

ExCap3D: Expressive 3D Scene Understanding via Object Captioning with Varying Detail

Supplementary Material

7. Additional Results

Modeling objects as a sum of parts improves caption quality ExCap3D models an object as a sum of its parts by conditioning the object captioner on the outputs of the part captioner. Alternately, we can model top-down with the parts being components of a whole, by first generating object descriptions and conditioning the part captioner on the corresponding hidden states. Results of this reversed information flow are shown in Tab. 4. Other consistency losses are kept the same in both experiments. Part-level details do not benefit from object \rightarrow part information sharing as object-level captions do not contain fine-grained part information, while part captions contain information that is useful for describing the whole object.

Method	Object-Level	Part-level
Object \rightarrow Part Info. Sharing	32.8	15.2
Part \rightarrow Object Info. Sharing	32.9	32.8

Table 4. Comparison of different directions of information sharing between the object and part captioners. Performance is reported in CIDEr@0.5.

Fine-grained context features improve part captioning

ExCap3D uses segment-level object context features for captioning. This is required particularly for describing low-level part details of smaller regions of objects, in addition to the caption-aware query features $Q_{c,o}$ which contain coarse information about the whole object. Results are shown in Tab. 5. Object-level details are sufficiently captured by the query features, while part details benefit from the fine-grained per-segment context features.

Method	Object-Level	Part-level
w/o context features	33.7	25.5
w/ context features	32.9	32.8

Table 5. Comparison of our model with and without object context features. Performance is reported in CIDEr@0.5.

End-to-end learned captions improve upon separate seg-

mentation and VLM models ExCap3D combines the tasks of instance segmentation and object captioning. Instead, we can first predict instance masks using Mask3D

and describe them with the VLM by projecting them onto the multiview DSLR images. However, these instance predictions are not as accurate as the GT instances and can limit the captioning performance of the VLM which expects precise object crops. Alternately, we can render the 3D mesh at the DSLR image camera poses and use these as input to the VLM. Since VLMs are primarily trained on natural images, this is expected to give low quality captions. In both cases, the VLM may produce slight inconsistencies at the different levels of detail. Results are shown in Tab. 6. ExCap3D’s end-to-end and learned captioning approach outperforms the VLM applied on DSLR images or rendered images of meshes.

Method	Obj	Part
Pred. Inst. + VLM on rendered mesh	18.6	11.4
Pred. Inst. + VLM on DSLR images	21.4	15.5
End-to-end captioning	32.9	32.8

Table 6. Comparison of our model with the VLM applied on predicted 3D instances, projected to multiview DSLR images and multiview 3D mesh renders. Performance is reported in CIDEr@0.5 at object- (Obj) and part-levels (Part).

Qualitative results We show additional qualitative results from ExCap3D in Fig. 6. ExCap3D makes use of fine-grained 3D features on predicted instances to predict expressive object and part properties such as material, texture and appearance.

8. Implementation Details

Instance segmentation For all baselines and ExCap3D, we pretrain the respective detection or segmentation backbones on ScanNet [18]. For voxel-based methods (D3Net, PQ3D and ExCap3D), we use a voxel size of 2cm. We use a graph-cut based oversegmentation [23] to precompute segments for PQ3D and ExCap3D. We use only 3D scan inputs for all methods.

For ExCap3D, we use the default configuration of Mask3D for instance segmentation training, with 100 instance queries and an embedding dimension of 128. Since ScanNet++ contains several large and complex scenes, we randomly sample 300k points for training on such scenes, while evaluation is carried out on all points. Training takes about 4 days on an Nvidia A6000. The instance segmen-

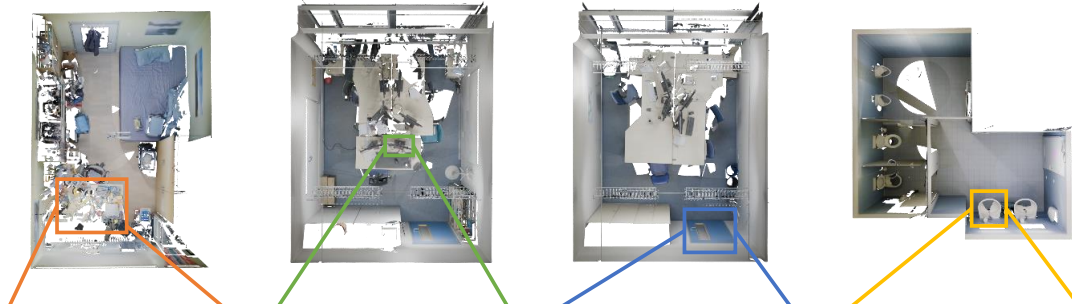

Input 3D Scene					
					
Detected 3D Object					
ExCap3D (Ours)	Object-level	Rectangular table made of light-colored wood or wood-like material with smooth texture and visible grain pattern.	Flat-screen monitor with black color, smooth texture, and rectangular shape.	Blue, smooth, rectangular door made of wood or wood-like composite material.	White, rectangular, porcelain sink with a smooth, glossy surface.
	Part-level	A rectangular table with a light-colored, smooth surface, possibly made of wood or laminate, has a cluttered surface with various items.	A flat, rectangular screen with a glossy finish, black color, and smooth texture, mounted on a black, rectangular stand with a flat base and tapered base.	The door is predominantly blue with a smooth texture, made of a light-colored material, possibly wood or a wood-like composite, with a metallic handle on the right side.	A white, porcelain or ceramic sink with a smooth, glossy surface, a metallic faucet with a lever handle and lever handle.
Ground Truth	Object-level	Dark brown, wooden, rectangular desk with visible grain texture.	Black, glossy, rectangular monitor with rounded corners.	Blue, smooth, metallic, rectangular door with a grid-patterned window and a silver metal handle.	White, oval-shaped, ceramic or porcelain sink with flat bottom and slightly narrowing front.
	Part-level	A wooden desk with a curved front edge, smooth texture, and a cluttered surface with various items in shades of orange, black, white, yellow, and brown.	A dark gray, matte-finish screen with a black, glossy bezel, attached to a black, smooth-textured stand with a tapered base.	A wooden door frame with a light brown color and smooth texture, attached to a wall. A blue door with a smooth texture, a frosted glass panel, and a silver handle on the right side.	A white, porcelain or ceramic basin with a smooth, glossy surface and a curved shape, flat bottom, and circular drain hole.

Figure 6. Object- and part-level captions predicted by ExCap3D on a diverse set of semantic classes and indoor scenes. ExCap3D accurately describes both part-level properties of objects such as material and texture, as well as object-level properties.

tation model has an AP50 score of 0.38 on the ScanNet++ validation set with 84 instance classes.

Multilevel captioning The instance segmentation model is kept frozen during caption training. During training we sample from the different captions for an object, and during inference we use a single fixed caption as the GT. Since all the objects in a training batch may not have captions after being filtered out by visibility constraints, $\mathcal{L}_{\text{caption}}$ is applied only on the objects that have captions. During inference, captions are produced for all 100 input queries

and matched with GT objects that have captions. For the object- and part-level captioners Ψ_{obj} and Ψ_{part} we use a language model with the GPT2 architecture trained from scratch, with an embedding size of 128, 1 layer and 4 attention heads.

For the consistency losses, caption projection models Φ_{obj} and Φ_{part} we use transformer encoders with 2 layers each, an embedding dimension of 16, feedforward dimension of 128 and 2 attention heads. The loss weights are determined as $w_1 = 1, w_2 = w_3 = 0.1$ empirically. During inference we use beam search with 5 beams, and pick

hidden states from the beams corresponding to the finally predicted caption.

Other model details ExCap3D has 53M parameters, including Mask3D (39.6M), object- and part-level captioners (6.7M each), embedding classifiers (8.9K each) and projectors (12.9K each). Inference takes 4.61s per scene on an Nvidia A6000 using 17 GB GPU memory and 6 GB RAM.

9. Implementation Details of Data Generation

Part segmentation pseudo-mask generation To generate 2D masks with SAM [33], we prompt it with points on the DSLR images. These points are sampled from the corresponding 3D precomputed segments on the mesh [23] to encourage consistency between masks from different views. Then we use MaskClustering [51] to backproject and combine the 2D masks on the 3D mesh and keep the smaller mask when two masks overlap. We subsample the densely captured DSLR images by a factor of 5 which resulted in sufficient multiview consensus.

Caption dataset generation Since ScanNet++ contains a dense capture trajectory of DSLR images, we subsample every 10th DSLR frame from the train and val splits. In addition to filtering the DSLR images for visibility of the object, we filter out bounding boxes with a dimension of less than 50 pixels to avoid very small inputs to the VLM. To avoid the VLM describing the distortion in the fisheye DSLR capture of ScanNet++ (e.g., *this is a distorted image of a table*) and match the training distribution of the VLM, we undistort the images using the provided distortion parameters before using them as input to the VLM. As the generated captions in the ExCap3D Dataset are relatively long and detailed, we simplify the captions for training to a shorter length using an LLM, for training and evaluation. During part-caption generation, in addition to a cropped image of the part, we generate a second caption based on a context image that includes the whole object, with the part indicated by a red bounding box. This improves caption quality when the crop of the part is small.

VLM prompts for caption generation We prompt the VLM with multiview crops of each object, along with its semantic class name and 3D dimensions in meters. The VLM is prompted to generate the object’s shape, structure, color and texture using only information available in the image, without adding any common-sense information. For object part crops we provide the name of the object in the VLM’s input.

LLM prompts for multiview and part caption aggregation To aggregate over captions generated in multiple

views, we prompt the LLM with a list of these captions and to output the most likely and uniquely identifying information of the object/part, and remove any commonsense information about it. For object parts, we first aggregate over multiple views, and then over all the parts belonging to an object to get a single part-level caption for the object. We also experimented with aggregation of the object-level captions within each view and then over all views, and found the resulting captions to be of lower quality.

10. Captioning Metrics and Evaluation

We provide a brief overview of the different captioning metrics used.

ROUGE ROUGE is based on BLEU [41], which measures the overlap of n -grams in the predicted and GT captions. ROUGE adds additional precision and recall terms to BLEU. We use the F1 score of the ROUGE-L variant that measures matches of longest common subsequences, even if the matching words are not contiguous. ROUGE focuses on caption recall, and hence rewards long captions that capture as much content of the GT as possible. ROUGE ranges from 0-1.

METEOR METEOR [20] adds the notion of semantic matching of words between candidate predictions and GT based on a fixed database [40], and computes matches between chunks of words that need not be contiguous. However, it has a larger weight for n -gram recall compared to precision. METEOR ranges from 0-1.

CIDEr CIDEr [44] is a widely used metric for evaluating caption predictions against GT captions in the 2D and 3D domains. It has been shown to correlate highly with human judgement. The initial version of CIDEr [48] improved upon ROUGE and METEOR by incorporating a TF-IDF term over the whole GT corpus, and hence downweights n -grams that occur frequently in the GT. This increases the weight of terms that occur rarely in the GT, and occur in the prediction. In our case, these rare terms are the unique and identifying features of the objects. We use CIDEr-R [44], simply written as CIDEr, which adds additional length and repetition penalties on the n -grams. The final score is averaged over multiple values of n . CIDEr score ranges from 0-10, due to a normalizing factor of 10 to make it comparable with other metrics.

Display convention We report results of all three metrics after multiplying by 100, as done in other 3D dense captioning works [10, 15, 60].

Evaluation These metrics are averaged over every GT instance, with a weight of 1 if the IoU of the object's predicted bounding box with the ground truth bounding box is greater than 0.5, and 0 otherwise.