# Towards Self-Supervised Pre-Training of 3DETR for Label-Efficient 3D Object Detection

Rishabh Jain
University of Freiburg, Robert Bosch GmbH
jainrish@cs.uni-freiburg.de

Narunas Vaskevicius
Robert Bosch GmbH
narunas.vaskevicius@de.bosch.com

Thomas Brox
University of Freiburg
brox@cs.uni-freiburg.de

## Abstract

*3D Detection Transformer (3DETR) is a recent end-to-end transformer architecture for 3D object detection in 3D point clouds. In this work, we explore training and evaluation of 3DETR in a label-efficient setting on the popular 3D object detection benchmark SUN RGB-D. The performance of 3DETR declines drastically with decreasing amount of labeled data. Therefore, we investigate self-supervised pre-training of the 3DETR encoder with the spatio-temporal representation learning (STRL) framework. Opposite to our expectations, we observe that straightforward application of this framework leads to degraded representations which in some cases can even impair learning of the downstream task. To remedy this issue we extend STRL framework by introducing an auxiliary loss, which is applied to intermediate transformer layers. Our experiments demonstrate that this extension enables successful pre-training of 3DETR encoder and significantly boosts its label efficiency in the 3D object detection task.*

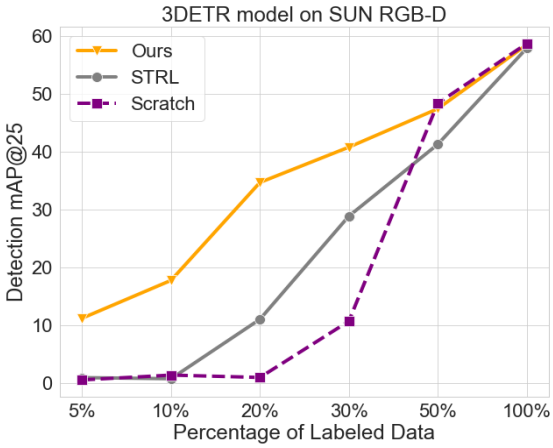


**Figure 1. Label efficiency of the pre-trained 3DETR.** 3DETR exhibits improved label efficiency when initialized with weights obtained from our pre-training. In our method, the encoder is pre-trained on synthetic shapes from ShapeNet dataset using STRL framework extended with an auxiliary loss applied to intermediate Transformer layers.

## 1. Introduction

3D object detection in point cloud data is a well studied problem [3, 6, 16, 17, 19, 20, 26, 27]. Most of the prior work relies on carefully designed deep neural networks with 3D domain-specific biases. Recent work 3DETR [15] proposes a simple alternative based on end-to-end Transformer architecture and demonstrates that it can achieve performance comparable to the state of the art. Point cloud is an unordered, permutation-invariant representation of a 3D scene which makes Transformers a natural choice for this data.

Transformer based architectures have pushed the frontiers of 2D scene understanding in the last few years [1, 5]. However, Transformers are data hungry and difficult to optimize. Large labeled datasets in 2D scene understanding [4, 14] have enabled the training of Transformers at scale to achieve superior performance. However, obtaining similar scale in 3D is prohibitively expensive. Therefore, effective representations and inductive biases must be learned from the plethora of unlabeled point cloud data and leveraged in learning the downstream scene understanding task.

Self-supervised learning has emerged as the leading approach to obtain generalizable representations and inductive biases in the feature extractors. Self-supervised pre-training has successfully supplanted supervised pre-training on ImageNet [8] in 2D scene understanding tasks. It has also found

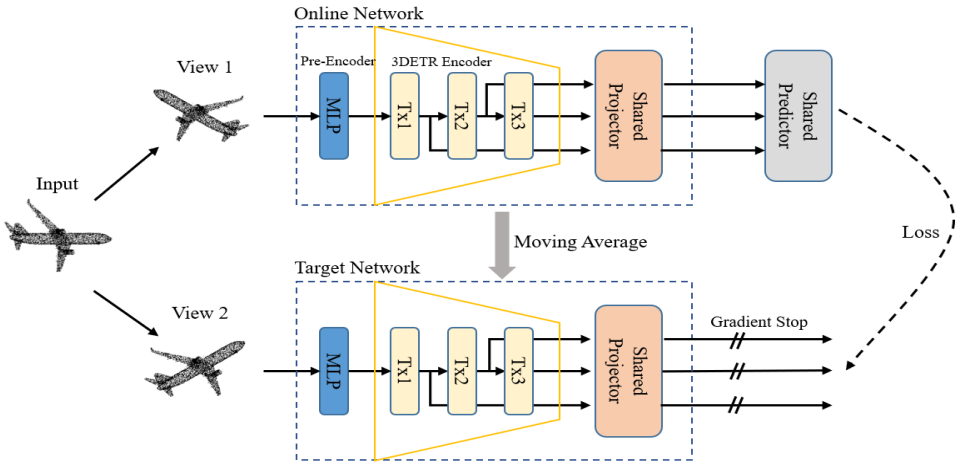


**Figure 2. Approach Overview.** We propose a simple idea of bringing close intermediate layer representations between the online and target network in the STRL framework. Tx1, Tx2 and Tx3 denotes the first, second and third Transformer layer respectively.

success in pre-training point-cloud architectures [10,23,25]. Depth Contrast [25] significantly improves label efficiency of VoteNet [16] with PointNet++ [18] pre-trained using their contrastive learning formulation. This motivates us to investigate similar approaches for 3DETR with the ultimate goal of making it less data-hungry.

In this work, we explore Spatio-temporal Self-Supervised Representation Learning (STRL) [10] for 3DETR. We choose STRL for its simplicity as it uses only-positive pairs to learn strong representations in a computationally inexpensive framework. We reveal that the straightforward application of STRL to 3DETR does not enable 3DETR to learn strong representations. We remedy it by extending the STRL framework with an auxiliary loss for intermediate feature representations. Lastly, we show that our proposed strategy learns strong representations by evaluating it for the 3D object detection task on various limited data settings, this can be seen in Fig. 1.

## 2. Method

In Sec. 2.1, first, we introduce key details about 3DETR and STRL. Then in Sec. 2.2, we describe our extension to the STRL framework.

### 2.1. Background

#### 2.1.1 3D Detection Transformer (3DETR)

3DETR consists of an encoder-decoder Transformer architecture. It takes an unordered set of $N$ points $\{p^i\}$ as the input. It is processed by a pre-encoder consisting of set-aggregation downsampling operation from [18] to extract per-point feature of dimension $d = 256$. The resulting set of $N'$ point-features $N'$ x $d$ are fed into the Transformer encoder for feature extraction. The Transformer encoder consists of 3 layers of multi-head self-attention and non-linear projections. Afterwards, the transformer decoder module uses these features and $B$ query embeddings to produce $B$ 3D bounding boxes. In our work, we focus on the masked variant of 3DETR encoder[1] because it imbues a local-feature aggregation bias and outperforms the standard Transformer model.

#### 2.1.2 Spatio-Temporal Self-Supervised Representation Learning (STRL)

STRL is based on "Bootstrap your own latent" [7] framework. It utilizes two copies of the neural network, referred to as *online* and *target* networks. For different views of the point cloud, online network is trained to predict the target network representations. At the same time, target network is updated as a slow-moving average of the online network. Different views of the point cloud are obtained by randomly applying augmentations such as cropping, cutout, translation and rotations. We discuss further implementation details in Sec. 3.1.2.

### 2.2. Auxiliary Loss in STRL

We observe in Fig. 3 that straightforward application of STRL to 3DETR leads to degraded feature quality. We hypothesize that this is caused by insufficient 3D-specific inductive bias in the 3DETR encoder making it harder to learn 3D representations using self-supervised learning. As a result, enforcing augmentation invariance at the final representations does not produce generalizable features from the unstructured point cloud data. To this end, we extend

[1] In this work, we refer to 3DETR-masked encoder simply as 3DETR encoder for brevity.

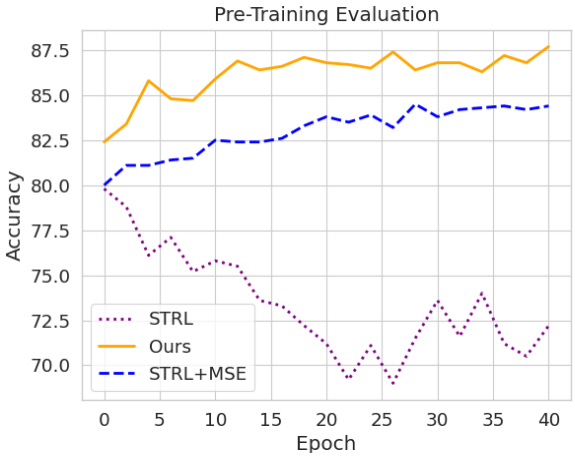


**Figure 3. Investigating Pretraining.** Linear Evaluation for shape classification on ModelNet40. STRL+MSE denotes the idea in [11] implemented in STRL framework.

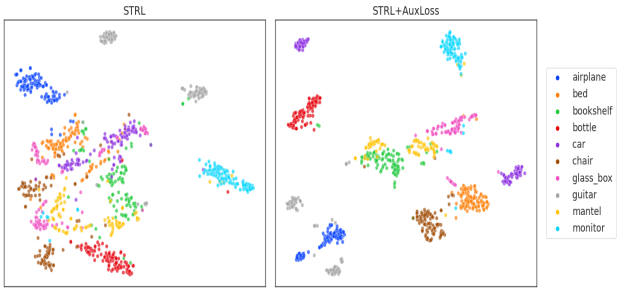


**Figure 4. Visualization of learned features.** We visualize the extracted features for each sample from 10 most frequent classes in ModelNet40 test set using t-SNE. Pretrained using STRL and STRL+AuxLoss (Ours). Best viewed in colour.

the STRL framework to include an auxiliary loss that enforces augmentation-invariance at intermediate layers. Our method brings close the representations learned at intermediate layers between the online and target network. As illustrated in Fig. 2, our approach projects the intermediate representations using the shared Projector and Predictor before applying the regression loss used in STRL. A similar idea of auxiliary loss at intermediate layers is used in DETR [1] and 3DETR [15] for the decoder. We hypothesize that the auxiliary loss introduces a stronger signal at intermediate layers improving the gradient flow in the network thereby facilitating the optimization.

Similar to our work, [11] showed that bringing intermediate layer representations closer improves MoCo [8] pre-training on medical datasets. Contrary to our approach, [11] minimizes the mean squared error between the intermediate representations without using the Projector and Predictor. In the Sec. 3.2, we compare their idea in the STRL framework to pre-train 3DETR with our approach.

# 3. Experiment and Results

In Sec. 3.1, we describe the implementation details of our experiments along with the datasets used. In Sec. 3.2, we investigate the features learned from pre-training. In Sec. 3.3, we present the results of downstream training.

## 3.1. Experimental Setup

### 3.1.1 Datasets

**ShapeNet.** [2] This is a synthetic dataset consisting of 3D CAD models of common objects. Following STRL, we learn self-supervised representations on the ShapeNet dataset. We utilize the pre-processed dataset from the official STRL GitHub repository. The version of ShapeNet we obtained and used consists of 57,448 synthetic objects from 55 categories.

**SUN RGB-D.** For training in the downstream 3D object detection task we use SUN RGB-D-v1 dataset [21]. SUN RGB-D has 5284 RGB-D training samples and 5051 validation samples with oriented bounding box labels for 37 object categories. Following 3DETR, we train and report the detection performance on 10 most frequent categories.

### 3.1.2 Implementation Details

**Pre-Training.** We use the 3DETR encoder as described in [15]. Pre-Training is done on ShapeNet dataset using the same augmentations as STRL. We pre-train for 40 epochs with Adam optimizer [12] and a learning rate of 1e-3. Momentum of 0.996 for updating the target network parameters, and an effective batch size of 32 is used on two NVIDIA Tesla V100 GPUs. STRL pipeline is used from the official repository and extended with our auxiliary loss.

**Downstream Training.** We use the same pipeline from the 3DETR official repository. Pre-encoder and encoder weights are initialized from the pre-trained models. The models are trained for 512 epochs with a batch size of 16 on one NVIDIA Tesla V100 GPU. All other hyperparameters are kept as described in [15]. For studying label efficiency, we used random search to sample the dataset to ensure that the label distribution of sampled data closely matches the original label distribution.

**Linear Evaluation.** For linear evaluation of the pre-trained models, we max-pool the feature extracted from the last layer of the transformer in the 3DETR encoder as done in [15] for evaluating 3DETR encoder on Shape Classification. A SVM [9] is trained on top of features extracted from the frozen encoder on ModelNet40 [24] train split and evaluated on test split similar to [10].

| Initialization | SUN RGB-D | | | | | |
|---|---|---|---|---|---|---|
| | 5% | 10% | 20% | 30% | 50% | 100% |
| Scratch | 0.5 | 1.4 | 1.1 | 10.7 | **48.4** | **58.7** |
| STRL | 0.9 | 0.8 | 11.1 | 28.9 | 41.3 | 57.9 |
| STRL+AuxLoss (Ours) | **11.3** | **17.8** | **34.7** | **40.8** | 47.4 | 58.5 |

Table 1. **Downstream Evaluation.** $mAP_{25}$ on the full SUN RGB-D validation set is reported for the model when trained with varying amount of data.

### 3.2. Linear Evaluation for Shape Classification

Linear evaluation of the features is a standard method in assessing the quality of features learned from pre-training. We perform it as described in Sec. 3.1.2 for the task of shape classification. Features are evaluated after every two epochs of pre-training. The Fig. 3 compares our approach against STRL. We observe that the accuracy deteriorates with pre-training in the case of STRL. After complete training, our method achieves an accuracy of 87.6% which is a 21% improvement over STRL.

We observe that combining STRL with MSE loss as done with MoCo in [11] also improves over STRL, but converges to 3% point lower accuracy than our approach as seen in Fig. 3.

We further investigate the pre-training by visualizing the learned features with STRL and our method using t-SNE [22] dimensionality reduction. Fig. 4 displays the embeddings of 10 most frequent classes in ModelNet40 test set. We observe that our method produces more separable embeddings than STRL.

### 3.3. Label Efficiency in 3DETR

We study label efficiency in 3DETR by training it on partial data from the training split of SUN RGB-D dataset and evaluating the model on the full validation split. Investigation is performed using 5%, 10%, 20%, 30%, 50% and 100% of the training split. We compare the 3DETR model trained from scratch *vs*. model initialized with pre-trainined encoder weights.

Complete results are reported in Tab. 1. Our approach significantly boosts the performance in 5%, 10%, 20% and 30% data setting with the **maximum gain of 23.6 points over STRL** in 20% data setting. However, we observe no gains when ample training data is available as seen on 50% and 100% data setting.

## 4. Feature Similarity Analysis

Quality of features extracted by the encoder is an important factor to achieve label efficiency. We observe that the success of our method in linear evaluation of the features translates to the downstream 3D object detection task.

| SUN RGB-D | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Method | 5% | | | 10% | | | 20% | | |
| | Tx1. | Tx2. | Tx3. | Tx1. | Tx2. | Tx3. | Tx1. | Tx2. | Tx3. |
| Scratch | 0.56 | 0.31 | 0.34 | 0.49 | 0.29 | **0.36** | 0.43 | 0.31 | 0.35 |
| Ours | **0.71** | **0.47** | **0.38** | **0.74** | **0.50** | 0.31 | **0.78** | **0.53** | **0.41** |

Table 2. **Investigating feature similarity using CKA.** Features of the models learned with limited annotations are compared to the features learned by model trained from scratch on 100% of labeled data. A higher value indicates more feature similarity. Tx1, Tx2 and Tx3 denotes the Transformer layers in the encoder.

To further analyze our method, we compute the similarity between features learned in the encoder after downstream training on 5%, 10% and 20% of the labeled data with the features learned on 100% of the labeled data. We use Centered Kernal Alignment (CKA) [13] to compute similarity as it has been shown to be reliable in comparing representations from neural networks trained with different initialization.

Tab. 2 reports the CKA values for features extracted from intermediate layers in the encoder. We observe that the representations learned with our method are closer to the ones learned using 100% labels than the representations learned on low data from scratch. This indicates the success of our pre-training approach in boosting learning with limited labeled data.

## 5. Conclusion

We present a simple extension to the STRL framework which leads to stronger representation learning in 3DETR architecture. We evaluate the proposed method on a simulated low-data setting by sampling the SUN RGB-D train set and find that 3DETR pre-trained with our approach exhibits better label efficiency compared to both STRL and training from scratch. We take a first step towards making 3DETR less data-hungry by utilizing self-supervised representation learning. In the future, we would like to explore other self-supervised learning methods to learn label-efficient Transformers for various 3D scene understanding tasks.

## Acknowledgments

This work was partly supported by the EU Horizon 2020 research and innovation program under grant agreement No. 101017274 (DARKO).

## References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European confer-*

*ence on computer vision*, pages 213–229. Springer, 2020. 1, 3

[2] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 3

[3] Jintai Chen, Biwen Lei, Qingyu Song, Haochao Ying, Danny Z Chen, and Jian Wu. A hierarchical graph network for 3d object detection on point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 392–401, 2020. 1

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1

[6] Francis Engelmann, Martin Bokeloh, Alireza Fathi, Bastian Leibe, and Matthias Nießner. 3d-mpa: Multi-proposal aggregation for 3d semantic instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9031–9040, 2020. 1

[7] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020. 2

[8] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 1, 3

[9] Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28, 1998. 3

[10] Siyuan Huang, Yichen Xie, Song-Chun Zhu, and Yixin Zhu. Spatio-temporal self-supervised representation learning for 3d point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6535–6545, 2021. 2, 3

[11] Aakash Kaku, Sahana Upadhya, and Narges Razavian. Intermediate layers matter in momentum contrastive self supervised learning. *Advances in Neural Information Processing Systems*, 34, 2021. 3, 4

[12] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 3

[13] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pages 3519–3529. PMLR, 2019. 4

[14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 1

[15] Ishan Misra, Rohit Girdhar, and Armand Joulin. An end-to-end transformer model for 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2906–2917, 2021. 1, 3

[16] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9277–9286, 2019. 1, 2

[17] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 918–927, 2018. 1

[18] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 2

[19] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10529–10538, 2020. 1

[20] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 770–779, 2019. 1

[21] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 567–576, 2015. 3

[22] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 4

[23] Hanchen Wang, Qi Liu, Xiangyu Yue, Joan Lasenby, and Matt J Kusner. Unsupervised point cloud pre-training via occlusion completion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9782–9792, 2021. 2

[24] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 3

[25] Zaiwei Zhang, Rohit Girdhar, Armand Joulin, and Ishan Misra. Self-supervised pretraining of 3d features on any point-cloud. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10252–10263, 2021. 2

[26] Zaiwei Zhang, Bo Sun, Haitao Yang, and Qixing Huang. H3dnet: 3d object detection using hybrid geometric primitives. In *European Conference on Computer Vision*, pages 311–329. Springer, 2020. 1

[27] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4490–4499, 2018. 1