Dense Captioning for 3D Scenes with Sparse Convolutions and Reinforcement Learning

Daniel-Jordi Regenbrecht*
Technical University of Munich

Yifeng Dong*
Technical University of Munich

Abstract

The task of 3D dense captioning is a standing and challenging problem in the intersection of computer vision (CV) and natural language processing (NLP). Benefiting from recent advancements in both fields, we present a novel neural architecture with two major improvements: 1. We employ a new object detection and feature extraction backbone based on the recently presented sparse convolution (SC) and submanifold sparse convolution (SSC) operators. 2. We incorporate reinforcement learning (RL) methods for training, allowing us to optimize the language generation module directly for relevant NLP metrics instead of relying on a proxy loss. We find that both changes individually yield improvement over the baseline, and the largest improvement is achieved when using both together. We also find that finetuning a classically-trained model using RL can allow us to exploit the advantages of RL while avoiding some of its difficulties.

1. Introduction

Recently, significant research has been done on machine learning tasks in the intersection of visual understanding and natural language processing. In general, captioning tasks aim to provide a natural language description for a given visual input. In **dense captioning**, every object in the scene is identified, localized, and provided with a description. The task of **3D dense captioning** was introduced by Chen *et al*. in [5].

Understanding of 2D images is a well-studied area in the field of computer vision. Conversely, 3D understanding remains much more challenging, and 2D methods can often not be directly extended to 3D tasks: While treating an RGB-D point cloud as an unordered **set** necessitates a requirement for symmetry as an additional constraint, attempts to **voxelize** the data easily run into the *curse of dimensionality*, making the data much more voluminous and thus being processed inefficiently. [3, 9, 20, 22]. However, Jiang *et al*. [9] have recently brought great advancement to the latter approach by using SC and SSC to handle sparsity and achieved state-of-the-art results in 3D instance segmentation. Experiments in [5] show that 3D object detection performance is a significant bottleneck in current overall captioning performance.

Another challenge in generating descriptive captions is optimizing for natural language similarity between predictions and ground-truths instead of token-wise identity, since common NLP metrics like CIDEr [19] are not generally differentiable. Previous approaches [5] use token cross-entropy as a proxy loss; however the closely related field of image captioning has recently seen great success with using reinforcement learning techniques to face this challenge. [12, 15, 18].

Task The input for our 3D dense captioning task is an RGB-D point cloud. The expected output is a list of objects, each accompanied by a bounding box and a natural language description which describes both the object itself and its spatial relationship to other objects in the scene.

Dataset Following [5], we use the ScanRefer dataset [4], which contains 800 unique RGB-D scenes with a total of 11,046 localized objects and 51,583 accompanying descriptions. We also use the Scan2CAD dataset [1] for object orientation information, which we utilize for an auxiliary loss.

2. Related Work

3D Dense Captioning This work builds on the Scan2Cap pipeline [5], a modularized architecture which can be divided into three stages: 1) object detection; 2) extraction of object relationship features using a graph architecture and; 3) caption generation per object using the previous features.

3D Object Detection Current attempts in this field can be categorized into two approaches: architectures that directly consume a set of points [14, 20, 22] and those which first canonicalize the data by *e.g*. voxelization. PointGroup [9] falls into the latter category. Apart from using SC and SSC, it employs a two-branch clustering model that simultaneously uses the original point set and one with points shifted to their instance centroid, allowing for better local feature detection while keeping global information induced by object distances.

*Equal contribution.

Reinforcement Learning for Captioning Applying RL techniques to NLP tasks such as machine translation [10] has been the subject of much recent research [13, 15, 23]. REINFORCE [21] has provided a framework for optimizing the expected reward, in this case NLP-metrics, that has been successfully used in [12,15,18]. Further, Liu *et al*. [12] have developed SPIDEr, a linear combination of CIDEr and SPICE, as a metric that can be used in computing the rewards and have shown that it correlates well with human judgment.

3. Method

We build on the Scan2Cap [5] architecture and make two major contributions. First, for the object detection backbone, we shift away from the point-set method PointNet++ [14] and instead base the new detection and feature extraction backbone on PointGroup [9], which works on voxelized data and draws heavily on the novel SC and SSC operators. Second, we incorporate reinforcement learning as an added training method, inspired by recent success in reinforcement learning for image captioning.

3.1. Object detection backbone

Sparse Convolutions Convolutional layers have proven to be a major success in 2D vision tasks. However, they tend to be too computationally unwiedly to be effective on 3D inputs. At the same time, unlike 2D images, 3D inputs such as point clouds tend to be sparse by nature. **Sparse convolutions** are mathematically equivalent to classical convolutions, but calculated using sparse data structures, boosting computational efficiency. However, convolutional layers do not generally preserve sparsity, since an output pixel will become active if any of the input pixels in its receptive field are active. Therefore, the benefits of sparse computational are quickly negated when multiple layers are added. **Submanifold** sparse convolutions mitigate this problem by specifying that an output pixel is only active if the input pixel placed at the *center* of the kernel is active (see Fig. 1b). Thus, sparsity is always preserved. SSCs therefore allow us to utilize the benefits of convolutional layers on 3D scan data in a computationally feasible way.

In our model, we use a U-Net architecture of SSCs as a feature extractor, following [9]. However, to accomodate our purpose of using the features for captioning, we modify the architecture to be asymmetric, such that the number of output channels is much larger than the number of input channels. The point cloud input is voxelized before feeding into the U-Net and devoxelized afterwards.

We again follow [9] and use a simple MLP to predict an offset vector from each point to its object centroid. Then, a clustering algorithm is performed on both the original and shifted point cloud, generating object proposals. A separate scoring module assigns an objectness score to each proposal based on its aggregated features.

3.2. Relational Graph

We follow [5] and use their graph architecture for object feature enhancement and object relationship feature extraction. The graph module treats the object proposals as nodes and the relationship between objects as edges; features are learned via message passing. We refer the reader to [5] for a more in-depth description of this module.

3.3. Captioning Module

We base the architecture of the captioning module on [5], making the necessary changes to train it using reinforcement learning. At time step t of caption generation, the **Fusion GRU** produces the hidden state h_t^1 from the enriched proposal features, the GRU hidden state from time step $t-1$ and the GloVE embeddings of the previous token. The **Attention Module** takes the enhanced object and object relation features and returns the aggregated context vector $\hat{v}_t$. The state h_{t-1}^1 and context vector $\hat{v}_t$ are processed using an MLP to the feature u_t^2, which is fed into the **Language GRU** which delivers the hidden state h_t^2. Another MLP is used to predict token y_t. Differing from [5], we add a softmax layer to the output of this MLP to be able to interpret the output as per-token probabilities, which are needed for reinforcement learning.

3.4. Reinforcement Learning

Training using Policy Gradient The slight modification in the implementation allows us to interpret the captioning as a policy in line with the REINFORCE [21] framework and thus optimize directly for non-differentiable NLP metrics.

During training, instead of choosing the word with the highest predicted probability, we interpret the output of the final MLP in the language GRU as a probability distribution $\pi_\theta(a|s_t)$ over the words, parameterized by the network parameters θ and depending on the state s_t at time step t. This distribution is the stochastic policy, from which an action a_t, corresponding to one word in the vocabulary, is sampled at each step. After a predefined maximum description length, the generated captions are used to compute a reward $r(a_t)$ for each action. The goal is to maximize the expected reward or equivalently minimize the loss

$$L(\theta) := -\mathbb{E}_{a_t \sim \pi_\theta}[r(a_t)].$$

As shown in [21], an unbiased estimator of its gradient is provided by

$$\nabla_\theta L(\theta) \approx -r(a_t) \nabla_\theta \log p_\theta(w_t).$$

We can therefore use this estimator with a standard gradient descent optimizer such as Adam [11] to optimize the model directly for the maximum expected reward.

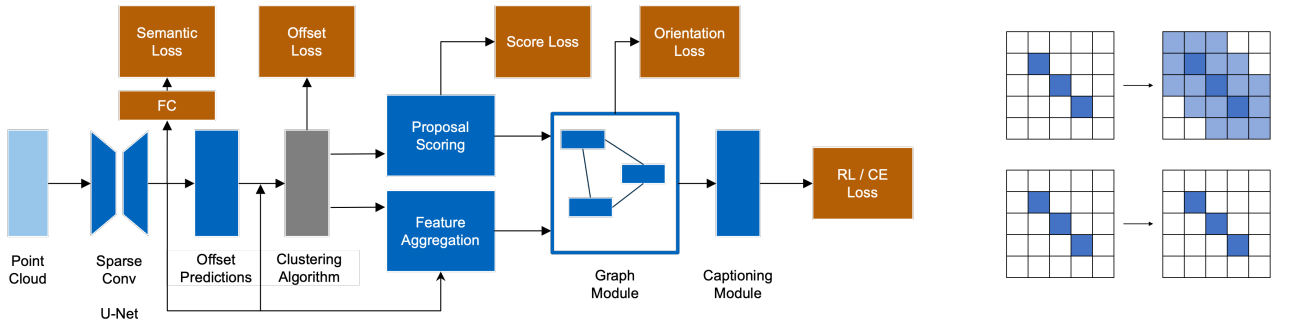


Figure 1. (a) General architectural overview of our model. (b) Effect of 3x3 convolution a sparse input when using classical (sparse) convolutions (top) vs. submanifold sparse convolutions (bottom). Diagram shows active (nonzero) areas.

Stabilizing Gradient Estimations The estimated gradients typically have a high variance, making learning more challenging [7]. Two choices we made in the design of the training process help mitigate this problem: 1. The gradients are calculated for all of the objects in a scene as a batch. In practice, this means that gradients are averaged over a high number of actions. 2. We subtract a baseline from the reward. The latter has been shown to reduce variance while still providing an unbiased gradient estimate [18]. We choose an exponentially decaying average of preceding rewards as the baseline. More sophisticated baselines have been successfully used previously [12, 15], though this is not the main focus of this work.

Rewards We follow the work of [12] and use a variant of SPIDEr for rewards. However, we decide to weigh the SPICE score three times as high as CIDEr to ensure roughly equal magnitude of both metrics. We also maintain the baseline on each score independently to encourage simultaneous improvement of both metrics. Every action, each corresponding to a word in the sentence, gets the final score of the entire sentence as a reward.

3.5. Training objective

Combining the approaches of [9] and [5], we use a number of component losses to optimize every part of the pipeline. To accelerate training of the feature extractor U-Net, we place a single linear layer after it, which is trained with a cross-entropy loss l_{sem} to predict the semantic class of each point given its extracted features. The offset vector predictions are trained with a directional loss l_{o_dir} and a norm loss l_{o_norm}. The score module is trained with the maximum IoU between the predicted object and all ground truth objects as the ground truth with a binary cross-entropy loss l_{sc}. To supervise the graph module, it is trained to also predict the relative orientations between the objects in terms of angle classes (intervals), which is measured by a cross-entropy loss l_{ori}. Finally, the captioning module can be trained with either a typical cross-entropy loss l_{cap} or with RL techniques. The total loss is calculated as a linear combination $l = \alpha l_{sem} + \beta l_{o\_dir} + \gamma l_{o\_norm} + \lambda l_{sc} + \zeta l_{ori} + \kappa l_{cap}$, with $\alpha = \beta = \gamma = \lambda = \kappa = 1$ and $\zeta = 0.1$, chosen so that the losses have roughly equal magnitudes.

We first train the entire pipeline on this total loss end-to-end, using a typical CE loss for the captioning module. Then, we finetune the captioning module separately using RL, while freezing the rest of the pipeline. This allows us to significantly reduce the computational cost, since reinforcement learning typically converges more slowly than traditional supervised learning.

4. Experimental Results

We conduct multiple experiments with three different variants of our model: one without any reinforcement learning finetuning (No-RL), one where RL is used for finetuning the captioning module trained using CE (warm start, WS), and one where the original captioning module is discarded and a new one trained from scratch with RL (cold start, CS). Tab. 1 shows the results. As performance metrics, we follow [5] and use the NLP metrics CIDEr (C), BLEU-4 (B), ROUGE (R), and METEOR (M) thresholded by IoU (@IoU). This means that if no predicted object corresponds to a particular ground truth object with an IoU larger than the threshold, the caption is given a score of zero. To measure object detection directly, we use object-wise mean average precision (mAP), again thresholded by IoU.

4.1. Quantitative Results

We observe that our **best performing model (WS-RL)** outperforms the baseline significantly across all metrics when measured @0.5IoU. It achieves comparable performance @0.25IoU. We observe that the biggest increases are seen in the CIDEr score, which probably results from the training schedule optimizing for a combination of CIDEr and SPICE. But the other metrics are also improved significantly, which shows that we are not just "overfitting"

Model	Variant	C@0.25	B@0.25	R@0.25	M@0.25	C@0.5	B@0.5	R@0.5	M@0.5	mAP@0.5
Scan2Cap	-	56.82	**34.18**	**55.27**	**26.29**	39.08	23.32	44.78	21.97	32.21
Ours	No-RL	49.17	31.50	53.00	24.83	39.85	24.24	**46.25**	22.05	**35.57**
Ours	WS-RL	**57.38**	33.82	52.74	25.42	**45.76**	**26.72**	46.01	**22.55**	**35.57**
Ours	CS-RL	-	-	-	-	0.01	0.00	20.65	10.45	**35.57**

Table 1. Performance of different variants of our model compared to the baseline Scan2Cap [5], as measured by various metrics thresholded by **@0.25IoU** or **@0.5IoU**. *GT-Objects* is a Scan2Cap model which has ground truth object detection information.

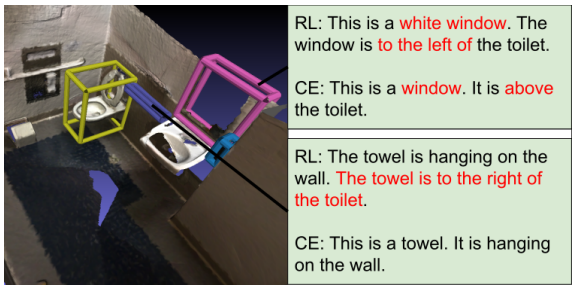
on a single metric. Even **without reinforcement learning**, our models perform better than the baseline when measured @0.5IoU. Our models also yield better object detection accuracy. When measured @0.25IoU, the No-RL model performs slightly worse than the baseline. This is not surprising, since here our advantage in object detection accuracy is not weighted as highly and may be outweighed by Scan2Cap's more extensive feature engineering, which includes multiple 2D views. On the other hand, the **cold-start** variant proved to be very difficult to train and did not achieve a useful performance, although the performance is better than randomness.

4.2. Qualitative Analysis

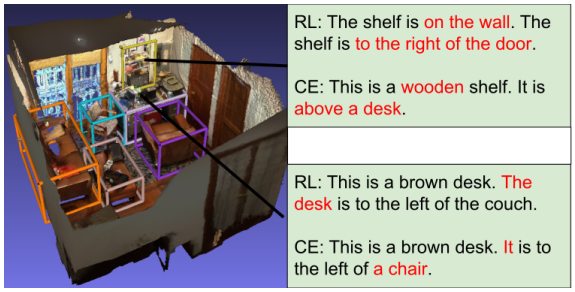
We see from Fig. 2 that the captioning module trained with reinforcement learning (RL) tends to produce more verbose results than one trained with cross-entropy (CE). In some cases, this does help to include relevant information. Consider as a particularly pronounced example of this phenomenon the second description in 2a. Both times, the towel *rail* is mistaken for a towel, however the RL-trained model correctly extends the description to include its spatial relation to the toilet. In other cases, we observe that the descriptions produced with reinforcement learning tend to introduce some redundancy and name the detected objects multiple times as opposed to the descriptions of the CE-trained module. This can be explained by the tendency of SPICE to reward multiple occurrences of the same word. In general, the choice of rewards highly impacts perceived quality of machine-generated texts [16]. For the cold-start model, we observed that it learned to reproduce some common 2- or 3-grams such as "this is a", but did not produce intelligible sentences.

5. Conclusion

Our work shows firstly that **submanifold sparse convolutions** do in fact improve both object detection and overall captioning performance. This shows that SSCs can extract features that are detailed enough for captioning tasks. Second, we show that **reinforcement learning** is feasible for captioning tasks and leads to **significantly better performance**. Lastly, we show that while it is very difficult to **directly** train using reinforcement learning from scratch, we can still reap significant benefits by just **finetuning** using reinforcement learning instead.



(a)



(b)

Figure 2. Sample sentences from two scenes, generated with a caption module trained using (warm-start) reinforcement learning (RL) and Cross-Entropy-Loss (CE).

Limitations and Future Work There remains a large number of opportunities to improve upon our method. First, we were not able to train the entire pipeline with RL end-to-end due to resource constraints. [5] showed that end-to-end training can have a positive effect. It remains to be seen whether this can be transferred to RL training as well. Similarly, it remains to be seen whether cold start RL can be made feasible. Further, our work does not yet incorporate the newest advancements in language generation, such as transformers and large pretrained models such as BERT [6] or GPT-3 [2]. Future work could also focus on using newer RL techniques such as Proximal Policy Optimization [17] or Soft Actor-Critic [8], as well as evaluating different choices of reward functions and their effect on generated captions.

References

[1] Armen Avetisyan, Manuel Dahnert, Angela Dai, Manolis Savva, Angel X. Chang, and Matthias Niessner. Scan2cad: Learning cad model alignment in rgb-d scans. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. 4

[3] R. Qi Charles, Hao Su, Mo Kaichun, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–85, 2017. 1

[4] Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language. *16th European Conference on Computer Vision (ECCV)*, 2020. 1

[5] Zhenyu Chen, Ali Gholami, Matthias Niessner, and Angel X. Chang. Scan2cap: Context-aware dense captioning in rgb-d scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3193–3203, June 2021. 1, 2, 3, 4

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. 4

[7] Evan Greensmith, Peter Bartlett, and Jonathan Baxter. Variance reduction techniques for gradient estimates in reinforcement learning. In T. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2001. 3

[8] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *CoRR*, abs/1801.01290, 2018. 4

[9] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2, 3

[10] Samuel Kiegeland and Julia Kreutzer. Revisiting the weaknesses of reinforcement learning for neural machine translation. *arXiv preprint arXiv:2106.08942*, 2021. 2

[11] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2

[12] Siqi Liu, Zhenhai Zhu, Ning Ye, Sergio Guadarrama, and Kevin Murphy. Improved image captioning via policy gradient optimization of spider. pages 873–881, 10 2017. 1, 2, 3

[13] Ramakanth Pasunuru and Mohit Bansal. Reinforced video captioning with entailment rewards. In *EMNLP*, 2017. 2

[14] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017. 1, 2

[15] Zhou Ren, Xiaoyu Wang, Ning Zhang, Xutao Lv, and Li-Jia Li. Deep reinforcement learning-based image captioning with embedding reward. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 290–298, 2017. 1, 2, 3

[16] Ananya B. Sai, Akash Kumar Mohankumar, and Mitesh M. Khapra. A survey of evaluation metrics used for nlg systems. *ACM Comput. Surv.*, 55(2), jan 2022. 4

[17] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. 4

[18] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018. 1, 2, 3

[19] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 1

[20] Weiyue Wang, Ronald Yu, Qiangui Huang, and Ulrich Neumann. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. In *CVPR*, 2018. 1

[21] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8(3–4):229–256, may 1992. 2

[22] Bo Yang, Jianan Wang, Ronald Clark, Qingyong Hu, Sen Wang, Andrew Markham, and Niki Trigoni. Learning object bounding boxes for 3d instance segmentation on point clouds. In *Advances in Neural Information Processing Systems*, pages 6737–6746, 2019. 1

[23] Wei Zhang, Bairui Wang, Lin Ma, and Wei Liu. Reconstruct and represent video contents for captioning via reinforcement learning. *IEEE transactions on pattern analysis and machine intelligence*, 42(12):3088–3101, 2019. 2