

Pointing3D: A Benchmark for 3D Object Referral via Pointing Gestures

Supplementary Material

1 Implementation Details

The pointing head is implemented as a three-layer MLP with a hidden dimension of 256 and ReLU activations. We train the model for 30 epochs using the AdamW optimizer [1] and a one-cycle learning rate schedule [2], with a peak learning rate of $1e-4$. Each batch contains 64 samples. To improve generalization, we apply random scaling and rotations around the z-axis as data augmentations.

We use MinkUNet Res16UNet34C [3] as the 3D backbone and extract feature maps from all five resolution levels, with channel dimensions of (96, 96, 128, 256, 256). The transformer decoder consists of 12 layers, each comprising a mask prediction and a query refinement module, with a hidden dimensionality of 128. We train the model for 300 epochs using the AdamW optimizer [1] and a one-cycle learning rate schedule [2], with a peak learning rate of $2e-3$. Training with a 2 cm voxel size takes 48 hours on an NVIDIA 4090 GPU. To enhance robustness, we apply data augmentations including horizontal flipping, elastic distortion, random scaling, and z-axis rotations. Color augmentations consist of jittering, brightness, and contrast adjustments. For faster convergence, we synthesize 100 pointing gestures per point cloud during training.

2 DP Dataset Filtering

We use the DP Dataset [4] to train our pointing head. As shown in Fig. 1, in this dataset, participants point at markers with known 3D positions, providing accurate ground truth for pointing direction annotations. The dataset includes annotated start and end frames of each pointing sequence; however, it also labels intermediate transition frames, such as when the arm is still being raised, as part of the gesture. For example, in Fig. 1 (left), a frame showing the participant in the process of lifting their arm is still labeled as pointing, even though the gesture is not yet stable as in Fig. 1 (right). This labeling approach is compatible with their proposed DeePoint model, which predicts a single pointing direction over the entire interval. In contrast, our method operates at the frame level and requires precise frame-wise annotations. To address this, we first estimate shoulder and hand joint positions using a state-of-the-art human pose estimation model [5]. We compute the direction vector from the shoulder to the hand and compare it with the ground truth pointing direction. Frames with an angular deviation greater than 20 degrees are flagged as transitional. We then manually review these cases to produce a cleaner, filtered subset of the DP dataset.



(a) Transitional pointing frame

(b) Final pointing frame

Figure 1: DP Dataset samples.

32 **References**

- 33 [1] I. Loshchilov and F. Hutter. Decoupled Weight Decay Regularization. *International Conference*
34 *on Learning Representations*, 2019.
- 35 [2] L. N. Smith and N. Topin. Super-Convergence: Very Fast Training of Neural Networks Us-
36 ing Large Learning Rates. In *Artificial Intelligence and Machine Learning for Multi-Domain*
37 *Operations Applications*, 2019.
- 38 [3] C. Choy, J. Gwak, and S. Savarese. 4D Spatio-Temporal ConvNets: Minkowski Convolutional
39 Neural Networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.
- 40 [4] S. Nakamura, Y. Kawanishi, S. Nobuhara, and K. Nishino. DeePoint: Visual Pointing Recog-
41 nition and Direction Estimation. In *IEEE/CVF International Conference on Computer Vision*,
42 2023.
- 43 [5] I. Sáráandi and G. Pons-Moll. Neural Localizer Fields for Continuous 3D Human Pose and Shape
44 Estimation. *Neural Information Processing Systems*, 2024.