3D CoMPaT: Composition of Materials on Parts of 3D Things Supplementary Materials

Yuchen Li[1,*], Ujjwal Upadhyay[1,⋆], Habib Slim[1,*], Ahmed Abdelreheem[1], Arpit Prajapati[2], Suhail Pothigara[2], Peter Wonka[1] Mohamed Elhoseiny[1]

[1] KAUST, Thuwal, Saudi Arabia: {firstname.lastname}@kaust.edu.sa
[2] Polynine, San Francisco, California: {firstname}@polynine.com

These supplementary materials include the following:

- A description of the web browser for a sample of the stylized models in the 3D CoMPaT dataset; see Sec. 1.
- Additional dataset details; see Sec. 2.
- Amazon Mechanical Turk 3D Models verification; see Sec. 3.
- Material tagging multi-label classification; see Sec 4
- Implementation details for the adapted BPNet [3] architecture for 3D-GCR-SEG; see Sec. 5.
- More results for the adapted BPNet [3] architecture for 3D-GCR-SEG including experiments with an additional 3D shape classifier; see Sec. 6.
- Details for the adapted PointGroup [4] architecture for 3D-GCR-SEG; see Sec. 7.

⋆ co-first authors

1 3D CoMPaT Web Browser

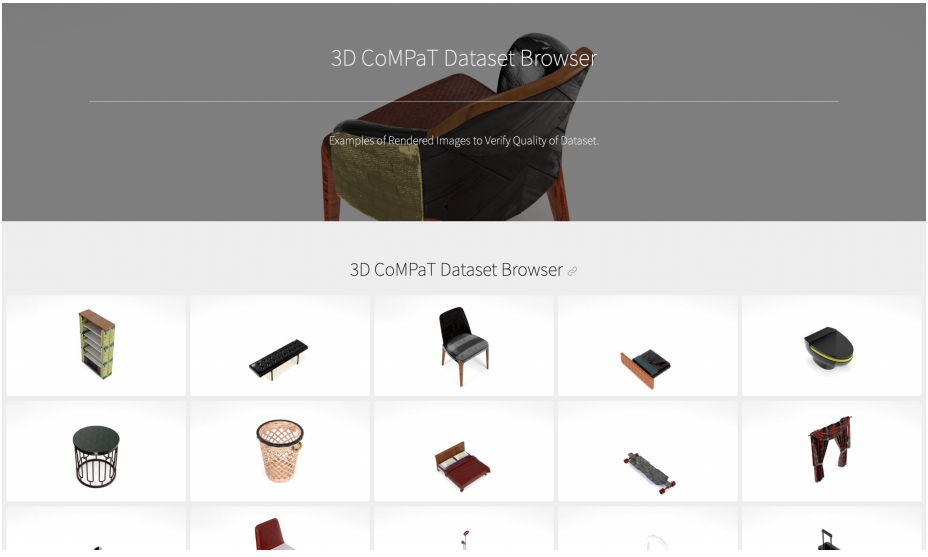

Fig. 1: 3D CoMPaT dataset sample web browser.

We provide a web browser for a sample of our proposed dataset 3D CoMPaT at https://3dcompatbrowser.github.io/gallery/. We currently only show images of randomly selected compositions. By showing many such renderings of 3D shapes in different categories, the main goal is to demonstrate the average quality of compositions. The images show randomly selected stylized objects. In Figure 1, we provide a screenshot of the webpage.

2 Additional Dataset Details

The dataset contains 7320 distinct models spanning 43 semantic classes. We have collected the dataset in a way that ensured that top 10 categories have at least 200 models each. The top 10 categories have been chosen based on the possible number of parts that can be annotated in their models, and the shape diversity of models in these categories. The number of instances of top model classes and top parts are reported in Table 1 and 2 respectively. Note that not all parts are reported in Table 2 to keep the table readable.

Table 1: Shape classes and their number of unique shape instances.

Model Name	Count	Model Name	Count	Model Name	Count	Model Name	Count
table	1174	**ottoman**	199	**shower**	46	**tray**	40
chair	1140	**toilet**	158	**gazebos**	46	**sun loungers**	40
lamp	502	**faucets**	139	**fans**	45	**ladder**	40
vase	444	**sinks**	82	**clock**	45	**bird house**	40
cabinet	417	**love seat**	67	**parasol**	44	**bicycle**	40
bed	385	**bags**	61	**car**	44	**basket**	40
sofa	381	**curtains**	58	**shelf**	43	**coat racks**	40
stool	305	**candle holder**	57	**dishwasher**	43	**airplane**	40
bench	297	**skateboard**	47	**trolley**	42	**sports table**	17
planter	244	**bbq grill**	19	**jug**	41	**rug**	6
dresser	201	**boat**	46	**garbage bin**	41		

Table 2: Part classes and their number of instances.

Part Name	Count	Part Name	Count	Part Name	Count	Part Name	Count
leg	4404	**head**	278	**wall_mount**	57	**finial**	40
base	3761	**footrest**	234	**faucet**	57	**fin**	40
support	2627	**bedskirt**	233	**bottom**	55	**seat_stay**	40
seat	2567	**pole**	233	**tank_cover**	55	**saddle**	40
back	1519	**rod**	214	**knob**	52	**fork**	40
top	1170	**wheel**	191	**side_panel**	52	**front_side_rail**	40
bush	1082	**arm**	171	**step**	51	**vertical_divider_panel**	40
handle	1051	**bowl**	158	**cap**	49	**spokes**	39
armrest	974	**tabletop_frame**	158	**deck**	49	**button**	39
cushion	702	**seat_cover**	156	**adjuster**	48	**side_walls**	38
container	682	**seat_cushion**	155	**hook**	47	**hanger**	38
containing_things	676	**wire**	151	**truck**	47	**short_ribs**	38
top_panel	650	**spout**	140	**grip_tape**	47	**grill**	38
back_panel	641	**aerator**	117	**slab**	46	**wiper**	38
vertical_side_panel	631	**shade_cloth**	108	**bracket**	46	**chain_stay**	37
bottom_panel	603	**frame**	101	**shower_head**	46	**long_ribs**	37
body	534	**nozzle**	87	**candle**	46	**lid**	36
door	519	**canopy**	86	**blade**	45	**runner**	36
pillow	515	**glass**	85	**downrod**	45	**motor_box**	44
drawer	505	**cabinet**	83	**hour_hand**	45	**rod_bracket**	44
bulb	498	**sink**	82	**shaft**	44	**floor**	43
backrest	457	**windshield**	82	**trunk**	44	**rear_view_mirror**	43
design	438	**drain**	81	**doors**	44	**pedal**	43
mattress	377	**roof**	78	**hood**	44	**stem**	41
headboard	347	**stand**	71	**minute_hand**	44	**headlight**	41
socket	332	**water_tank**	64	**pillars**	44	**side_windows**	40
lamp_surrounding_frame	331	**bag_body**	61	**dial**	40	**tyre**	40
shelf	323	**fabric_design**	58	**rear_window**	40	**dial**	40
duvet	285	**flush_push_button**	58	**wing**	40	**rear_window**	40

3 Amazon Mechanical Turk AMT 3D Models Verification

In this section, we discuss the AMT experiment conducted in order to verify the manually annotated model and part names. We ask five MTurk participants to choose from the following four options per 3D Model/Part:

1. "yes, I would name the model the same"
2. "yes, but I would have given the model a different name"
3. "no, this is a wrong name (please specify a name)"
4. "no, the model cannot be given a specific name"

We provide instructions for these tasks to workers as shown in Figure 2.

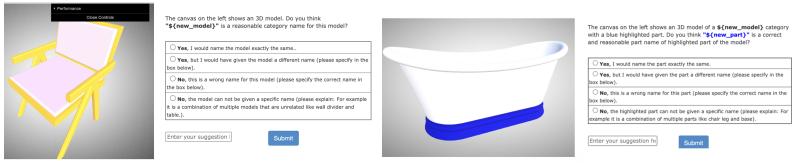

Fig. 2: Amazon MTurk user interface used to verify all model names (left) and part names (right). We allow users to choose between four options (5 participants per example).

Given the scale of the 3D CoMPaT dataset, manually checking all models is not feasible. A subset of 2000 models was thus selected for verification. Each of these models was inspected by at least five reviewers for part names and model classes. Regardless of their choice of option, they were encouraged to suggest an alternative name for parts and model class. These suggestions were used intensively to appropriately name certain parts despite heavy support for the previous annotation. This process not only made the part and model class more aligned with generally used vocabulary but also helped remove confusing instances of certain data points.

In Figures 3, 4 and 5 is the distribution of options picked by participants for the two different tasks.

From Figure 3, it is evident that most of our manually annotated parts and model names were correct. The reviewers agreed with most of the annotations and taking their suggestions for alternative names into account increased the robustness of the dataset from a natural language perspective.

Figure 4 shows that there was consensus on earlier annotations for nearly all of the top 10 models. Some suggestions were made for model classes such as bowl and table. This can be attributed to table being a superclass grouping a much wider variety of tables like side tables, office desks, etc.

Figure 5 shows the experiment for part names resulted in much more diverse annotations being suggested by reviewers. The presence of substantial "Yes2"

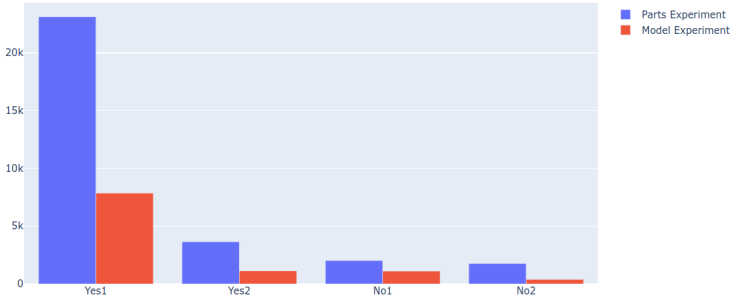

Fig. 3: Vote distribution for part and model names.

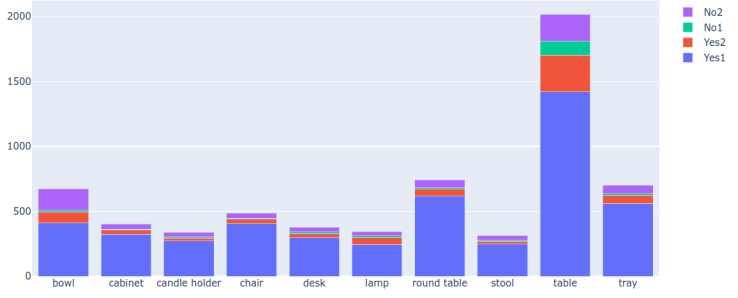

Fig. 4: Vote distribution for 10 randomly selected model classes.

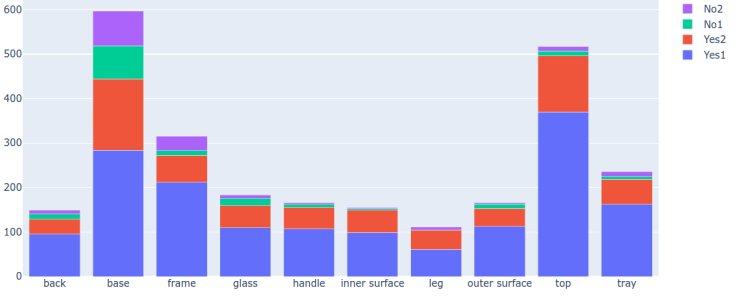

Fig. 5: Vote distribution for 10 randomly selected part classes.

votes indicates that although reviewers agreed with proposed annotations, most of them preferred an alternative name. As an example, the "base" part name appears to be the most disagreed upon name among all top 10 parts. This can be attributed to it vagueness, but reviewers do not reject it nonetheless because of the complexity of naming unusual parts in some models.

4 Material tagging multi-label classification

In Table 3, we give test results for a ResNet50 model trained for the multi-label material classification task given a single rendered view. The results show that the front views (1 & 3) have slightly better performance than the back views (2 & 4).

Table 3: F1-score and Average Precision of a ResNet50 model trained on our proposed dataset 3D CoMPaT for material tagging, evaluated on each of the four views separately. Results show that the model performs slightly better on front views.

	View 1 (Front Left)	View 2 (Back Right)	View 3 (Front Right)	View 4 (Back Left)	Max Ensemble	Avg Ensemble
F1-score	**64.3%**	63.8%	63.9%	62.2%	**71.76%**	65.00%
Avg. Precision	**72.7%**	69.8%	71.2%	68.1%	78.65%	**80.65%**

The task in this experiment is to correctly label all materials appearing in a given rendered view. This experiment allows us to further understand how well a CNN encoder can identify materials. We use a ResNet50 [2] backbone for encoding the image, which makes predictions selecting among all 193 possible materials. We used the F1 score with a threshold of 0.5 and the Average Precision as metrics for this baseline. In Table 3, we show the test scores of this baseline on our proposed 3D CoMPaT dataset. From Table 3, it is evident that the model performed better on view 1 and view 3 compared to the back views, view 2 and view 4. This difference can be explained by the changing distribution of parts across views and the design of certain frequently occurring models like table, cabinet, etc. These frequently occurring models are often found to have only "back" as a part. The diversity for material assignment is lesser for these views which partly explains why the model didn't perform on par with front view renderings.

We also proposed **max ensemble** and **average ensemble** methods to overcome missing materials in the renderings, due to parts overlapped during photography from the different viewpoints. We aggregate by average (or maximum) the 4 views encodings in the last layers of a ResNet50, then compute the corresponding F1 score and Average Precision. We found that using the ensemble methods is more reasonable and effective, since there may be some missing parts in a specific view of a model; see Table 3.

Notably, there are other tasks that can be performed by leveraging our dataset beyond the experimental scope of this paper, including part retrieval from a given set of styled models from various views, material segmentation, segmenting materials from rendered views of a styled model. All these annotations are part of the design of our data collection procedure. Our proposed dataset enables a wide range of applications like image-based 3D shape retrieval, 3D shape reconstruction from single/multiple images, 3D object classification, 3D semantic segmentation, and 3D shape generation.

5 Implementation details for the adapted BPNet [3] architecture for 3D-GCR-SEG

We provide here implementation details for our adapted BPNet [3] architecture. The input of our network is a non-stylized 3D model and stylized 2D images. The outputs are part segmentation maps, material segmentation maps and shape classification result. We train the network end-to-end by minimizing the cross entropy loss using momentum SGD with a polynomial learning rate decaying from 1e-2. The model was trained on 4 V100 GPUs for 50 epochs. Each epoch takes an average of 3.83 hours with a batch size of 16. Once we have the segmentation maps, we vote material over the part segmentation maps (i.e check which materials are predicted for the segmentation map of a part). We set the voxel size to 5cm for the benchmark results.

6 More results for the adapted BPNet [3] architecture for 3D-GCR-SEG including experiments with an additional 3D shape classifier

We compare BPNet with top methods on the 3D CoMPaT benchmark in Table 4, including 2D-only methods with UNet [5] and 3D-only methods with MinkowskiNet [1]. For evaluation metrics, we use mean classwise intersection over union (mIoU), mean of classwise accuracy (mAcc), and overall pixel-wise accuracy (OA). BPNet outperforms these other models ($\geq$ 2.33 mIoU in 3D and $\geq$ 2.85 OA in 2D), thanks to the bidirectional projection module (BPM). The module effectively interacts with 2D and 3D elements to integrate the advantages of these two vision fields. The result of 3D Grounded CoMPaT recognition (GCR-SEG) on BPNet with separate shape classifier (DGCNN) are given in Table 5. As seen in the following tables, results on top-1 and top-5 predicted shape were significantly improved.

Table 4: Comparison of BPNet with top 2D and 3D methods on 3D CoMPaT on 2D and 3D segmentation tasks.

Method	2D			3D		
	mIoU	mAcc	OA	mIoU	mAcc	OA
BPNet	21.80	**33.44**	**96.36**	**2.73**	**4.16**	**84.86**
UNet	26.37	33.42	93.51	-	-	-
MinkowskiNet	-	-	-	0.40	0.90	18.91

Table 5: 3D Ground CoMPaT recognition GCR-SEG result on BPNet [3] with a separate shape classifier (DGCNN [6]).

	Top-1 predicted shape					Top-5 predicted shape					Ground Truth Shape			
	Shape Acc.	Value	Value-all	Value-grnd	Value-all grnd	Shape Acc.	Value	Value-all	Value-grnd	Value-all grnd	Value	Value-all	Value-grnd	Value-all grnd
Standard	71.1	42.33	11.23	5.78	2.55	82.7	51.21	13.62	6.73	0.28	64.61	16.78	8.14	0.37
GT Material	71.1	42.67	11.55	5.86	0.92	82.7	51.29	13.68	9.67	1.12	64.94	16.83	12.17	1.4
GT Part	71.1	49.75	37.21	17.65	9.68	82.7	64.51	44.08	20.28	11.26	76.73	53.31	25.31	14.01

7 Details for the adapted PointGroup [4] architecture for 3D-GCR-SEG

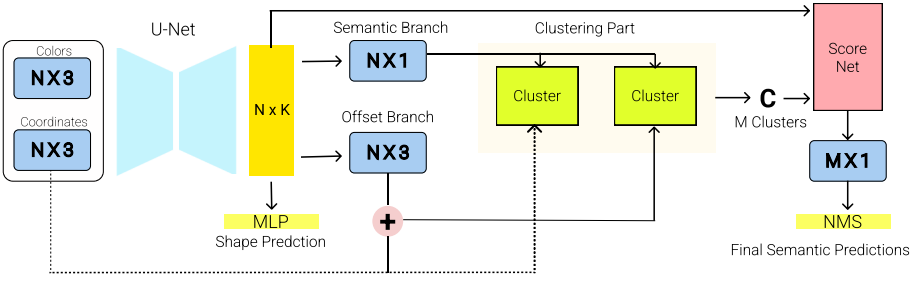

Fig. 6: Adapted architecture for semantic segmentation task on 3D CoMPaT dataset.

In Figure 6, we illustrate the small changes made to the original PointGroup [4] architecture. An MLP was simply added after the U-Net output features. This MLP was trained to predict model class. We train the network end-to-end by minimizing the cross-entropy loss using the momentum SGD optimizer, and a multi-step learning rate scheduler with a decay of 1e-2. The training was conducted on 4 V100 GPUs.

References

1. Choy, C., Gwak, J., Savarese, S.: 4d spatio-temporal convnets: Minkowski convolutional neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3075–3084 (2019) 7
2. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition (2015) 6
3. Hu, W., Zhao, H., Jiang, L., Jia, J., Wong, T.T.: Bidirectional projection network for cross dimension scene understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14373–14382 (2021) 1, 7, 8
4. Jiang, L., Zhao, H., Shi, S., Liu, S., Fu, C.W., Jia, J.: Pointgroup: Dual-set point grouping for 3d instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4867–4876 (2020) 1, 8
5. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation (2015) 7
6. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. ACM Transactions on Graphics (TOG) (2019) 8