PatchComplete: Learning Multi-Resolution Patch Priors for 3D Shape Completion on Unseen Categories
Supplementary Material

Yuchen Rao **Yinyu Nie** **Angela Dai**

Technical University of Munich

{yuchen.rao, yinyu.nie, angela.dai}@tum.de

In this supplementary material, we describe the data generation (Sec. A), settings for baseline methods (Sec. B), evaluation of category-wise error bars on ShapeNet and ScanNet (Sec. C), evaluation on seen categories for all the methods on ScanNet (Sec. D), additional ablation studies (Sec. E), network parameters and specifications (Sec. F), and additional qualitative results (Sec. G).

A Data Generation

ShapeNet [2] We use ShapeNet[1] to test our performance on synthetic data. In order to generate watertight meshes as ground truth, we first normalize ShapeNet CAD models, and render depth maps under 20 different viewpoints for each model. We then use volumetric fusion [4] to generate 32^3 truncated signed distance fields (TSDFs) with truncation value as 2.5 voxel units. Finally, we choose 4 views of TSDF as the input, which mimic the partial scan in real data (e.g., ScanNet). The main idea can be referred to [8][2].

We split the training and testing object categories on ShapeNet as follows. The 18 training categories are *table, chair, sofa, cabinet, clock, bookshelf, piano, microwave, stove, file cabinet, trash bin, bowl, display, keyboard, dishwasher, washing machine, pots, faucet, and guitar*; and the 8 novel testing categories are *bathtub, lamp, bed, bag, printer, laptop, bench, and basket*.

ScanNet [5] We use ScanNet[3] to test our method on real-world data. The inputs are directly extracted from ScanNet scenes based on the bounding box annotations from Scan2CAD [1][4]. We keep their real scale and convert them to 32^3 voxel grids with truncation value at 3 voxel units, and save their voxel size separately. These inputs could contain walls, floors, or other cluttered backgrounds, which are transformed to canonical space to be aligned with the ShapeNet model coordinate system. The ground-truths are the corresponding complete and watertight ShapeNet meshes based on Scan2CAD annotations, which are generated with the similar method as above.

We split the training and testing object categories on ScanNet as follows. The 8 training categories are *chair, table, sofa, trash bin, cabinet, bookshelf, file cabinet, and monitor*; and the 6 novel testing categories are *bathtub, lamp, bed, bag, basket, and printer*, and each category has more than 50 samples for testing.

[1]The license can be found here: https://shapenet.org/, received permission after registration without personally identifiable information or offensive content

[2]https://github.com/yinyunie/depth_renderer

[3]The license can be found here: https://github.com/ScanNet/ScanNet, filled out an agreement without personally identifiable information or offensive content

[4]The license can be found here::https://github.com/skanti/Scan2CAD, filled out an agreement without personally identifiable information or offensive content

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

Alternative Train Category Splits Here are two new category splits for the last ablation study. For Category Split 1, the 8 novel testing categories are *trash bin, bed, piano bench, chair, monitor, lamp, laptop, washing machine*. For Category Split 2, the 8 novel testing categories are *basket, bookshelf, bowl, cabinet, laptop, pot, sofa, stove*.

B Baseline Comparisons

We use the authors' original implementations and hyperparameters in all the baselines for fair comparisons.

3D-EPN [6] 3D-EPN is a two-stage network, which completes partial 3D scans first and then reconstructs the completed shapes to a higher resolution by retrieving priors from a category-wise shape pool. In our case, priors for novel categories are not accessible, thus, we only compare its 3D Encoder-Predictor Network (the 3D completion model) on our dataset.

Wallace and Hariharan [9] (Few-Shot) This method uses a few-shot learning strategy for single view completion with averaged shape prior for each category. For a fair comparison with other works, we adapt it to a zero-shot learning mechanism here. We pre-compute the averaged shape priors for each training category; during training, we use two voxel encoder modules in parallel for the occupied voxel grids inputs and the averaged shape priors based on the input category; in the testing step, since we cannot provide shape priors for novel categories, we average shape prior from all the training categories, and use this averaged shape prior as input to the prior encoder module, along with the testing samples for shape completion.

IF-Net [3] IF-Net can predict implicit shape representations conditioned on different input modalities. (e.g., voxels, point clouds). We use 128^3 occupied surface voxel grids as inputs, and use point clouds sampled from watertight ShapeNet meshes as the ground-truths for training and testing. We also normalize the ground-truth meshes from the ScanNet dataset to sample points.

AutoSDF [7] AutoSDF learns latent patch priors using VQ-VAE along with a transformer-based autoregressive model for 3D shape completion, and manually picks the unknown patches during testing. Following their settings, we apply their method by using ground-truth SDFs as the training data; during testing on the ShapeNet data, we choose the patches that have more than 400 voxel grids (each patch has 8^3 voxel grids) with negative signs as the unknown patches (unseen parts) that need to be generated.

Note that since AutoSDF work focuses on multi-model shape completion and produces multiple output possibilities, we report the performance of only the best prediction among the nine given an oracle to indicate the best (highest IoU value with respect to ground truth).

Furthermore, as there are no absolutely unknown patches for ScanNet scans because of the cluttered environments, we use the pipeline of their single view reconstruction task. We first replace their *ResNet* in *resnet2vq_model* with three 3D encoders (the same as 3D-EPN encoders) to extract the encoding features of desired dimensions; then we train this modified model along with the pre-trained *pvq_vae_model* with partial ScanNet inputs; finally we test our partial ScanNet inputs along with all the pre-trained models: *resnet2vq_model*, *pvq_vae_model*, and *rand_tf_model*.

C Category-wise Evaluations with Error Bars

Table 1 and Table 2 show the category-wise error bars on ShapeNet and ScanNet respectively; each method is run $n = 2$ times to obtain the error bars.

D Evaluation on Seen Categories

Table 3 shows the comparisons on seen train categories with state of the art on real-world data from ScanNet [5]. We evaluate 1060 samples for 7 seen categories including: *chair, table, sofa, trash bin, cabinet, bookshelf, and monitor*; categories are selected as those which have more than 50 test samples.

Table 1: Quantitative comparisons with state of the art on ShapeNet [2].

	Chamfer Distance ($\times 10^2$) ↓					IoU↑				
	3D-EPN [6]	Few-Shot [9]	IF-Nets [3]	Auto-SDF [7]	Ours	3D-EPN [6]	Few-Shot [9]	IF-Nets [3]	Auto-SDF [7]	Ours
Bag	5.01 $\pm 2e^{-1}$	8.00 $\pm 4e^{-1}$	4.77 $\pm 3e^{-1}$	5.81 $\pm 2e^{-1}$	**3.94** $\pm 2e^{-2}$	0.738 $\pm 5e^{-3}$	0.561 $\pm 2e^{-2}$	0.698 $\pm 2e^{-3}$	0.563 $\pm 1e^{-2}$	**0.776** $\pm 3e^{-3}$
Lamp	8.07 $\pm 6e^{-1}$	15.10 $\pm 2e^{-1}$	5.70 $\pm 5e^{-1}$	6.57 $\pm 1e^{-1}$	**4.68** $\pm 2e^{-2}$	0.472 $\pm 1e^{-2}$	0.254 $\pm 2e^{-3}$	0.508 $\pm 1e^{-3}$	0.391 $\pm 1e^{-2}$	**0.564** $\pm 3e^{-3}$
Bathtub	4.21 $\pm 1e^{-2}$	7.05 $\pm 1e^{-1}$	4.72 $\pm 8e^{-2}$	5.17 $\pm 2e^{-2}$	**3.78** $\pm 2e^{-2}$	0.579 $\pm 2e^{-2}$	0.457 $\pm 4e^{-4}$	0.550 $\pm 9e^{-3}$	0.410 $\pm 8e^{-3}$	**0.663** $\pm 9e^{-4}$
Bed	5.84 $\pm 3e^{-2}$	10.03 $\pm 2e^{-1}$	5.34 $\pm 2e^{-1}$	6.01 $\pm 1e^{-1}$	**4.49** $\pm 2e^{-4}$	0.584 $\pm 4e^{-3}$	0.396 $\pm 1e^{-3}$	0.607 $\pm 2e^{-3}$	0.446 $\pm 1e^{-2}$	**0.668** $\pm 2e^{-3}$
Basket	7.90 $\pm 4e^{-1}$	8.72 $\pm 9e^{-2}$	**4.44** $\pm 1e^{-2}$	6.70 $\pm 3e^{-1}$	5.15 $\pm 3e^{-1}$	0.540 $\pm 1e^{-2}$	0.406 $\pm 5e^{-3}$	0.502 $\pm 1e^{-4}$	0.398 $\pm 1e^{-2}$	**0.610** $\pm 3e^{-3}$
Printer	5.15 $\pm 1e^{-2}$	9.26 $\pm 5e^{-2}$	5.83 $\pm 1e^{-1}$	7.52 $\pm 1e^{-1}$	**4.63** $\pm 7e^{-2}$	0.736 $\pm 6e^{-3}$	0.567 $\pm 3e^{-3}$	0.705 $\pm 4e^{-3}$	0.499 $\pm 3e^{-2}$	**0.776** $\pm 2e^{-4}$
Laptop	3.90 $\pm 1e^{-1}$	10.35 $\pm 3e^{-1}$	6.47 $\pm 8e^{-1}$	4.81 $\pm 2e^{-1}$	**3.77** $\pm 9e^{-2}$	0.620 $\pm 6e^{-3}$	0.313 $\pm 2e^{-2}$	0.583 $\pm 1e^{-3}$	0.511 $\pm 1e^{-2}$	**0.638** $\pm 8e^{-3}$
Bench	4.54 $\pm 3e^{-2}$	8.11 $\pm 8e^{-1}$	5.03 $\pm 9e^{-1}$	4.31 $\pm 5e^{-2}$	**3.70** $\pm 1e^{-2}$	0.483 $\pm 1e^{-2}$	0.272 $\pm 1e^{-2}$	0.497 $\pm 4e^{-3}$	0.395 $\pm 3e^{-3}$	**0.539** $\pm 3e^{-4}$
Inst-Avg	5.48 $\pm 2e^{-1}$	9.75 $\pm 9e^{-2}$	5.37 $\pm 1e^{-1}$	5.76 $\pm 3e^{-2}$	**4.23** $\pm 4e^{-2}$	0.582 $\pm 9e^{-3}$	0.386 $\pm 1e^{-3}$	0.574 $\pm 4e^{-5}$	0.446 $\pm 6e^{-3}$	**0.644** $\pm 1e^{-3}$
Cat-Avg	5.58 $\pm 2e^{-1}$	9.58 $\pm 1e^{-1}$	5.29 $\pm 1e^{-1}$	5.86 $\pm 5e^{-3}$	**4.27** $\pm 5e^{-2}$	0.594 $\pm 8e^{-3}$	0.403 $\pm 1e^{-3}$	0.581 $\pm 3e^{-4}$	0.452 $\pm 7e^{-3}$	**0.654** $\pm 1e^{-3}$

Table 2: Quantitative comparisons with state of the art on ScanNet [5].

	Chamfer Distance ($\times 10^2$) ↓					IoU↑				
	3D-EPN [6]	Few-Shot [9]	IF-Nets [3]	Auto-SDF [7]	Ours	3D-EPN [6]	Few-Shot [9]	IF-Nets [3]	Auto-SDF [7]	Ours
Bag	8.83 $\pm 1e^{-1}$	9.10 $\pm 2e^{-1}$	8.96 $\pm 1e^{-1}$	9.30 $\pm 3e^{-2}$	**8.23** $\pm 4e^{-2}$	0.537 $\pm 1e^{-2}$	0.449 $\pm 6e^{-3}$	0.442 $\pm 5e^{-3}$	0.487 $\pm 1e^{-4}$	**0.583** $\pm 8e^{-3}$
Lamp	14.27 $\pm 2e^{0}$	11.88 $\pm 3e^{-1}$	10.16 $\pm 2e^{-1}$	11.17 $\pm 3e^{-2}$	**9.42** $\pm 1e^{-2}$	0.207 $\pm 4e^{-2}$	0.196 $\pm 5e^{-4}$	0.249 $\pm 4e^{-3}$	0.244 $\pm 9e^{-4}$	**0.284** $\pm 2e^{-2}$
Bathtub	7.56 $\pm 7e^{-2}$	7.77 $\pm 1e^{-1}$	7.19 $\pm 5e^{-2}$	7.84 $\pm 1e^{-2}$	**6.77** $\pm 1e^{-1}$	0.410 $\pm 7e^{-3}$	0.382 $\pm 3e^{-3}$	0.395 $\pm 4e^{-3}$	0.366 $\pm 1e^{-3}$	**0.480** $\pm 2e^{-3}$
Bed	7.76 $\pm 8e^{-2}$	9.07 $\pm 1e^{-1}$	8.24 $\pm 1e^{-2}$	7.91 $\pm 5e^{-2}$	**7.24** $\pm 1e^{-1}$	0.478 $\pm 7e^{-3}$	0.349 $\pm 1e^{-2}$	0.449 $\pm 9e^{-3}$	0.380 $\pm 2e^{-3}$	**0.484** $\pm 3e^{-3}$
Basket	7.74 $\pm 1e^{-1}$	8.02 $\pm 3e^{-1}$	6.74 $\pm 4e^{-2}$	7.54 $\pm 2e^{-2}$	**6.60** $\pm 1e^{-1}$	0.365 $\pm 9e^{-3}$	0.343 $\pm 5e^{-3}$	0.427 $\pm 4e^{-3}$	0.361 $\pm 2e^{-3}$	**0.455** $\pm 3e^{-3}$
Printer	8.36 $\pm 7e^{-1}$	8.30 $\pm 3e^{-1}$	8.28 $\pm 2e^{-1}$	9.66 $\pm 2e^{-2}$	**6.84** $\pm 2e^{-1}$	0.630 $\pm 4e^{-2}$	0.622 $\pm 7e^{-4}$	0.607 $\pm 1e^{-2}$	0.499 $\pm 1e^{-4}$	**0.705** $\pm 2e^{-2}$
Inst-Avg	8.60 $\pm 2e^{-1}$	8.83 $\pm 2e^{-2}$	8.12 $\pm 7e^{-2}$	8.56 $\pm 2e^{-2}$	**7.38** $\pm 6e^{-2}$	0.441 $\pm 2e^{-3}$	0.387 $\pm 1e^{-3}$	0.426 $\pm 3e^{-3}$	0.386 $\pm 1e^{-4}$	**0.498** $\pm 9e^{-3}$
Cat-Avg	9.09 $\pm 3e^{-1}$	9.02 $\pm 8e^{-2}$	8.26 $\pm 8e^{-2}$	8.90 $\pm 2e^{-2}$	**7.52** $\pm 2e^{-2}$	0.440 $\pm 3e^{-3}$	0.386 $\pm 6e^{-3}$	0.426 $\pm 7e^{-3}$	0.389 $\pm 3e^{-4}$	**0.495** $\pm 5e^{-3}$

Table 3 shows that our performance on seen categories is on par with state of the art, particularly when evaluating category averages, as our learned multiresolution priors maintain robustness across categories. Note that similar to the previous evaluation, AutoSDF results are reported as the best among their nine predictions with the highest IoU value given an oracle to indicate the best choice. Our method thus achieves performance on par with state of the art on seen categories, and notably improves shape completion for unseen categories.

Table 3: Quantitative comparison with state of the art on real-world ScanNet [5] shape completion for seen categories. We bold the best results and underline the second best results in the table.

	Chamfer Distance ($\times 10^2$) ↓					IoU↑				
	3D-EPN [6]	Few-Shot [9]	IF-Nets [3]	Auto-SDF [7]	Ours	3D-EPN [6]	Few-Shot [9]	IF-Nets [3]	Auto-SDF [7]	Ours
Trash Bin	5.03	5.65	5.23	4.48	**4.44**	0.61	**0.70**	0.62	0.66	0.68
Chair	9.99	6.88	7.93	**6.00**	7.14	0.40	0.46	0.43	**0.49**	0.45
Bookshelf	4.87	4.33	5.17	4.12	**3.80**	0.53	**0.65**	0.58	0.61	0.61
Table	8.74	7.13	10.15	6.72	**6.60**	0.47	0.50	0.46	0.49	**0.54**
Cabinet	4.60	4.36	5.64	4.53	**4.17**	0.76	**0.80**	0.74	0.78	0.79
Sofa	4.94	**4.28**	7.87	4.58	4.53	0.69	**0.75**	0.67	0.72	0.73
Monitor	5.75	4.98	6.39	5.92	**4.74**	0.52	**0.59**	0.53	0.49	0.56
Inst Avg	7.94	6.18	7.65	**5.68**	6.02	0.50	**0.56**	0.51	0.55	0.55
Cat Avg	6.27	5.37	6.91	5.20	**5.06**	0.57	**0.63**	0.58	0.61	0.62

E Additional Ablation Studies

Runtime efficiency. We evaluate runtime efficiency in Table 4. Times are measured for each method for a single shape prediction (running with batch size of 1), averaged over 20 samples. Here, *Ours (M^3 only)* denotes our approach with only single-resolution M^3 priors.

Table 4: Quantitative comparison for testing time efficiency (s).

3D-EPN	Few-Shot	IF-Nets	AutoSDF	Ours (4^3 only)	Ours (8^3 only)	Ours (32^3 only)	Ours
0.015	0.004	0.421	0.958	0.025	0.017	0.016	0.063

What is the impact of the number of priors? We evaluate the effect of different numbers of priors on ShapeNet data in Table 5 (with 50% priors and 150% priors). We see that performance degrades with 50% priors, while further increasing the prior number reaches a performance plateau (and requiring additional storage). In our approach, our prior storage takes 14.68 MB in memory.

Table 5: Ablation on the number of shape priors on ShapeNet [2].

	Inst-CD↓	Cat-CD↓	Inst-IoU↑	Cat-IoU↑
Ours (50% priors)	4.41	4.45	0.632	0.640
Ours	4.23	**4.27**	**0.644**	**0.654**
Ours (150% priors)	**4.22**	4.30	0.638	0.647

What is the effect of different multi-resolution combinations? We considered patch resolutions of 4^3, 8^3, 16^3, and 32^3. We found 16^3 and 8^3 to perform very similarly (variance of $8e^{-6}$ IoU and $6e^{-5}$ CD), and used 8^3 to potentially resolve more detailed patches.

We evaluate alternative multi-resolution combinations in Table 6, which shows that all resolutions benefit the more detailed chamfer evaluation (whereas IoU only penalizes non-intersections, rather than how far the predictions are from the GT object).

Table 6: Ablation study of multi-resolution combinations on synthetic ShapeNet [2].

	Inst-CD↓	Cat-CD↓	Inst-IoU↑	Cat-IoU↑
Ours (4^3 with 32^3)	4.30	4.35	0.642	0.651
Ours (4^3 with 8^3)	4.35	4.42	**0.644**	**0.654**
Ours (all resolutions)	**4.23**	**4.27**	**0.644**	**0.654**

What is the impact of the concatenation in Eq. 3. We evaluate the effectiveness of concatenation in Eq. 3 in the main paper in Table 7, considering the attention-based term only (the core of our approach). We note that when excluding the attention-based term, this does not consider local patches anymore and becomes similar to the encoder-decoder training of 3D-EPN. As the attention-based learning of correspondence to local priors is the core of our approach, this produces the most relative benefit, with a slight improvement when combining the terms together.

Table 7: Concatenation ablation study for each term in Eq. 3 on the ShapeNet [2].

	Inst-CD↓	Cat-CD↓	Inst-IoU↑	Cat-IoU↑
3D-EPN	5.48	5.58	0.582	0.594
Ours (attention term only)	4.25	4.29	0.640	0.650
Ours	**4.23**	**4.27**	**0.644**	**0.654**

Ablation for fixed priors, no pre-training, and no attention on 4^3 priors only. We evaluate this scenario as the lower bound for our task in Table 8, which produces significantly worse results due to the lack of learnable priors in combination with attention.

Table 8: Evaluation for fixed priors, no pre-training, and no-attention on 4^3 priors only on ScanNet [5].

	Inst-CD↓	Cat-CD↓	Inst-IoU↑	Cat-IoU↑
Ours (fixed priors, no pre-training, and no-attention on 4^3 priors only)	9.53	9.73	0.35	0.37
Ours	**7.38**	**7.52**	**0.50**	**0.50**

F Model Architecture Details

Figure 1 details our model architecture. Figure 1 (a), (b) and (c) respectively present the submodule for learning patch priors at resolutions 32^3, 8^3, and 4^3. The network in Figure 1 (d) shows our multi-resolution patching learning stage. Inputs are partial scans and the learnable shape priors, and the outputs are completed shapes. The specifications of encoder and decoder blocks in these models are shown in Figure 2.

G Additional Qualitative Results

Figure 3 shows more examples for qualitative results on ShapeNet, and Figure 4 shows more examples for qualitative results on ScanNet scans.

References

[1] Avetisyan, A., Dahnert, M., Dai, A., Savva, M., Chang, A.X., Nießner, M.: Scan2cad: Learning cad model alignment in rgb-d scans (2018) 1

[2] Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F.: Shapenet: An information-rich 3d model repository (2015) 1, 3, 4, 5, 8

[3] Chibane, J., Alldieck, T., Pons-Moll, G.: Implicit functions in feature space for 3d shape reconstruction and completion (2020) 2, 3, 4

[4] Curless, B., Levoy, M.: A volumetric method for building complex models from range images. In: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques. pp. 303–312 (1996) 1

[5] Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes (2017) 1, 2, 3, 4, 5, 9

[6] Dai, A., Qi, C.R., Nießner, M.: Shape completion using 3d-encoder-predictor cnns and shape synthesis (2017) 2, 3, 4

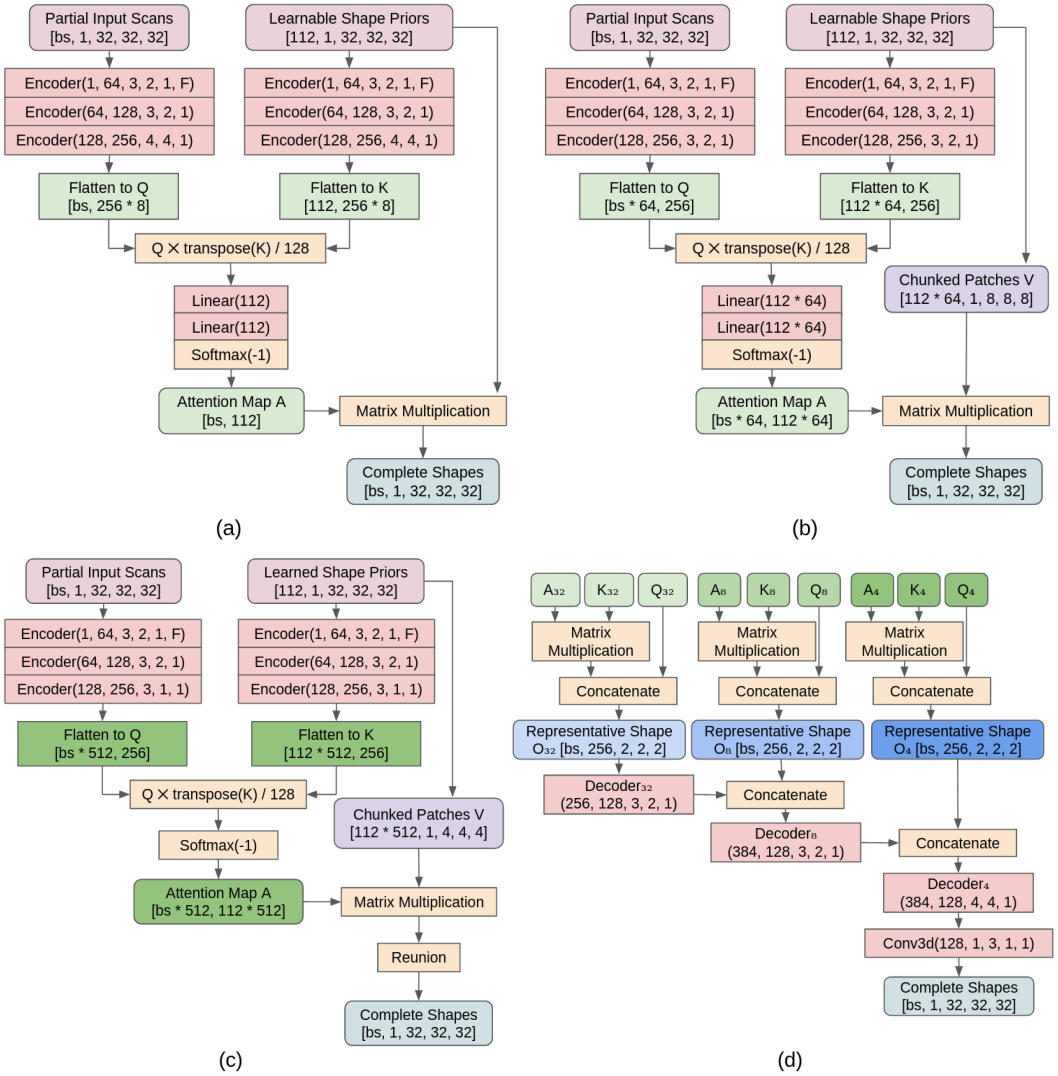

Figure 1: Model specifications in our method. (a) represents the patch learning model structure for resolution at 32^3; (b) represents the patch learning model structure for resolution at 8^3; (c) represents the patch learning model structure for resolution at 4^3; (d) represents the multi-resolution model structure. In figure (d), A_i represents the obtained attention map, Q_i represents the input local features, and K_i represents the learned prior patch features, where i is the resolution for patch learning model, $i = 32, 8, 4$.

[7] Mittal, P., Cheng, Y.C., Singh, M., Tulsiani, S.: AutoSDF: Shape priors for 3d completion, reconstruction and generation. In: CVPR (2022) 2, 3, 4

[8] Nie, Y., Lin, Y., Han, X., Guo, S., Chang, J., Cui, S., Zhang, J.: Skeleton-bridged point completion: From global inference to local adjustment. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 16119–16130. Curran Associates, Inc. (2020) 1

[9] Wallace, B., Hariharan, B.: Few-shot generalization for single-image 3d reconstruction via priors (2019) 2, 3, 4

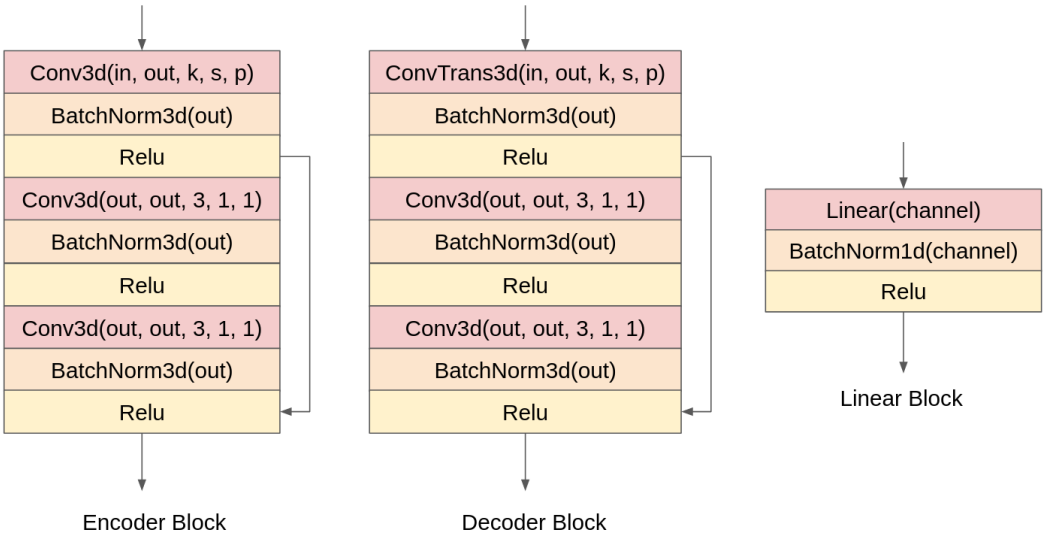

Figure 2: Layer specifications in our model. During the process of learning patch priors in a single resolution, we use encoder blocks to encode partial input scans and learnable shape priors to local features, and then use linear blocks to post-process the obtained attention map. The decoder block is used for decoding complete shapes in a multi-resolution patch learning module.

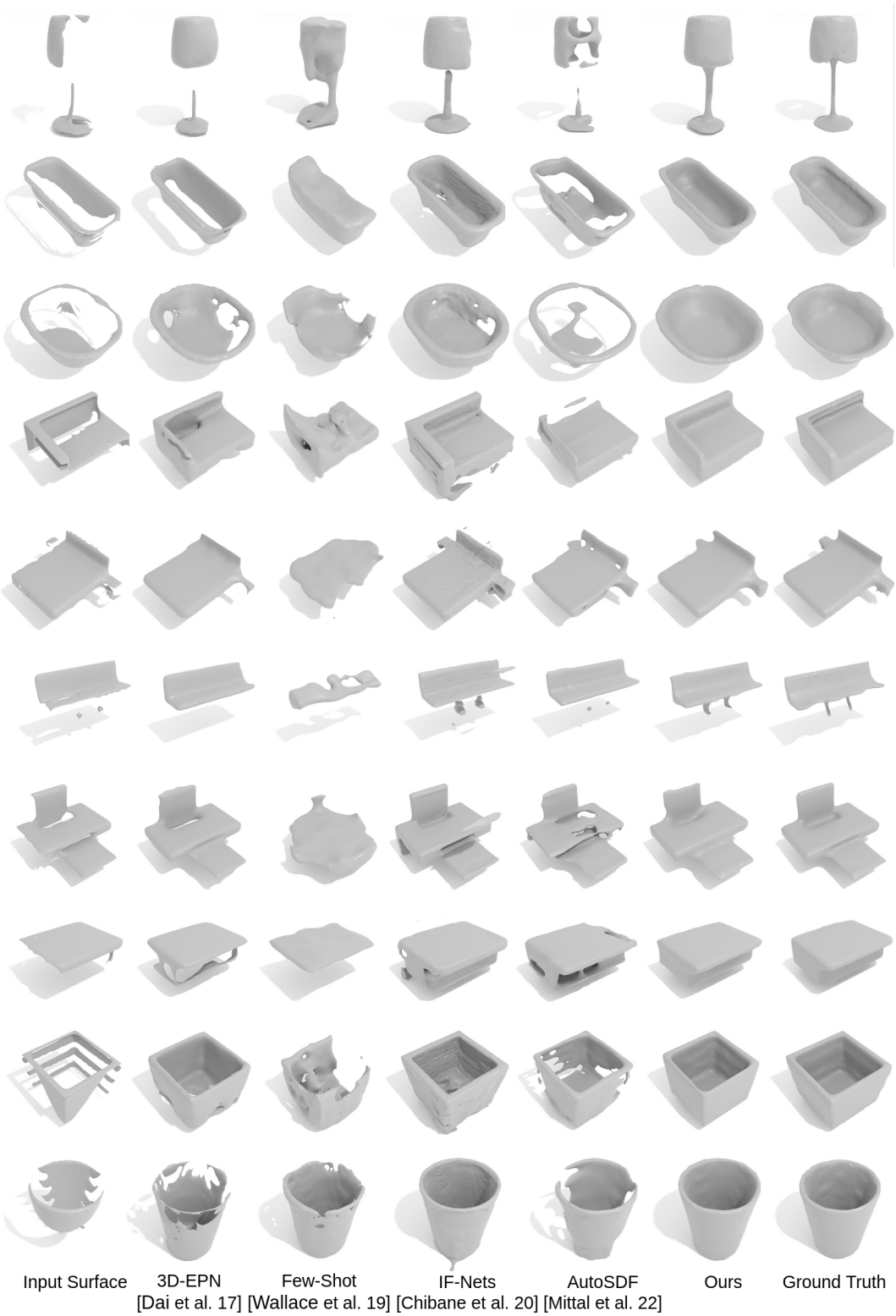

Figure 3: Qualitative comparisons with state of the art on ShapeNet [2].

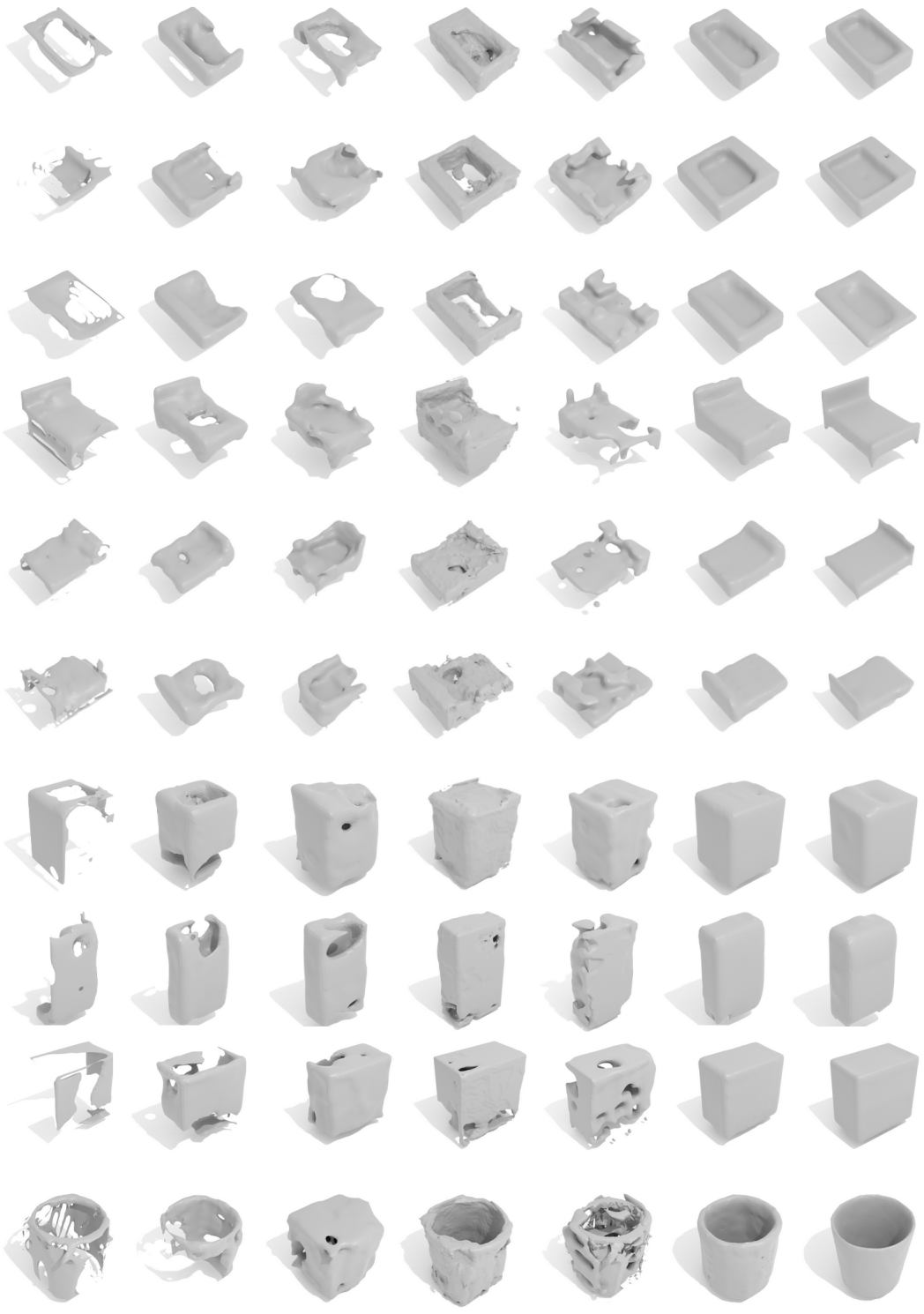

Figure 4: Qualitative comparisons with state of the art on ScanNet [5].