNet++ [20] being a popular choice in the  
428 literature. In this ablation, we investigate the efficacy of GCN in the multi-embodiment grasping  
429 setup compared to PointNet++ by replacing both our GCN object and end-effector encoders with a  
430 PointNet++ architecture<sup>1</sup>.

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

### 434 A.3 Non-Shared Weights Ablation

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

<sup>1</sup>We used the implementation from [https://github.com/yanx27/Pointnet\\_Pointnet2\\_pytorch](https://github.com/yanx27/Pointnet_Pointnet2_pytorch)

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

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

437 performance gets impacted if each autoregressive module is free to influence geometry representa-  
438 tions for the keypoint it is responsible for. We thus, disentangled encoding weights for each of the  
439 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	<b>75.0</b>	<b>90.0</b>	<b>72.5</b>	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 au-  
toregressive learning.

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

## 444 B Implementation Details

445 Implementation of all experiments was done using an Adam optimizer with learning rate of 1e-4 for  
446 200 epochs. An assortment of GPU was used, namely RTX3090, V100, T4. Other hyperparameters  
447 used were provided in the main paper but for completeness, we include all hyperparameters here.  
448 The GNN used had 3 hidden layers of size 256. The output feature size of the GNN encoder was  
449 512. The two parts of the loss were weighed by 0.5 each while the two positive weights used for  
450 the two BCE losses were 500 and 200 for the independent distributions and marginals respectively.  
451 The dataset used was the subset of MultiDex used by [12] to train the CMap-CVAE model of their  
452 approach, which contains 50,802 diverse grasping poses for 5 hands and 58 objects from YCB and  
453 ContactDB. The training set contained 38 objects and the validation set the remaining 10. The  
454 projection layer was a Linear layer without bias with an output dimension of 64 and each of the  
455 MLP autoregressive modules had 3 hidden layers of size 256.

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

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