# Supplementary Material

**Renrui Zhang**$^{1,2}$, **Ziyu Guo**$^{2}$, **Rongyao Fang**$^{1}$,
**Bin Zhao**$^{2}$, **Dong Wang**$^{2}$, **Yu Qiao**$^{2}$, **Hongsheng Li**$^{1,3}$, **Peng Gao**$^{2}$

$^{1}$ CUHK-SenseTime Joint Laboratory, The Chinese University of Hong Kong,
$^{2}$ Shanghai AI Laboratory, $^{3}$ Centre for Perceptual and Interactive Intelligence Limited
`{zhangrenrui, gaopeng}@pjlab.org.cn`
`hsli@ee.cuhk.edu.hk`

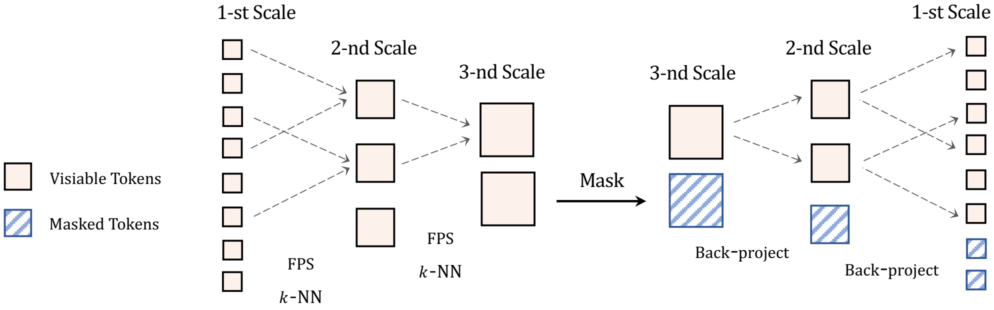


Figure 1: **Pipeline of the multi-scale masking.** We first obtain the multi-scale representation of input point clouds by FPS and $k$-NN. Then, we random mask the points at the highest level and back-project the visible positions into precedent scales.

## 1 Additional Related Work

**Transformers.** Transformers [21] are first proposed in natural language processing to capture the inter-word relations in a long sentence, and have dominated most language tasks [6, 16, 17, 2]. Motivated by this, Vision Transformers [7] and DETR [3] introduce the transformer architecture into computer vision, and stimulate follow-up works to effectively apply transformers to a wide range of vision tasks, such as image classification [18, 13, 14], object detection [31, 30, 9], semantic segmentation [24] and so on [12]. For 3D understanding, transformer-based networks are also adopted for shape classification, part segmentation [10, 29], 3D object detection from point clouds [15] and monocular images [28]. As a pioneer work, PCT [10] utilizes neighbor embedding layers to aggregate local features and processes the downsampled point clouds by transformer blocks. PoinTr [26] and Point-BERT [27] divide point clouds into multiple spatial local patches and utilize standard transformers of plain architectures to encode the patches. On top of that, we propose Point-M2AE with a hierarchical encoder-decoder transformer, which is designed for MAE-style self-supervised point cloud pre-training and can well capture the multi-scale features of point clouds.

## 2 Implementation Details

**Positional Encodings.** To complement the 3D spatial information, we apply positional encodings to all attention layers in Point-M2AE. For point tokens $T_i^v$ or $\{H_i^m, H_i^v\}$ at stage $i$, we utilize a two-layer MLP to encode its corresponding 3D coordinates $P_i^v$ or $\{P_i^m, P_i^v\}$ into $C_i$-channel vectors, and element-wisely add them with the token features before feeding into the attention layer.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

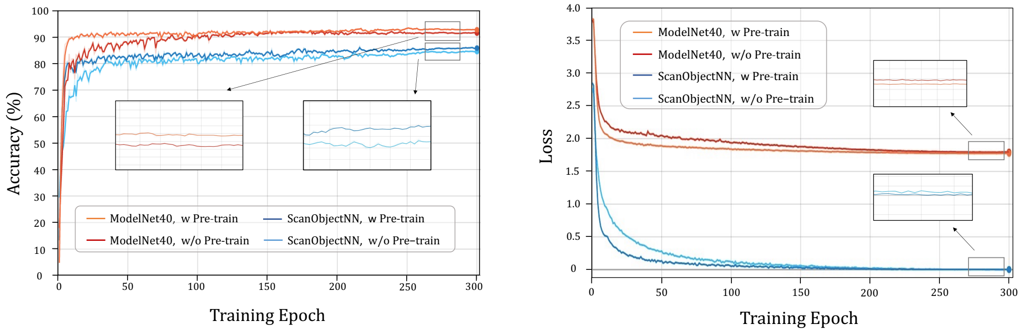
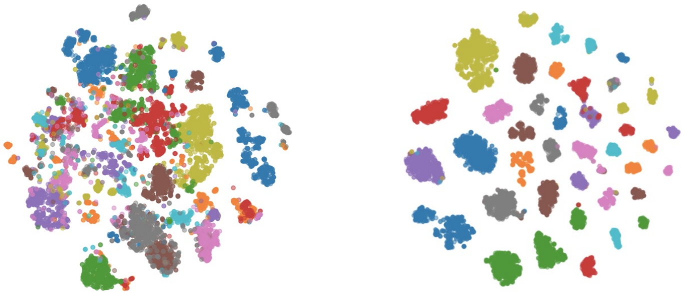


Figure 2: **Learning curves of Point-M2AE with and without pre-training.** We visualize the accuracy (Left) and loss curves (Right) on ModelNet40 [23] and ScanObjectNN [19]. We zoom in on the converged accuracy and loss for comparison.



Before Fine-tuning

After Fine-tuning

Figure 3: **t-SNE [20] visualization on ModelNet40 [23].** We show the features distribution extracted by Point-M2AE before (Left) and after (After) the fine-tuning.

**Self-supervised Pre-training.** Following previous works [1, 22], we sample 2,048 points from each 3D shape in ShapeNet [4] for pre-training Point-M2AE. We pre-train the network for 300 epochs with a batch size 128 and adopt AdamW [11] as the optimizer. We set the initial learning rate and the weight decay as $10^{-4}$ and $5\times10^{-2}$, respectively, and adopt the cosine scheduler along with a 10-epoch warm-up. We utilize the common random scaling and random translation for data augmentation during pre-training. For linear SVM on ModelNet40 [23], after the hierarchical encoder, we use both max and average pooling to aggregate the features between point tokens, and sum the two pooled features as the encoded global feature of the point cloud.

**Shape Classification.** We fine-tune Point-M2AE on two datasets for shape classification. The widely adopted ModelNet40 [23] consists synthetic 3D shapes of 40 categories, in which 9,843 samples are for training, and the other 2,468 are for validation. The challenging ScanObjectNN [19] contains 11,416 training and 2,882 validation point clouds of 15 categories, which are captured from the noisy real-world scenes and thus have domain gaps with the pre-trained ShapeNet [4] dataset. ScanObjectNN is divided into three splits for evaluation, OBJ-BG, OBJ-ONLY and PB-T50-RS, where PB-T50-RS is the most difficult for recognition. We respectively sample 1,024 and 2,048 points from each 3D shape of ModelNet40 and ScanObjectNN, and utilize only 3-channel coordinates as inputs. The same training settings are adopted for the two datasets. We fine-tune the network for 300 epochs with a batch size 32, and set the learning rate as $5\times10^{-4}$ with a weight decay $5\times10^{-2}$. For other training hyper-parameters, we keep them the same as the pre-training experiment.

**Part Segmentation.** ShapeNetPart [25] contains 16,881 synthetic 3D shapes of 16 object categories and 50 part categories, where 14,007 and 2,874 samples are respectively for training and validation. We sample 2,048 points from each shape as inputs, and predict the part categories for all points. We fine-tune Point-M2AE for 300 epochs with a batch size 16 and set the learning rate as $2\times10^{-4}$ with a weight decay 0.1. Other training settings are the same as the shape classification experiments.

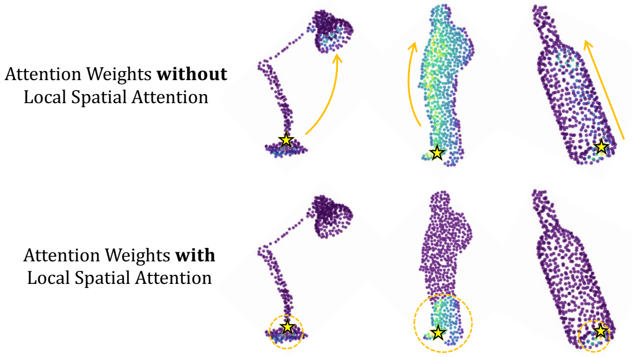

Attention Weights **without** Local Spatial Attention
Attention Weights **with** Local Spatial Attention

Figure 4: **Visualization of local spatial attention.** We visualize the attention weights without (Top) and with (Bottom) local spatial attention. The query points are marked by stars. The attention scopes are marked by arrows and dotted circles in yellow.

**Few-shot Classification.** We follow previous works [27, 1, 22], to adopt the "$K$-way $N$-shot" settings on ModelNet40 [23] for few-shot classification. We randomly select $K$ out of 40 classes and sample $N$+20 3D shapes per class, $N$ for training and 20 for testing. We evaluate Point-M2AE on four few-shot settings: 5-way 10-shot, 5-way 20-shot, 10-way 10-shot, and 10-way 20-shot. To alleviate the variance of random sampling, we conduct 10 independent runs for each few-shot setting and report the average accuracy and standard deviation. We adopt the same training settings as shape classification experiments but only fine-tune Point-M2AE for 150 epochs.

**3D Object Detection.** We pre-train and fine-tune Point-MAE for 3D object detection both on ScanNetV2 [5]. The dataset contains 1,513 scanned indoor scenes with axis-aligned 3D bounding boxes for 18 categories, 1,201 for training and 312 for validation. As we adopt the same encoder architecture in 3DETR-m [15] with 2 stages, we set the stage number of decoder as 1, which accords with the regulation of $S$-stage encoder and $(S-1)$-stage decoder. We pre-train Point-M2AE for 1,080 epochs with the learning rate $5\times10^{-4}$, and follow other hyper-parameters in the experiment of pre-training on ShapeNet [4]. For fine-tuning, we adopt the same settings as training 3DETR-m from scratch in the original paper [15] for fair comparison.

## 3 Additional Visualization

**Multi-scale Masking Pipeline.** In Figure 1, we show the simplified masking pipeline, which clearly illustrates how the mask is generated at the highest scale and back-projects to precedent layers.

**Learning Curves.** To compare the training with and without pre-training, we present their loss and accuracy curves on ModelNet40 [23] and ScanObjectNN [19]. As shown in Figure 2, the curves with pre-training converge faster and achieve higher classification accuracy than the curves without pre-training. This fully demonstrates the effectiveness of Point-M2AE's hierarchical pre-training.

**t-SNE [20].** In Figure 3, we visualize the features distribution extracted by Point-M2AE before and after fine-tuning on ModelNet40 [23]. As shown, Point-M2AE right after pre-training can already produce discriminative features for different categories without fine-tuning. Then, the fine-tuning further clusters the features of the same category and separates those of different categories.

**Local Spatial Attention.** We visualize the attention weights with and without the local attention on ModelNet40 [23] in Figure 4. As shown, with the local attention, the query point (marked by star) only has large attention values within a local spatial range (marked by yellow dotted circles), other than scattering over the entire 3D shape (marked by yellow arrows). This enables each point to concentrate more on neighboring local features in early stages for capturing detailed structures.

**Part Segmentation Results.** The fine-grained 3D patterns learned by our hierarchical architecture largely benefits 3D downstream tasks with dense prediction, e.g., part segmentation. In Figure 5, we

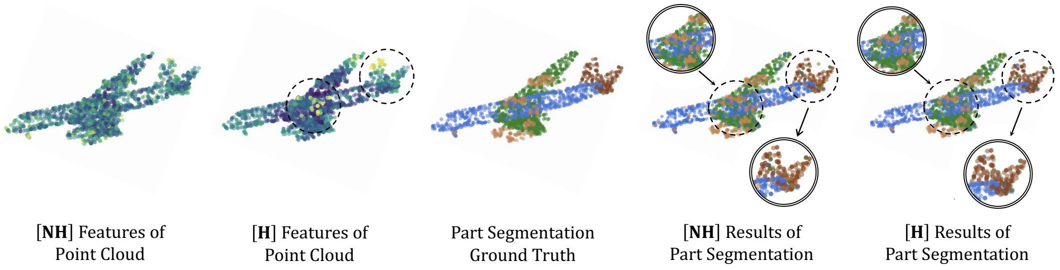

[NH] Features of Point Cloud
[H] Features of Point Cloud
Part Segmentation Ground Truth
[NH] Results of Part Segmentation
[H] Results of Part Segmentation

Figure 5: **Visualization of part segmentation results.** We denote the outputs from hierarchical and non-hierarchical architectures as **[NH]** and **[H]**, respectively. For an input point cloud (Middle), we visualize its extracted features (Left) and part segmentation results (Right).

Table 1: **Transformer stages.** We experiment different stage number of the hierarchical encoder and decoder in Point-M2AE.

| Encoder | Decoder | Acc. (%) |
|---|---|---|
| 3 | 2 | **92.9** |
| 2 | 1 | 91.8 |
| 4 | 3 | 90.4 |
| 3 | 3 | 90.7 |

Table 2: **Transformer blocks.** Based on the 3-stage encoder and 2-stage decoder, we experiment different block numbers per stage.

| Encoder | Decoder | Acc. (%) |
|---|---|---|
| 5 | 1 | **92.9** |
| 4 | 1 | 92.7 |
| 3 | 1 | 92.6 |
| 5 | 2 | 91.7 |

compare our Point-M2AE with multi-stage, [H], and single-scale, [NH], architectures by visualizing the extracted point features and the segmentation results on ShapeNetPart [25]. As shown, the multi-scale architecture predicts more fine-grained part labels for the objects.

## 4 Additional Ablation Study

**Transformer Stages.** Each stage in Point-M2AE encodes the corresponding scale of the point cloud. In Table 1, we explore the best stage number of both encoder and decoder for learning multi-scale point cloud features during pre-training. As reported, the 3-stage encoder with 2-stage decoder performs the best. If the decoder also has three stages as the encoder, and reconstructs the point cloud at the 1-th scale, the performance would be adversely influenced.

**Transformer Blocks.** In each stage, we apply several transformer blocks to encode features of the point tokens. We experiment with different block numbers in each stage of the encoder and decoder in Table 2. We observe that stacking five blocks per stage for encoder and only one block for decoder achieve the highest accuracy. This asymmetric architecture enforces the encoder to contain more semantic information of the point cloud, which benefits the transfer capacity of Point-M2AE.

**Fine-tuning Settings.** For fine-tuning on downstream classification tasks, we obtain the global feature from point tokens by pooling, and apply a MLP-based head for classification. In Table 3, we investigate different pooling operations along with the class token method to integrate features of all point tokens. Referring to [7], we concatenate a learnable class token with the point tokens at the 1-st scale, and feed them into the hierarchical encoder. After encoding, we directly utilize this class token as the global feature for classification. As reported, 'max

Table 3: **Fine-tuning settings.** For 'max + ave. pooling', we adopt max and average pooling to obtain two global features and sum them as the input of classification head. 'w/o Local Atten.' denotes vanilla global self-attention.

| Settings | ModelNet40 | ScanObjectNN |
|---|---|---|
| max pooling | 93.3 | 85.98 |
| average pooling | 92.8 | 85.66 |
| max + ave. pooling | **94.0** | **86.43** |
| class token | 93.4 | 86.02 |
| w/o Local Atten. | 93.5 | 85.82 |

+ ave. pooling' performs the best for fine-tuning, which is our default in all shape classification experiments. We also show the classification results without local spatial attention layers, which illustrates the significance of encoding local features with increasing receptive fields.

Table 4: **Pre-training losses.** 'CD' and 'EMD' denote Chamfer Distance and Earth Mover's Distance losses.

| L2-norm CD | L1-norm CD | EMD | Acc. (%) |
|---|---|---|---|
| ✓ | - | - | **92.9** |
| - | ✓ | - | 91.1 |
| - | - | ✓ | 91.9 |
| ✓ | - | ✓ | 92.4 |
| - | ✓ | ✓ | 91.3 |

**Pre-training Loss Functions.** Except for the Chamfer Distance loss [8] with L2 normalization (L2-norm CD), we further evaluate the L1-normalized Chamfer Distance loss (L1-norm CD), Earth Mover's Distance loss (EMD), and their combinations. As shown in the table 4, the original L2-norm CD loss performs better than all other compared losses. We denote the reconstructed and ground-truth point sets as $S_1$ and $S_2$. Compared to EMD loss that requires an optimal mapping for every point between $S_1$ and $S_2$, L2-norm CD loss only optimizes the separate pair-wise distances and is thus more robust to the variation of 3D structures. Compared to L1-norm CD loss, L2 norm of Euclidean Distances can better depict spatial distribution and pay more attention to the far away points.

## References

[1] Mohamed Afham, Isuru Dissanayake, Dinithi Dissanayake, Amaya Dharmasiri, Kanchana Thilakarathna, and Ranga Rodrigo. Crosspoint: Self-supervised cross-modal contrastive learning for 3d point cloud understanding. *arXiv preprint arXiv:2203.00680*, 2022. 2, 3

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 1

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 1

[4] 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. 2, 3

[5] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 3

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1

[7] 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, 4

[8] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017. 5

[9] Peng Gao, Minghang Zheng, Xiaogang Wang, Jifeng Dai, and Hongsheng Li. Fast convergence of detr with spatially modulated co-attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3621–3630, 2021. 1

[10] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R Martin, and Shi-Min Hu. Pct: Point cloud transformer. *Computational Visual Media*, 7(2):187–199, 2021. 1

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

[12] Kunchang Li, Yali Wang, Junhao Zhang, Peng Gao, Guanglu Song, Yu Liu, Hongsheng Li, and Yu Qiao. Uniformer: Unifying convolution and self-attention for visual recognition. *arXiv preprint arXiv:2201.09450*, 2022. 1

[13] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 1

[14] Mingyuan Mao, Renrui Zhang, Honghui Zheng, Teli Ma, Yan Peng, Errui Ding, Baochang Zhang, Shumin Han, et al. Dual-stream network for visual recognition. *Advances in Neural Information Processing Systems*, 34, 2021. 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 (ICCV)*, pages 2906–2917, October 2021. 1, 3

[16] Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. 2018. 1

[17] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 1

[18] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. 1

[19] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1588–1597, 2019. 2, 3

[20] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 2, 3

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1

[22] 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, 3

[23] 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. 2, 3

[24] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34, 2021. 1

[25] Li Yi, Vladimir G Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (ToG)*, 35(6):1–12, 2016. 2, 4

[26] Xumin Yu, Yongming Rao, Ziyi Wang, Zuyan Liu, Jiwen Lu, and Jie Zhou. Pointr: Diverse point cloud completion with geometry-aware transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12498–12507, 2021. 1

[27] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. *arXiv preprint arXiv:2111.14819*, 2021. 1, 3

[28] Renrui Zhang, Han Qiu, Tai Wang, Xuanzhuo Xu, Ziyu Guo, Yu Qiao, Peng Gao, and Hongsheng Li. Monodetr: Depth-aware transformer for monocular 3d object detection. *arXiv preprint arXiv:2203.13310*, 2022. 1

[29] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16259–16268, 2021. 1

[30] Minghang Zheng, Peng Gao, Renrui Zhang, Kunchang Li, Xiaogang Wang, Hongsheng Li, and Hao Dong. End-to-end object detection with adaptive clustering transformer. *arXiv preprint arXiv:2011.09315*, 2020. 1

[31] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 1