4DContrast: Contrastive Learning with Dynamic Correspondences for 3D Scene Understanding (Supplemental Material)

Yujin Chen, Matthias Nießner, and Angela Dai

Technical University of Munich
{yujin.chen,niessner,angela.dai}@tum.de

A Additional Quantitative Analysis

Alternative 3D Backbone vs 4D Pre-training. Our 4D pre-training can help to learn objectness priors from dynamic object movement, in contrast to multiple 3D backbones. To demonstrate this, we pre-trained with an additional 3D backbone (comparable to our 4D backbone in parameters and UNet structure); this resulted in worse performance than our 4D pre-training which has +0.5 mIoU and +1.4 mAP@0.5 vs. multiple 3D backbones in the tasks of 3D semantic and instance segmentation on ScanNet (as shown in Table 8).

Sequence Length Ablation. We study the effect of the sequence length of the generated dynamic data used for pre-training in Table 9. We consider sequences of length 3, 4, or 5, and set the batch size (number of sequences) to 16, 12, and 10, respectively, to balance the scene frames in each batch during pre-training. We find a sequence length of 4 results in more effective feature learning for downstream tasks.

Comparison of Different Contrastive Frameworks. As analyzed in Section 3.1 of the main paper, SimSiam [3] enables contrastive learning without requiring negative samples or large batch size. We thus verify how these attributes fit our high-dimensional pre-training design by comparing SimSiam and SimCLR [2] as our contrastive framework. As shown in Figure 7, Ours (SimCLR) removes the 3D and 4D predictors from Ours (SimSiam), and uses a match average pooling to average 4D features in different frames according to spatial correspondences. For each pair of frames (F_i, F_j) in a train sequence, we apply a 3D contrastive loss $\bar{\mathcal{L}}^{3D}$ as $\mathcal{L}^{3D}$ (Eq. 4 in Section 3.2). Similar to $\mathcal{L}^{3D4D}$ (Eq. 6 in Section 3.2), we use a 3D-4D contrastive loss $\bar{\mathcal{L}}^{3D4D}$ to establish

Table 8. Comparisons of alternative 3D backbone and our 4D backbone on ScanNet fine-tuning.

Task	Baseline	Two 3D Backbones	Ours
Sem.Seg (mIoU)	70.0	71.8 (+1.8)	**72.3** (+2.3)
Ins.Seg (mAP@0.5)	53.4	56.2 (+2.8)	**57.6** (+4.2)

Table 9. Effect of sequence length of pre-training dynamic data on ScanNet semantic segmentation fine-tuning. A sequence length of 4 helps 4DContrast get higher semantic segmentation mIoU.

Sequence Length	3	4	5
mIoU	71.9	**72.3**	71.0

Table 10. Comparisons of SimCLR and SimSiam as our contrastive learning framework on ScanNet semantic segmentation fine-tuning.

Contrastive Framework	Baseline	Ours (SimCLR)	Ours (SimSiam)
mIoU	70.0	71.6 (+1.6)	**72.3** (+2.3)

Table 11. 3D object detection on ScanNet with H3DNet.

Method	mAP@0.5
H3DNet	43.4
Ours + H3DNet	**47.7** (+4.3)

correspondence between 3D features and the averaged 4D features. The Hardest-Contrastive loss is borrowed from FCGF [5] and PointContrast [7]. Note that in our implementation, we find that the PointInfoNCE loss [7] is not as stable as Hardest-Contrastive loss, likely due to different data augmentation methods between Ours (SimCLR) and PointContrast. As shown in Table 10, 4DContrast coupled with SimSiam framework more effectively leverages the learned representations for improved semantic segmentation performance on ScanNet.

H3DNet Object Detection with 4DContrast. In Table 11, we apply our pre-trained weights to H3DNet [8] (1 descriptor computation tower of its backbone architecture). 4DContrast surpasses training from scratch by 4.3 mAP@0.5 on ScanNet.

Mix3D Semantic Segmentation with 4DContrast. While 4DContrast focuses on imbuing 4D priors during pre-training to provide effective features for a variety of downstream tasks, Mix3D [6] tackles a complementary problem of data augmentation during training. As shown in Table 12, our pre-training can be used together with Mix3D to further improve semantic segmentation performance on ScanNet (geometry only input).

MinkowskiNet 3D Classification with 4DContrast. We evaluate ModelNet classification accuracy in comparison with MinkowskiNet [4] trained from scratch for various voxel sizes in Table 13. Our pre-training shows consistent improvements in both settings.

S3DIS dataset. We finetune our pre-trained weights for S3DIS [1] segmentation (geometry-only), and consistently improve over training from scratch (as shown in Table 14): we achieve +2.4 mIoU in semantic segmentation (61.0 ours vs 58.6 scratch) and +7.4 mAP@0.5 in instance segmentation (53.2 vs 45.8).

Table 12. Semantic segmentation on ScanNet val with Mix3D.

Method	mIoU
Mix3D + MinkowskiNet	73.9
Ours + Mix3D + MinkowskiNet	**74.6** (+0.7)

Table 13. 3D classification on ModelNet with Mix3D.

Method	Voxel Size	Acc
MinkowskiNet	0.05	86.1
Ours + MinkowskiNet	0.05	**88.5** (+2.4)
MinkowskiNet	0.02	90.7
Ours + MinkowskiNet	0.02	**91.8** (+1.1)

Table 14. Semantic and instance segmentation on S3DIS.

Task	scratch	Ours
Sem.Seg. (mIoU)	58.6	**61.0** (+2.4)
Ins.Seg. (mAP@0.5)	45.8	**53.2** (+7.4)

B Network Architecture Details

Figure 8 details our network architectures. The backbones are a U-Net architecture with sparse convolutions [4]. We use a 34-layer U-Net as the 3D backbone and a 14-layer U-Net as the 4D backbone. For the 3D and 4D projectors, we use a one-layer sparse convolutional layer with kernel size as $1\times1\times1$ and $1\times1\times1\times1$, respectively. For the 3D and 4D predictor, we use two sparse convolutional layers. We repeat occupancy into 3-dimension to fit the network input dimension of 3.

C Visualization of the Generated 4D Data

Figure 9 shows the generated 4D data by scene-object augmentation (as described in Section 3.3 of the main paper).

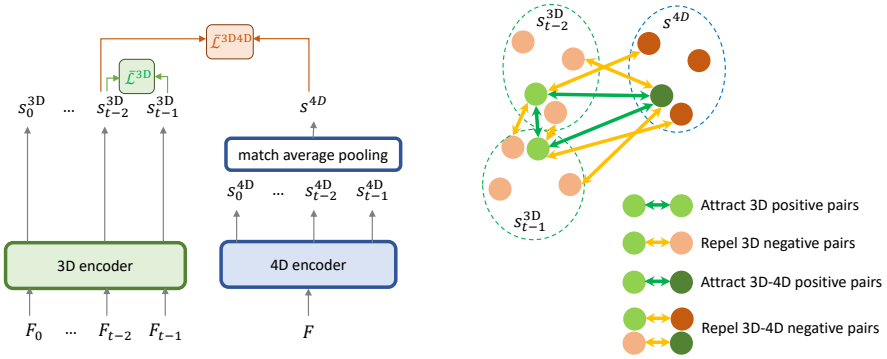

Fig. 7. Network architectures of our method using SimCLR as the contrastive learning framework. **Left:** we show 3D-3D and 4D-4D losses across frame and spatio-temporal correspondence. We only visualize the inter-frame correspondence for F_{t-2} and F_{t-1}, and only spatio-temporal correspondence for for F_{t-2}, while those loss are established across all pairs of frames for $\bar{\mathcal{L}}^{3D}$ and all frames for $\bar{\mathcal{L}}^{3D4D}$. **Right:** we visualize the contrastive losses between 3D feautres of F_{t-2} and F_{t-1} and the 4D feature after match average pooling. The positive pairs is same with Section 3.2 and the negative losses is only calculated for the hardest negative pairs.

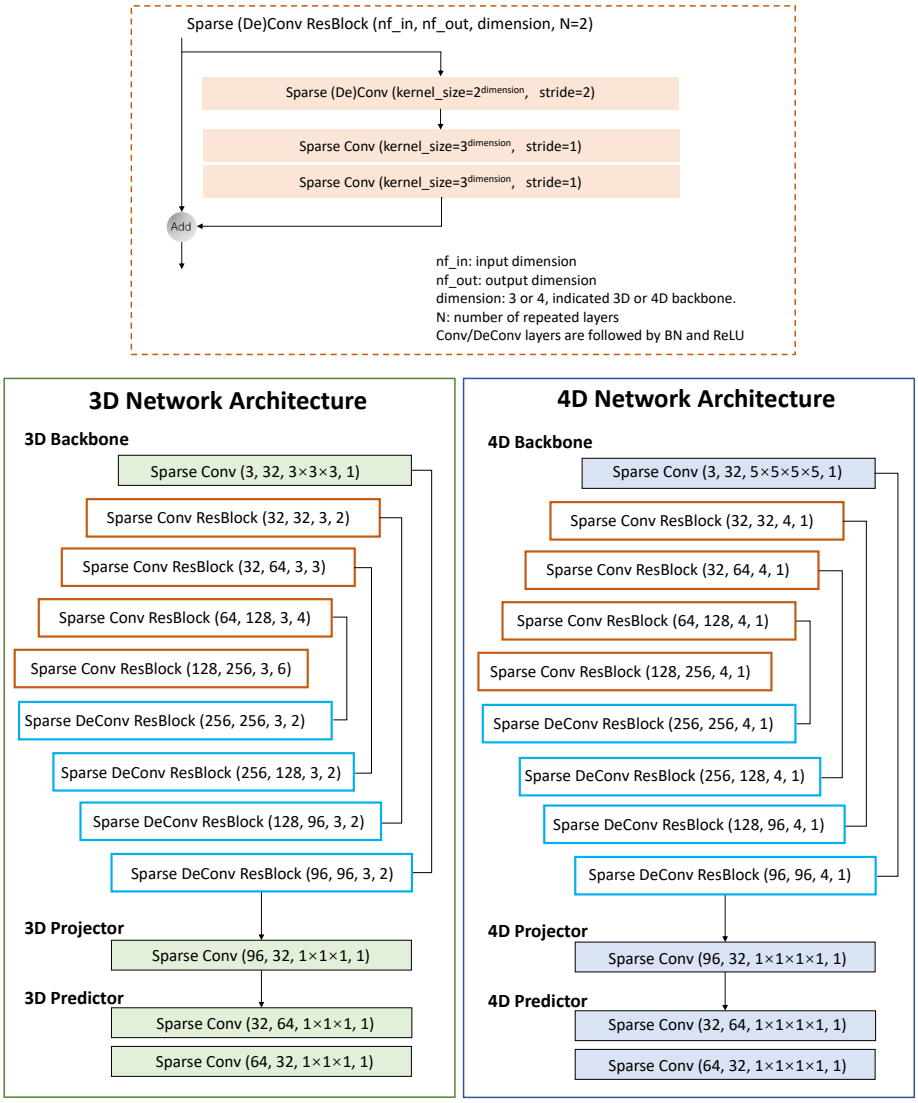

Fig. 8. Network architectures of 4DContrast for pre-training. For downstream fine-tuning, only the 3D backbone is kept and fine-tuned.

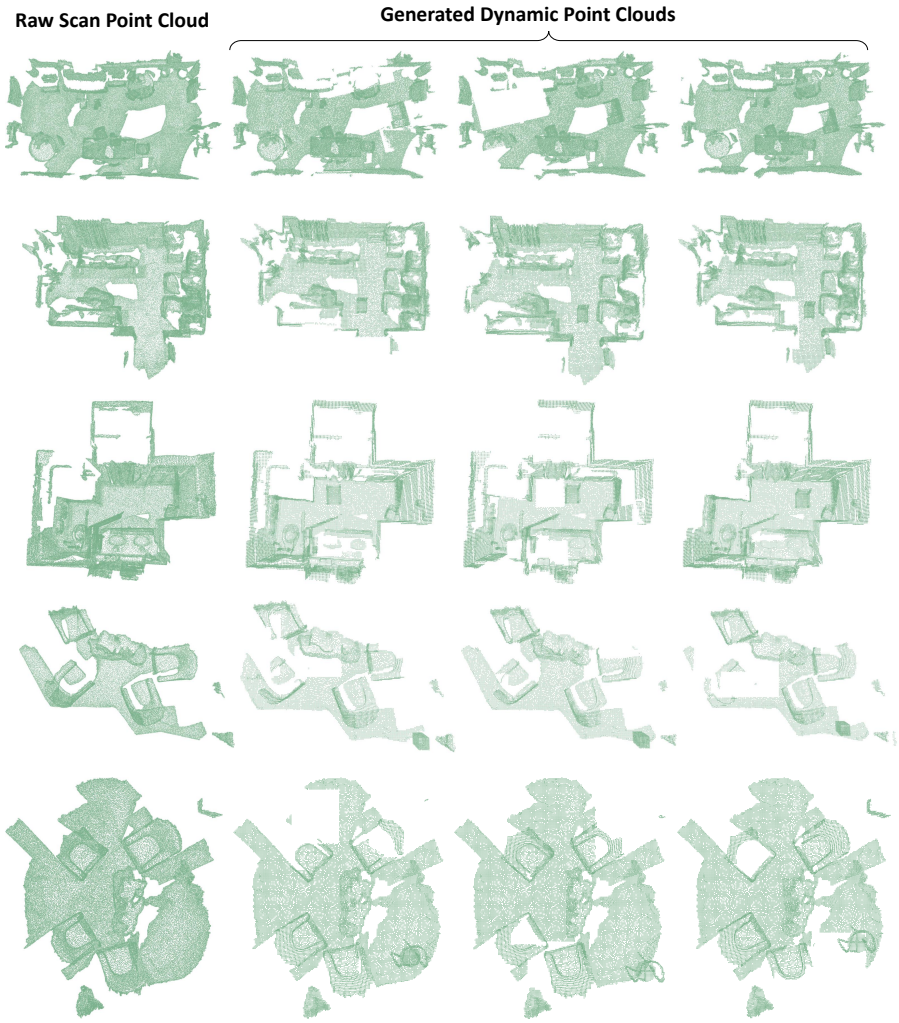

Fig. 9. Visualization of generated 4D sequence data. Each row corresponds to a sampled scene. From left to right: raw scan mesh vertices as input cloud, generated dynamic point clouds with scene augmentation and object motion (in three frames).

References

1. Armeni, I., Sax, S., Zamir, A.R., Savarese, S.: Joint 2d-3d-semantic data for indoor scene understanding. arXiv preprint arXiv:1702.01105 (2017) 2
2. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International Conference on Machine Learning. pp. 1597–1607 (2020) 1
3. Chen, X., He, K.: Exploring simple siamese representation learning. In: Conference on Computer Vision and Pattern Recognition. pp. 15750–15758 (2021) 1
4. Choy, C., Gwak, J., Savarese, S.: 4d spatio-temporal convnets: Minkowski convolutional neural networks. In: Conference on Computer Vision and Pattern Recognition. pp. 3075–3084 (2019) 2, 3
5. Choy, C., Park, J., Koltun, V.: Fully convolutional geometric features. In: International Conference on Computer Vision. pp. 8958–8966 (2019) 2
6. Nekrasov, A., Schult, J., Litany, O., Leibe, B., Engelmann, F.: Mix3d: Out-of-context data augmentation for 3d scenes. In: 2021 International Conference on 3D Vision (3DV). pp. 116–125. IEEE (2021) 2
7. Xie, S., Gu, J., Guo, D., Qi, C.R., Guibas, L., Litany, O.: Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In: European Conference on Computer Vision. pp. 574–591 (2020) 2
8. Zhang, Z., Sun, B., Yang, H., Huang, Q.: H3dnet: 3d object detection using hybrid geometric primitives. In: European Conference on Computer Vision. pp. 311–329 (2020) 2