Supplementary Material

1 Network Details

In this section, we give more details on the proposed scale balanced grasp detection approach.

1.1 Transformer-based Point Encoder

For the point-cloud encoder, we build a transformer-based architecture modified from [1] to model global relationship among points. Specifically, [1] proposes a set operator, namely point transformer layer, to capture the relationship between a point and its neighbors, described in Equation (1):

$$y_i = \sum_{x_j \in X(i)} Softmax\big(\gamma(\varphi(x_i) - \psi(x_j) + \delta)\big) \odot \big(\alpha(x_j) + \delta\big) \tag{1}$$

where $\gamma, \varphi, \psi, \alpha$ are MLPs for feature transformation and δ is a position encoding. $X(i)$ is the set of neighboring points of x_i.

Considering the trade-off between the encoder performance and computational complexity, we add point transformer layers to a two-layer PointNet++ network [2], as shown in Figure 1.

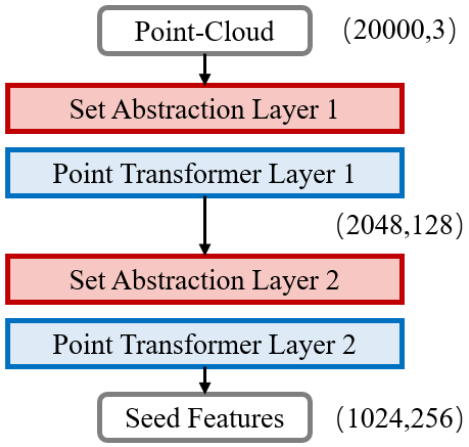

Figure 1: Architecture of the point encoder.

The initial point-cloud is sampled and grouped by the set abstraction layer [2] and the point transformer layer is employed to encode long-range dependencies for points and its neighbors.

1.2 Instance Segmentation Network

For the instance segmentation network adopted in Object Balanced Sampling (OBS), inspired by [3], we construct an architecture to deal with point-clouds rather than images, where the point encoder introduced in Sec 1.1 is used as the backbone. A decoder is added with two heads to predict foreground masks and object center offsets, on which we apply the mean-shift clustering algorithm to obtain object mask prediction. The whole process is shown in Figure 2. Only the deep seeding network is used for training and inference in our experiments and if RGB information is available, the region refinement network can be further integrated for better segmentation results.

6th Conference on Robot Learning (CoRL 2022), Auckland, New Zealand.

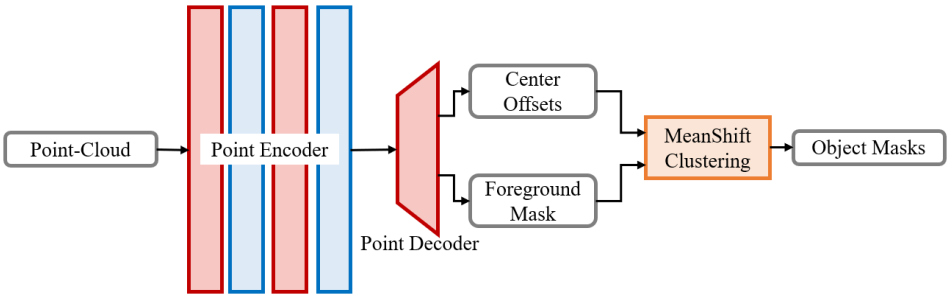

Figure 2: Framework of instance segmentation.

1.3 Loss Functions

Following [4], the loss of our network is composed of an approach term and a rotation term. The approach term is:

$$L^{Approach}(c_i, s_{ij}) = \frac{1}{N_{cls}} \sum_i L_{cls}(c_i, c_i^*) \\ +\lambda_1 \frac{1}{N_{reg}} \sum_i \sum_j c_i^* \mathbf{1}(|v_{ij}, v_{ij}^*| < 5°) L_{reg}(s_{ij}, s_{ij}^*) \tag{2}$$

where c_i is the graspable prediction result, s_{ij} is the j-th view confidence for point i and v_{ij} is the view direction. The rotation term is:

$$L^{Rotation}(R_{ij}, S_{ij}, W_{ij}) = \sum_{d=1}^{K} (\frac{1}{N_{cls}} \sum_{ij} L_{cls}^d(R_{ij}, R_{ij}^*) \\ +\lambda_2 \frac{1}{N_{reg}} \sum_{ij} L_{reg}^d(S_{ij}, S_{ij}^*) + \lambda_3 \frac{1}{N_{reg}} \sum_{ij} L_{reg}^d(W_{ij}, W_{ij}^*)) \tag{3}$$

where R_{ij}, S_{ij}, Wij denote the rotation degrees, grasp confidence scores and gripper widths respectively. d is the depth of the approaching direction.

2 Implementation Details

The raw point-cloud is sampled to $20,000$ points as input. After the point encoder, we sample $1,024$ candidates with 256-dimensional features. For the loss in the approach head, we discrete the approach direction to 300 views and employ the $L1$ loss to regress the view score. The cross-entropy loss is used for graspable discovery. In MsCG, we set up four cylinders with radii $r = 0.02m, 0.04m, 0.06m, 0.08m$. 64 points are sampled from each cylinder and encoded to a 256-dimensional feature. For the loss in the operation head, we divide the plane rotation to 12 bins and use the cross-entropy loss for classification. For the width and grasp score, we predict them during regression with the $L1$ loss. The trade-off hyper-parameter α for the operation loss is set to 0.2. In SBL, we divide the max width of the gripper into $T = 32$ bins.

3 Visualization of Grasps

Some results of the scale balanced grasp detection approach are visualized in Figure 3. The top-50 ranked grasps of each scene are displayed, where successful grasps are shown in red while grasps which collide with the scene or do not satisfy the force closure condition are shown in purple and blue respectively.

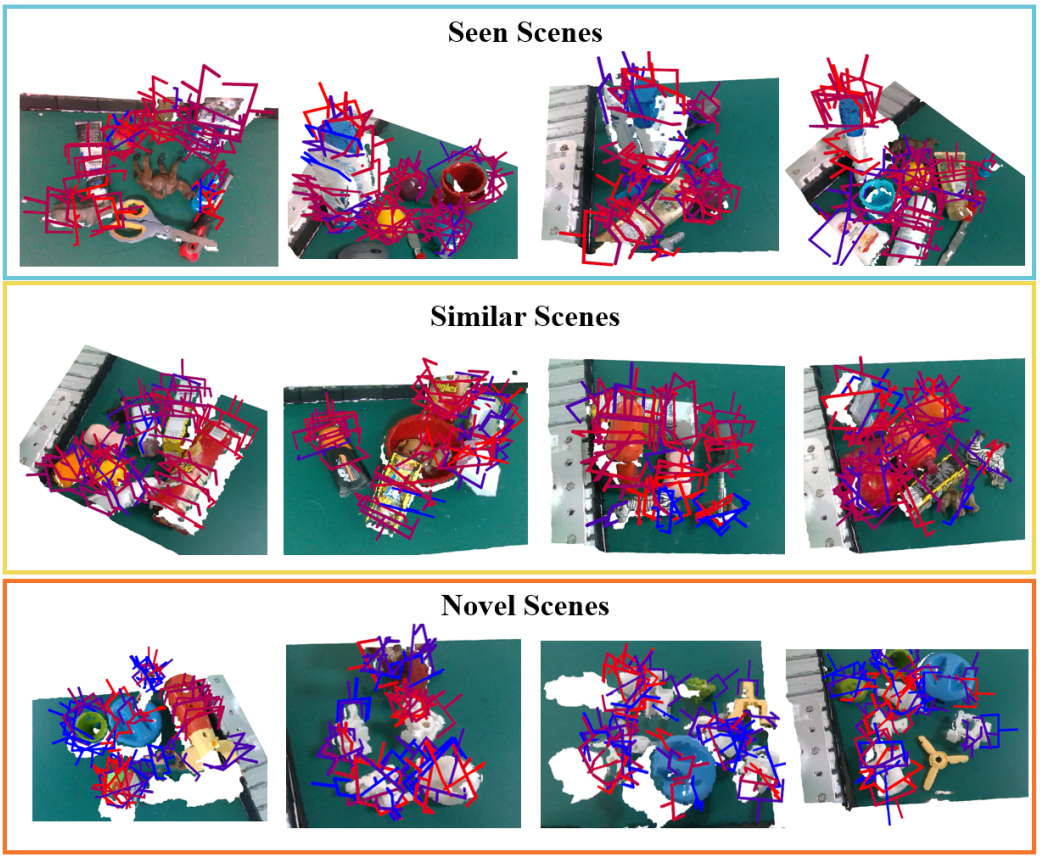

Figure 3: Grasp visualization on seen, similar and novel scenes.

4 More Experiments

4.1 Small-scale Grasp in terms of Threshold

To describe the grasp scale in convenience, we divide the max width of the gripper to three equal sections to denote small-, medium- and large-scale grasps respectively. Here, we test different thresholds for small-scale grasps, and besides the original threshold at $4cm$, a smaller one ($3cm$) and a larger one ($5cm$) are additionally considered. The results are shown in Table 1. From the table, we can see that when the threshold varies, the proposed approach consistently delivers performance gains on the baseline.

Model	**AP**$_{0cm-3cm}$			**AP**$_{0cm-4cm}$			**AP**$_{0cm-5cm}$		
	Seen	**Similar**	**Novel**	**Seen**	**Similar**	**Novel**	**Seen**	**Similar**	**Novel**
Baseline	3.06	0.38	0.62	9.44	5.15	4.91	19.83	14.85	11.79
Ours	4.75	0.83	1.30	13.47	6.23	7.60	24.81	16.99	13.52
Ours (with OBS)	7.21	1.39	1.74	18.29	10.03	9.29	30.00	22.38	15.83

Table 1: Results for small-scale grasps with different thresholds.

4.2 Grasp Results at Object-level

To validate the ability of the proposed approach to generate grasps for objects at different scales, we make evaluation at object-level where we choose the top-5 ranked grasps for each object and use the AP averaged in objects as the metric. The results are shown in Table 2, and our approach significantly improves the grasp quality at object-level.

Model	Seen Objects			Similar Objects			Novel Objects		
	AP	**AP**$_{0.8}$	**AP**$_{0.4}$	**AP**	**AP**$_{0.8}$	**AP**$_{0.4}$	**AP**	**AP**$_{0.8}$	**AP**$_{0.4}$
Baseline	37.12	44.74	30.36	32.05	39.21	25.45	20.19	25.20	11.92
Ours	40.90	48.75	34.56	35.84	43.16	29.96	23.07	28.74	14.45
Ours (with OBS)	42.84	51.04	36.18	37.64	45.34	31.46	24.76	30.73	15.52

Table 2: Grasp results at object-level.

4.3 Comparison between NcM and Fine-tuning

The proposed NcM module is a one-phase method to address the Sim2Real gap. Compared to NcM, a straightforward counterpart to mitigate this gap is to conduct a two-phase procedure, *i.e.* Clean-Train and Noisy-Finetune (CTNF), where the model is firstly trained on synthetic data and then fine-tuned on raw data. Although effective, CTNF incurs additional problems for expected results, including specifically designing the training strategy (*e.g.* freezing some layers or training different parts of the network with individual learning rates) and carefully setting the relevant hyper-parameters (*e.g.* training iterations and learning rate). In contrary, NcM aims to bridge the domain gap by generating more data which mix synthetic and raw scenes into single samples at instance-level so that the trained model can directly work without fine-tuning.

We give a comparison between NcM and CTNF. In CTNF, we train our model using synthetic data for 18 epochs (the same as in NcM) and finetune it using raw data for another 12 epochs, with good convergences achieved at both phases. During fine-tuning, all the layers of the model are adjusted by a small learning rate (1/10 to that used in training). The results are shown below.

Model	Seen				Similar				Novel			
	AP$_S$	**AP**$_M$	**AP**$_L$	**Mean**	**AP**$_S$	**AP**$_M$	**AP**$_L$	**Mean**	**AP**$_S$	**AP**$_M$	**AP**$_L$	**Mean**
Noisy-Train	12.69	46.25	61.78	40.24	6.00	36.88	52.55	31.81	7.06	16.38	23.27	15.57
NcM	13.47	48.12	61.81	**41.13**	6.23	37.90	53.89	**32.67**	7.60	17.04	23.10	15.91
CTNF	14.87	44.75	61.60	40.41	6.66	36.99	53.60	32.42	7.75	16.95	23.44	**16.05**

Table 3: Comparison between NcM and CTNF.

In the experiments, we can see that NcM works comparably with CTNF (performs better on the whole) but in a more efficient manner. The inferiority of CTNF is mainly caused by the problem of catastrophic forgetting due to the large gap between the two domains.

4.4 Mix Ratio in NcM

We conduct an ablation study on the value of the mix ratio of NcM, where the ratio ranges from 0 to 100% clean data with a step of 25%. The results are shown below.

Mix Ratio	Seen				Similar				Novel			
	AP$_S$	**AP**$_M$	**AP**$_L$	**Mean**	**AP**$_S$	**AP**$_M$	**AP**$_L$	**Mean**	**AP**$_S$	**AP**$_M$	**AP**$_L$	**Mean**
0% clean (w/o NcM)	12.69	46.25	61.78	40.24	6.00	36.88	52.55	31.81	7.06	16.38	23.27	15.57
25% clean (w/ NcM)	13.53	48.45	62.23	**41.40**	6.95	39.72	53.81	**33.49**	7.90	17.35	23.27	**16.17**
50% clean (w/ NcM)	13.47	48.12	61.81	41.13	6.23	37.90	53.89	32.67	7.60	17.04	23.10	15.91
75% clean (w/ NcM)	12.74	47.31	61.78	40.61	5.80	37.07	54.17	32.35	7.03	17.31	23.12	15.82
100% clean (w/o NcM)	9.62	38.87	59.86	36.12	4.10	30.23	54.41	29.58	5.17	14.67	20.64	13.49

Table 4: Ablation study on the value of the mix ratio of NcM.

When the NcM module is introduced, it delivers a consistent improvement no matter what the mix ratio is. The value of 25% clean data achieves the best performance and the performance only with clean data (the value is set at 100%) is largely inferior to the others because of the Sim2Real gap.

References

[1] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun. Point transformer. In *IEEE/CVF International Conference on Computer Vision*, 2021.

[2] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, 2017.

[3] C. Xie, Y. Xiang, A. Mousavian, and D. Fox. Unseen object instance segmentation for robotic environments. *IEEE Transactions on Robotics*, 37(5):1343–1359, 2021.

[4] H. Fang, C. Wang, M. Gou, and C. Lu. Graspnet-1billion: A large-scale benchmark for general object grasping. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.