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

Mert Arslanoglu[1,⋆] **Kadir Yilmaz**[1,⋆] **Cemhan Kaan Özaltan**[1] **Timm Linder**[2] **Bastian Leibe**[1]
[1]RWTH Aachen University, Germany [2]Bosch Center for AI, Germany
⋆Equal contribution
https://vision.rwth-aachen.de/Pointing3D

Abstract: Pointing gestures provide a natural and efficient way to communicate spatial information in human-machine interaction, yet their potential for 3D object referral remains largely underexplored. To fill this gap, we introduce the task of pointing-based 3D segmentation. In this task, given an image of a person pointing at an object and the 3D point cloud of the environment, the goal is to predict the 3D segmentation mask of the referred object. To enable the standardized evaluation of this task, we introduce POINTR3D, a curated dataset of 65,000 frames captured with three cameras across four indoor scenes, featuring diverse pointing scenarios. Each frame is annotated with the information of the active hand, the corresponding object ID, and the 3D segmentation mask of the object. To showcase the application of the proposed dataset, we further introduce Pointing3D, a transformer-based architecture that predicts the pointing direction from RGB images and uses this prediction as a prompt to segment the referred object in the 3D point cloud. Experimental results show that Pointing3D outperforms other strong baselines we introduce and lays the groundwork for future research. The dataset, source code, and evaluation tools will be made publicly available to support further research in this area, enabling a natural human-machine interaction.

Keywords: Object Referral, Pointing Gesture, 3D Segmentation

1 Introduction

Robots are increasingly deployed in complex settings, where their ability to interpret human intent accurately is crucial for effective human-robot interaction (HRI) [1, 2, 3]. A central component of this interaction is object referral [4, 5], the process by which a human user directs the robot's attention to a specific object within a shared environment. This capability is critical for a wide range of applications—from industrial tasks like assembly to everyday household chores—where intuitive and rapid communication of intent can significantly enhance operational efficiency and safety. Conventional approaches to object referral typically rely on explicit spatial inputs, such as metric-space coordinates [6, 7] (e.g., "move to $x = 22.3$, $y = -10.1$"), often supplied via 2D map interfaces or external tracking systems. While effective in structured environments, these methods suffer from several limitations: they are device-dependent, unintuitive for untrained users, and decoupled from natural human communication modalities [8, 3]. This raises a key research question: Can we enable natural and efficient object referral in 3D space using only minimal, uninstrumented human input?

Among natural interaction modalities, pointing gestures emerge as a compelling candidate [9, 10]. They are universally understood, require no additional hardware beyond visual perception, and seamlessly integrate into everyday human behavior. Although numerous methods estimate pointing direction [11, 12], they do not focus on the semantic understanding of the scene, making them unable to resolve the type and the extent of the object being referred to. This limits their effective-

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

ness in HRI tasks such as manipulation, inspection, or object handover. Thus, the use of pointing gestures for 3D object referral remains underexplored.

To achieve the full potential of pointing gestures in HRI, we propose a new task: *pointing-based 3D object segmentation*. Unlike prior work that solely estimates the pointing direction, this task aims to infer the full 3D extent of the referred object by jointly reasoning over human gestures and the semantic structure of the scene. Specifically, given an image showing a person pointing and the corresponding 3D environment, the objective is to predict a 3D segmentation mask for the intended object. However, upon reviewing the current literature, we observe that no existing dataset supports this task. While prior 3D datasets offer extensive instance segmentation annotations in indoor environments [13, 14, 15, 16], they lack any human interaction, including pointing behavior. Conversely, datasets that study pointing gestures [11, 17] generally lack a 3D representation of the environment and do not provide segmentation masks of the referred objects. To bridge this gap, we introduce POINTR3D, a novel dataset that comprises ≈65,000 curated samples capturing pointing gestures within annotated 3D scenes. Each data sample contains an RGB image of pointing and the corresponding 3D point cloud, along with the active hand and the object segmentation annotations, enabling the study and standardized evaluation of this new task.

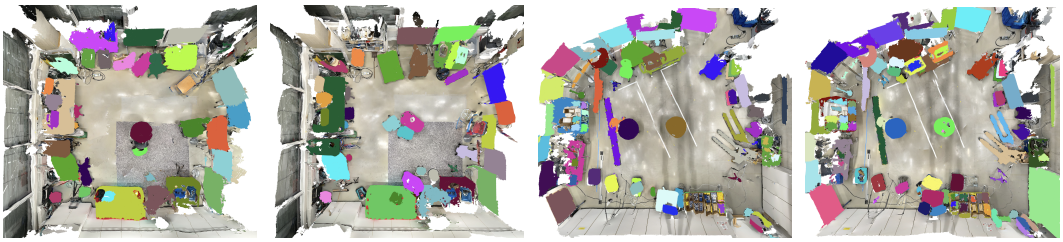
Building on the proposed task and dataset, we introduce Pointing3D, a two-stage model that combines pointing direction estimation with 3D segmentation. The first stage leverages skeletal joint positions extracted from a state-of-the-art human pose estimator [18] to predict the pointing vector. This estimate is then used as a prompt in the second stage, where a transformer-based segmentation module [19, 20, 21] infers the 3D mask of the referred object within the point cloud. Empirical evaluations demonstrate that Pointing3D outperforms competitive proposed baselines, establishing a strong foundation for future research on pointing-based object referral in 3D environments.

Overall, our contributions are as follows:

- We introduce the task of pointing-based 3D object segmentation, which goes beyond estimating pointing direction to infer the full 3D extent of the referred object by jointly reasoning over human gestures and scene semantics.
- We present POINTR3D, the first dataset designed for this task, containing multi-view recordings of humans pointing in real-world indoor environments, along with 3D point clouds and instance segmentation masks of the referred objects.
- We propose Pointing3D, a two-stage model that first estimates the pointing direction using human skeletal joints and then performs promptable 3D segmentation conditioned on this estimate.

2 Related Work

Pointing Gestures. Early studies on pointing primarily focus on gesture recognition rather than estimating pointing directions, often targeting AR/VR applications using egocentric views [22], multi-camera setups [23], or wearable IMU devices [24]. Subsequent works extend pointing recognition to 3D pointing direction estimation, either using RGB-D cameras in tabletop settings [17], or relying on robust 3D hand detection approaches [12] for 3D inference. DeePoint [11] made a significant advance by introducing a large-scale dataset and a transformer-based model for estimating 3D pointing directions from videos. However, all these methods focus solely on direction estimation without identifying the referred object. More recent approaches explore combining pointing with additional modalities. Nakagawa *et al.* [25] aligns pointing gestures with speech to recognize the pointing gesture, while Deguchi *et al.* [26] and VGPN [27] leverage pointing mainly to support language-driven navigation tasks. Object referral methods like Exophora [28], YouRefIt [29], and GIRAF [30] integrate pointing gestures and language, often assuming a prior map of the environment or limiting the setting to tabletop scenes. Similarly, Constantin *et al.* [31] combine pointing recognition and natural language understanding and resolve ambiguities that might arise with an LLM. These works demonstrate the value of pointing gestures in object referral, but often depend on strong language cues, prior exploration, or restricted setups. Other works [32, 33] rely solely on



(a) Validation scene 1 (b) Validation scene 2 (c) Test scene 1 (d) Test scene 2

Figure 1: POINTR3D scene meshes with segmentation masks overlaid.

language for assigning semantic navigation tasks to a robot. Without gestures, these approaches can fail to differentiate between multiple identical objects, requiring lengthy and complex descriptions.

In contrast, our work focuses on 3D object referral using only natural pointing gestures, without assuming prior knowledge about the environment or requiring verbal input. The most closely related work to ours is ScanERU [34], which tackles 3D visual grounding by integrating gestures into point cloud-based object referral. They introduce a semi-synthetic dataset that combines synthetic humans, real-world 3D point clouds, and textual descriptions. However, their focus remains on multimodal fusion and uses synthetic data for evaluations, while our work targets purely pointing-driven object referral in 3D scenes on real data.

3D Segmentation. The field of 3D segmentation has seen rapid progress, driven largely by the introduction of large-scale annotated datasets such as ScanNet [14] and S3DIS [13]. Building on advances in 2D vision, particularly the success of mask transformers [19] for image segmentation, researchers adapted this paradigm to 3D segmentation [20, 35], achieving impressive performance by directly operating on point clouds. Inspired by the success of interactive segmentation approaches in image segmentation [36], subsequent efforts [21, 37] extend mask transformers to interactive 3D segmentation, leveraging sparse user inputs—typically in the form of mouse clicks—to guide and refine instance masks. However, such interactive methods, including the earlier SemanticPaint [38] that additionally integrates gestures and language, are primarily designed for annotation purposes, relying on iterative corrections through multiple user interactions.

In contrast, robotics applications demand one-shot, efficient interpretation of human intent from minimal input. Addressing this gap, Pointing3D predicts fine-grained 3D instance masks from a single pointing gesture, and operates in real time, making it well-suited for practical human-robot collaboration.

3 POINTR3D Dataset

While datasets for pointing direction estimation [11, 12] and synthetic 3D object referral [34] exist, none, to the best of our knowledge, simultaneously provide (i) images of humans pointing at objects, (ii) corresponding 3D point clouds of the environment, and (iii) 3D segmentation masks of the referred objects. We fill this gap by introducing POINTR3D, a novel dataset designed to enable the development and standardized evaluation of pointing-based 3D object segmentation methods.

Data Acquisition. We collect data of indoor scenes in two distinct locations, with each scene recorded under two configurations featuring different object arrangements, resulting in four unique environmental setups, shown in Figure 1. The scenes include a diverse set of objects at varying sizes, such as a mobile manipulator with a robotic arm, a quadruped robot, a conveyor, a forklift, a pushcart, a power supply, a desktop computer, a backpack, a trash can, cardboard and plastic trays, beverage bottles and spray cans, a watering can, a potted plant, a sofa and multiple chairs and tables. At the beginning of each session, the 3D geometry of each scene was captured using an iPhone 14 Pro [16], ensuring a clean, static reconstruction. During this process, no humans were present in the scene to minimize domain gaps with common 3D segmentation datasets [14, 15, 16].

(a) User pointing at quadruped robot

(b) User pointing at gray stool

Figure 2: Two examples from POINTR3D dataset showing RGB images with pointing gestures (left) and their corresponding 3D segmentation masks (right).

Following 3D scene reconstruction, participants were equipped with only a tiny clip-on microphone, but no other measurement devices (such as IMU suits) to reflect realistic use cases. Then, they were allowed to freely point at objects and provide verbal descriptions of the objects without any time constraints. Since our dataset focuses on 3D object referral rather than precise pointing direction estimation, we do not require participants to point at markers at known locations as in [11]. Three synchronized Azure Kinect cameras, placed at varying angles and heights to enhance viewpoint diversity, recorded RGB videos at 2048×1536 resolution and 15 FPS. All cameras were intrinsically and extrinsically calibrated and registered to the reconstructed point cloud via manually provided pixel-to-point correspondences. While participants pointed at objects, synchronized audio recordings have been collected to assist with the disambiguation of pointed objects during the annotation process. The resulting raw data, including both pointing and non-pointing frames, consists of 254,073 images, camera calibration parameters, synchronized audio, and four point cloud reconstructions of the scenes.

Data Postprocessing. To eliminate ambiguous cases, we discard frames in which the participant faces away from the camera, resulting in occlusion of the pointing hand. We first extract skeletal joint positions using a state-of-the-art human pose estimation model [18], and then assess whether the hand joints are occluded by the torso from the camera's perspective. Frames identified as occluded are automatically flagged and subsequently verified through manual inspection, resulting in the removal of approximately 13% of the data. Finally, we partition the data based on recording locations to prevent information leakage between the validation and testing sets, resulting in 34,795 frames in the validation split and 30,108 in the test split.

Data Annotation. The annotation process consists of two main components: 3D object segmentation in point clouds and temporal labeling of pointing actions in videos. For 3D segmentation, we adopt Agile3D [21], an interactive annotation tool that allows annotators to iteratively refine segmentation masks through mouse clicks. This enables efficient and accurate labeling with minimal manual effort. Over 24 man-hours, a total of 243 distinct object instances have been annotated by the authors of this paper. For temporal annotation of pointing actions, we first align the video recordings with transcriptions obtained from synchronized audio using WhisperX [39]. Annotators then utilized both the video and the aligned transcript to identify the time intervals during which pointing gestures occur, the active pointing hand (left/right), and the corresponding object ID. This temporal annotation effort required an additional 43 man-hours. In total, the dataset includes 832 annotated pointing actions, with an average duration of 5.20 seconds per gesture, distributed across 64,903 video frames performed by seven different participants, providing diversity in both appearance and motion patterns. Two representative examples are shown in Figure 2.

4 Method

Building on recent advances in 3D interactive segmentation [21, 37] and human pose estimation [18, 40], we introduce Pointing3D, a unified framework for pointing-based 3D object segmentation. Given (i) a single RGB image depicting a user in a pointing pose, (ii) a colored point cloud of

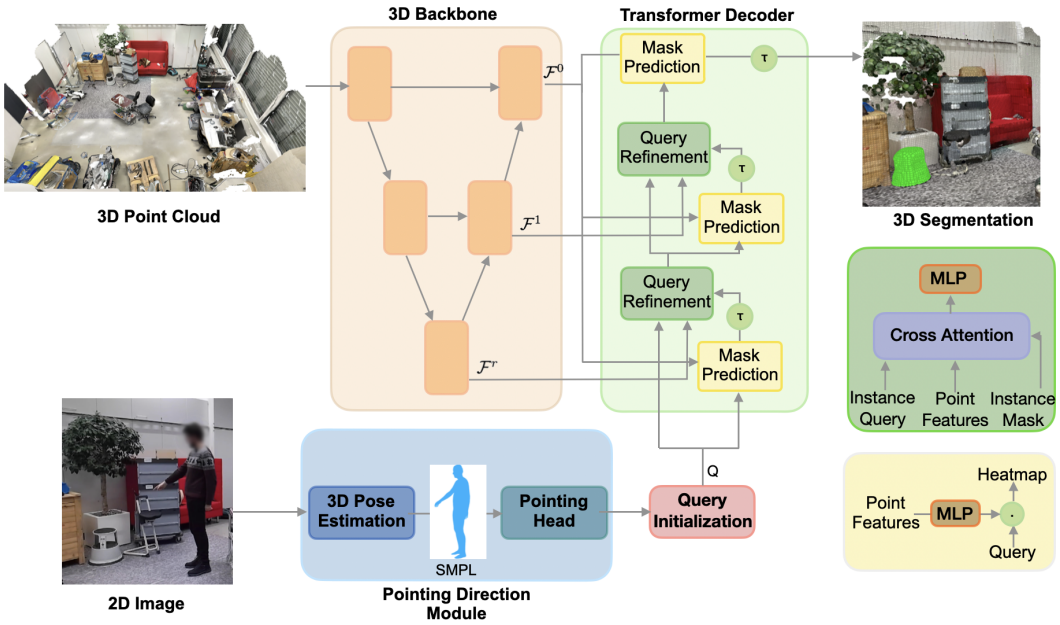


Figure 3: **Pointing3D model.** We extract multi-scale point cloud features $\mathcal{F}^r$ via the feature backbone ☐ and concurrently estimate the 3D skeletal joint positions $\mathbf{J}$ from an input image. Using these joint positions, the pointing direction $\mathbf{d}_{\text{pred}}$ is estimated and used for initializing ☐ the segmentation query $\mathbf{Q}$. Then, the transformer decoder ☐ iteratively refines this query with point features $\mathcal{F}^r$, producing pointwise heatmaps H and yielding the final object segmentation.

the scene, and (iii) camera intrinsics and extrinsic transform between point cloud and RGB image, our goal is to predict the 3D instance mask of the object being referred to by the user. Importantly, the RGB image may be captured by a (mobile) robot-mounted camera and does not necessarily contain the target object due to occlusions or a non-overlapping field of view.

Our framework consists of two main components: (1) a *Pointing Direction Module* that estimates the 3D pointing vector from the RGB image, and (2) a *3D Instance Segmentation Module* that uses this direction as a prompt to segment the target object in 3D space. An overview of our model architecture is shown in Figure 3.

4.1 Pointing Direction Module

The goal of this module (Fig. 3, ☐) is to estimate the position of the active hand and the pointing direction from a single RGB image. To supervise this module during training, we utilize the DP dataset [11], which provides images of individuals pointing at known 3D marker locations in the environment. We first estimate absolute, metric-scale 3D human poses from the input image using a state-of-the-art method [18], which returns both 3D joint positions $\mathbf{J} \in \mathbb{R}^{24\times 3}$ and the SMPL [41] mesh vertices $\mathbf{V} \in \mathbb{R}^{6890\times 3}$. Let $\mathbf{j}_{\text{hand}} \in \mathbb{R}^3$ denote the position of the active hand joint, and $\mathbf{m} \in \mathbb{R}^3$ be the known 3D marker location corresponding to the pointing gesture. The ground truth pointing direction, $\mathbf{d}_{\text{gt}} \in \mathbb{R}^3$, is defined as the normalized vector from the hand joint to the marker:

$$\mathbf{d}_{\text{gt}} = \frac{\mathbf{m} - \mathbf{j}_{\text{hand}}}{\|\mathbf{m} - \mathbf{j}_{\text{hand}}\|}. \tag{1}$$

Then, we regress the pointing direction $\mathbf{d}_{\text{pred}}$ from the SMPL mesh vertices $\mathbf{V}$ using a multilayer perceptron (MLP). The model is trained using a combined cosine similarity and L1 loss:

$$\mathcal{L}(\mathbf{d}_{\text{pred}}, \mathbf{d}_{\text{gt}}) = \left(1 - \frac{\mathbf{d}_{\text{pred}} \cdot \mathbf{d}_{\text{gt}}}{\|\mathbf{d}_{\text{pred}}\|\|\mathbf{d}_{\text{gt}}\|}\right) + \|\mathbf{d}_{\text{pred}} - \mathbf{d}_{\text{gt}}\|_1 \tag{2}$$

During evaluation on the POINTR3D dataset, we use the human pose estimation model and the trained MLP to predict the active hand joint $\mathbf{j}_{\text{hand}}$ and the pointing direction $\mathbf{d}_{\text{pred}}$ from input images.

4.2 3D Instance Segmentation Module

This module takes as input the active hand joint $\mathbf{j}_{\text{hand}}$, the predicted pointing direction $\mathbf{d}_{\text{pred}}$, and the 3D point cloud of the environment. The objective is to predict a binary mask that segments the object referred by the user. At its core, the module uses a transformer-based architecture that leverages the pointing gesture as a prompt to guide segmentation. The process begins by encoding the point cloud with a 3D backbone that extracts multi-scale feature representations. A segmentation query—representing the referred object—is then initialized using the pointing ray, capturing both geometric locality and semantic cues. This query is iteratively refined to better represent the object of interest using the extracted point features, and the final refined query is used for segmentation.

3D Backbone. (Fig. 3, ▢) The input colored point cloud $\mathcal{P} \in \mathbb{R}^{N\times 6}$ is first quantized into a sparse voxel grid $\mathcal{V} \in \mathbb{R}^{M\times 6}$ for memory-efficient processing with sparse convolutions. A 3D U-Net built on sparse convolutional networks [42, 43] extracts multi-scale point features $\mathcal{F}^r \in \mathbb{R}^{M\times D^r}$, capturing both local geometry and semantic context. These features are later used by the transformer decoder to refine the segmentation query and predict the segmentation mask.

Query Initialization. (Fig. 3, ▢) To initialize the segmentation query $Q^0 \in \mathbb{R}^{D^0}$, we cast a ray from the estimated hand position $\mathbf{j}_{\text{hand}}$ along the predicted pointing direction $\mathbf{d}_{\text{pred}}$. Points within a fixed angular threshold θ from the ray define a conical region of interest. The closest point to the hand within this region, denoted as $\mathbf{p}_{\text{target}}$, is selected and assumed to lie on the referred object. This target point is used to initialize the segmentation query Q^0 by summing its Fourier positional encoding [44] $\text{PE}_{\text{target}}$ and feature embedding $\mathcal{F}^0_{\text{target}}$ extracted from the 3D backbone.

$$Q^0 = \text{PE}_{\text{target}} + \mathcal{F}^0_{\text{target}} \tag{3}$$

This initial query Q^0 encodes both geometric and semantic context, serving as the input to the transformer decoder and guiding attention toward the referred object for subsequent instance segmentation.

Mask Prediction. (Fig. 3, ▢) At each transformer decoder layer $l = 0, \ldots, L-1$, the segmentation query Q^l is used to predict a binary mask $\mathbf{M^l} \in \{0,1\}^N$, indicating whether a point belongs to the referred object. To achieve this, per-point features $\mathcal{F}^0$ are projected into the dimensionality of Q with a linear layer. The instance heatmap $H^l \in \mathbb{R}^M$ is then calculated as follows:

$$H^l = \sigma(\mathcal{F}^0 \cdot Q^l) \tag{4}$$

where σ represents the sigmoid function and $\cdot$ is the dot product operation. Points with features similar to the query will have higher values in the heatmap, indicating a higher likelihood of belonging to the referred object. During training, the predicted mask is supervised with binary cross-entropy loss and dice loss against the ground truth instance masks.

Query Refinement. (Fig. 3, ▢) The segmentation query Q^l is iteratively refined through transformer decoder layers, each integrating contextual information from the 3D scene via cross-attention to multi-scale point features $\mathcal{F}^r$. During attention, the query can only attend to points predicted as part of the object in the previous mask prediction step, guiding it toward relevant regions. The query is then passed through a feed-forward network with layer normalization and residual connections to produce the refined query for the next stage. Unlike traditional mask transformer approaches [19, 20], we do not apply self-attention between queries, as only a single segmentation query is used.

$$Q^{l+1} = \text{FFN}\left(\text{Norm}\left(Q^l + \text{CrossAttention}(Q^l, \mathcal{F}^r_{\text{object}})\right)\right) \tag{5}$$

Inference. The final segmentation query Q^{L-1} produced by the last query refinement layer is used to predict the 3D object mask by taking the dot product with the point features $\mathcal{F}^0$ and applying a threshold to separate the object foreground from the background.

5 Experiments

Training Datasets. To train the pointing direction estimation module, we leverage the DP dataset [11], which contains multi-view video sequences of individuals pointing at calibrated 3D

Table 1: Pointing accuracy, angular deviation from target centroid (AD_c) and from the nearest target point (AD_n), and frames per second comparison on validation and test splits of POINTR3D.

Method	Validation			Test			Overall
	Acc.↑	AD_c↓	AD_n↓	Acc.↑	AD_c↓	AD_n↓	FPS↑
Deepoint [11]	34.9%	28.9°	13.6°	30.1%	23.4°	14.5°	0.5
Elbow to Hand	53.2%	22.3°	7.4°	37.0%	17.4°	8.9°	**41**
Head to Hand	54.1%	20.7°	7.2°	50.5%	14.4°	7.4°	**41**
Shoulder to Hand	66.5%	18.5°	4.8°	56.8%	13.3°	5.6°	**41**
Pointing3D (ours)	**72.2%**	**17.4°**	**4.4°**	**64.6%**	**12.4°**	**4.5°**	35

markers within indoor environments. Captured from 15 distinct viewpoints with varying elevations and orientations, it provides precise 3D target locations and temporal annotations for pointing intervals. However, the temporal intervals often include transitional frames (e.g., during arm lifting) where the gesture does not align with the intended target. We address this by filtering out non-representative frames, as detailed in the supplementary material.

For training the 3D instance segmentation module, we use the ScanNet200 dataset [15], which comprises 1,513 colored point clouds from diverse indoor scenes, annotated with instance-level masks across 200 object categories. As ScanNet200 does not include humans or pointing gestures, we simulate pointing by randomly sampling locations on object instances and treating the corresponding segmentation masks as targets. This synthetic supervision enables independent training of the segmentation module while ensuring exposure to a wide variety of environments and object types.

Evaluation. Both the pointing direction estimation and the 3D instance segmentation modules are evaluated on the POINTR3D dataset. To assess pointing direction estimation, we report two metrics. First, we measure pointing accuracy, which captures whether the predicted pointing vector intersects the ground-truth object mask. Second, we report angular deviation, using two variants: (i) deviation to the target object centroid (AD_c), and (ii) deviation to the nearest point on the target object (AD_n). These metrics together provide a comprehensive assessment of the precision of the estimated pointing direction. For segmentation performance, we compute the Intersection-over-Union (IoU_{mean}) between the predicted mask and the ground-truth mask of the pointed object. We also report the Intersection-over-Union separately for the accurate (IoU_+) and inaccurate pointing cases (IoU_-) to give a better insight. We use the validation set for hyperparameter selection, and the test set, whose labels remain hidden, for final evaluation, establishing a benchmark to enable fair comparisons across future methods. Notably, POINTR3D is not used during training, creating a **zero-shot setting** that allows us to assess how well the approach generalizes to unseen objects.

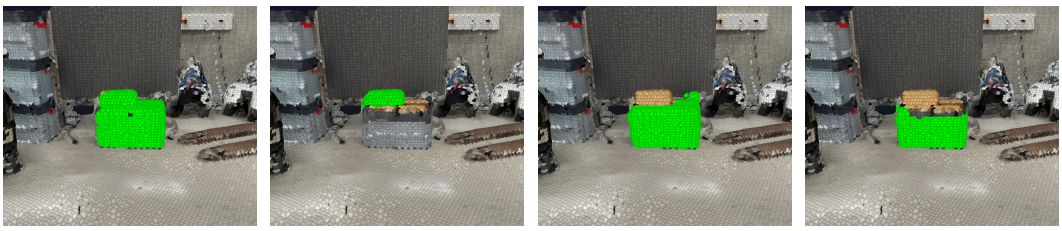
Pointing Direction Results. We compare Pointing3D against a range of baselines, including heuristic approaches commonly used in the literature [11, 12], such as vectors from the head to the hand or from the shoulder to the hand, which offer simple approximations of pointing direction. We also evaluate DeePoint [11], the current state-of-the-art method for pointing direction estimation. Table 1 presents the results on the validation and test splits of POINTR3D, using pointing accuracy and angular deviation metrics, along with frames per second. While heuristic methods perform surprisingly well, likely due to the robustness and generalization capabilities of the pose estimation backbone [18], Pointing3D consistently outperforms all baselines across metrics and splits. We attribute this to our model's ability to learn effectively from a large and diverse dataset. In contrast, DeePoint performs suboptimally in this zero-shot evaluation setting. Its reliance on image features, while effective within its training dataset, limits its generalization when applied to new scenes without fine-tuning. Moreover, the use of multi-frame (15 per prediction) processing results in high computational overhead, making it less suitable for real-time deployment scenarios.

Segmentation Results. To evaluate the effectiveness of Pointing3D, we compare it against several segmentation baselines in Table 2, all using the same pointing predictions from our model to isolate differences in segmentation performance. As a non-learned baseline, we use the predicted pointing location as a seed and apply a classical 3D region growing algorithm [45] to generate a segmentation

Table 2: Segmentation performance comparison on validation and test splits of the POINTR3D dataset across segmentation methods.

Method	Validation			Test			Overall
	$\text{IoU}_{\text{mean}} \uparrow$	$\text{IoU}_{+} \uparrow$	$\text{IoU}_{-} \uparrow$	$\text{IoU}_{\text{mean}} \uparrow$	$\text{IoU}_{+} \uparrow$	$\text{IoU}_{-} \uparrow$	FPS↑
Region Growing	12.2%	16.4%	1.3%	16.4%	23.5%	3.2%	5.1
Interactive 2D [36]	35.6%	48.7%	1.9%	33.4%	45.6%	10.7%	2.2
Non-interactive 3D [20]	36.1%	47.2%	**13.0%**	39.4%	51.9%	**16.2%**	**35**
Pointing3D (ours)	**43.9%**	**60.0%**	2.1%	**42.6%**	**59.6%**	11.1%	19

mask. This approach performs poorly, underscoring the need for learning-based methods that can capture semantic structure in complex scenes. Next, we evaluate a 2D interactive baseline using SAM [36]. An image of the scene is rendered from the mesh, with the virtual camera positioned at the active hand and oriented along the predicted pointing direction. The projected pointing location is then used to prompt SAM, and the resulting 2D mask is unprojected back into 3D. While conceptually appealing, this pipeline suffers from occlusions and projection errors. In contrast, Pointing3D operates directly in 3D, leading to more accurate segmentation. Finally, as a non-interactive 3D segmentation baseline, we adopt Mask3D [20] and perform instance segmentation on the point cloud independently of the pointing gesture. Then, the pointing ray is used to select the best object in terms of spatial proximity. However, if the corresponding object is not segmented correctly in the first place, the pointing cue cannot correct the error. Pointing3D outperforms this baseline as well, demonstrating the benefit of integrating pointing cues into the segmentation process.



(a) Interactive 2D (IoU = 44.1%) (b) Non-interactive 3D (IoU = 1.2%) (c) Pointing3D (ours) (IoU = 66.3%) (d) Ground truth (IoU = 100%)

Figure 4: Qualitative comparison of predicted segmentation masks of a gray box by various models and the ground truth. The interactive 2D baseline [36] tends to overestimate the mask while the non-interactive 3D baseline [20] tends to underestimate.

Qualitative Results. Figure 4 presents a comparison of segmentation performance across different models. The SAM [36]-based interactive 2D baseline segments both the box and its contents as a single object, failing to distinguish between them. The Mask3D baseline does not recognize the gray box as a distinct object, resulting in a low segmentation score despite being spatially close. Pointing3D produces a reasonable segmentation, although the object boundaries are not perfect.

6 Conclusion

While pointing gestures are a natural way to refer to objects, their use for 3D object referral remains largely underexplored in human-robot interaction. To address this gap, we introduce a new task and benchmark, POINTR3D, for evaluating pointing-based object referral with 3D segmentation in indoor scenes. We provide several strong baselines that lay the groundwork for future research, and propose our method Pointing3D, a unified model that combines human pose estimation with interactive 3D instance segmentation. Our zero-shot evaluations demonstrate that Pointing3D outperforms both non-interactive 3D and interactive 2D baselines, highlighting the benefits of interactive segmentation and operating directly in 3D. We believe that this work lays a solid ground for the development and evaluation of future research on pointing-based HRI.

7 Limitations

While Pointing3D demonstrates promising results for pointing-based 3D object segmentation, it has several limitations. To minimize the domain gap between POINTR3D and other commonly used 3D indoor segmentation datasets, our framework assumes a static point cloud of the environment. This limits its application to a static environment or requires continuous reconstruction of the scene using SLAM. Besides, our pipeline assumes the presence of only a single user in the scene at a time. Although the human pose estimation module can predict multiple poses, POINTR3D is designed for single-user interactions. This limits its applications to environments containing one person at a time. Also, it is important to remark the fact that pointing behavior can vary significantly across individuals due to differences in age, gender, cultural background, and situational context. A larger dataset with greater participant diversity would likely further improve generalization.

Acknowledgments

This project was partially funded by the project "Context Understanding for Autonomous Systems" by Robert Bosch GmbH. Computations were performed with computing resources granted by RWTH Aachen under project rwth1730.

References

[1] T. B. Sheridan. Human-Robot Interaction: Status and Challenges. *Human Factors The Journal of the Human Factors and Ergonomics Society*, 2016.

[2] M. Higger, P. Rygina, L. Daigler, L. F. Bezerra, Z. Han, and T. Williams. Toward open-world human-robot interaction: What types of gestures are used in task-based open-world referential communication? In *The 27th Workshop on the Semantics and Pragmatics of Dialogue (SemDial)*, 2023.

[3] S. S. Rautaray and A. Agrawal. Vision Based Hand Gesture Recognition for Human Computer Interaction: a Survey. *Artificial intelligence review*, 2015.

[4] P. Gao, A. Jaafar, B. Reily, C. Reardon, and H. Zhang. Compositional Zero-Shot Learning for Attribute-Based Object Reference in Human-Robot Interaction. In *Conference on Robot Learning (Workshops)*, 2023.

[5] A. B. Vasudevan, D. Dai, and L. Van Gool. Object Referring in Visual Scene with Spoken Language. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, 2018.

[6] K. Mahadevan, B. Lewis, J. Li, B. Mutlu, A. Tang, and T. Grossman. ImageInThat: Manipulating Images to Convey User Instructions to Robots. *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2025.

[7] A. Mani, N. Yoo, W. Hinthorn, and O. Russakovsky. Point and Ask: Incorporating Pointing into Visual Question Answering. *arXiv preprint arXiv:2011.13681*, 2020.

[8] A. Carfì and F. Mastrogiovanni. Gesture-based Human–machine Interaction: Taxonomy, Problem Definition, and Analysis. *IEEE Transactions on Cybernetics*, 2021.

[9] E. Bamani, E. Nissinman, L. Koenigsberg, I. Meir, Y. Matalon, and A. Sintov. Recognition and Estimation of Human Finger Pointing with an RGB Camera for Robot Directive. *arXiv preprint arXiv:2307.02949*, 2023.

[10] D. Jirak, D. Biertimpel, M. Kerzel, and S. Wermter. Solving visual Object Ambiguities when Pointing: An Unsupervised Learning Approach. *Neural Computing and Applications*, 2021.

[11] S. Nakamura, Y. Kawanishi, S. Nobuhara, and K. Nishino. DeePoint: Visual Pointing Recognition and Direction Estimation. In *IEEE/CVF International Conference on Computer Vision*, 2023.

[12] M. Ürkmez and H. I. Bozma. Detecting 3D Hand Pointing Direction from RGB-D Data in Wide-Ranging HRI Scenarios. In *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2022.

[13] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese. 3D Semantic Parsing of Large-Scale Indoor Spaces. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

[14] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

[15] D. Rozenberszki, O. Litany, and A. Dai. Language-Grounded Indoor 3D Semantic Segmentation in the Wild. In *European Conference on Computer Vision*, 2022.

[16] C. Yeshwanth, Y.-C. Liu, M. Nießner, and A. Dai. ScanNet++: A High-Fidelity Dataset of 3D Indoor Scenes. In *IEEE/CVF International Conference on Computer Vision*, 2023.

[17] D. Shukla, O. Erkent, and J. Piater. Probabilistic Detection of Pointing Directions for Human-Robot Interaction. In *International Conference on Digital Image Computing Techniques and Applications*, 2015.

[18] I. Sárándi and G. Pons-Moll. Neural Localizer Fields for Continuous 3D Human Pose and Shape Estimation. *Neural Information Processing Systems*, 2024.

[19] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar. Masked-attention Mask Transformer for Universal Image Segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[20] J. Schult, F. Engelmann, A. Hermans, O. Litany, S. Tang, and B. Leibe. Mask3D: Mask Transformer for 3D Semantic Instance Segmentation. In *IEEE International Conference on Robotics and Automation*, 2023.

[21] Y. Yue, S. Mahadevan, J. Schult, F. Engelmann, B. Leibe, K. Schindler, and T. Kontogianni. AGILE3D: Attention Guided Interactive Multi-object 3D Segmentation. In *International Conference on Learning Representations*, 2024.

[22] Y. Huang, X. Liu, X. Zhang, and L. Jin. A Pointing Gesture Based Egocentric Interaction System: Dataset, Approach and Application. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2016.

[23] D. Fujita and T. Komuro. Three-Dimensional Hand Pointing Recognition Using Two Cameras by Interpolation and Integration of Classification Scores. In *European Conference on Computer Vision*, 2014.

[24] D. Broggini, B. Gromov, A. Giusti, and L. Gambardella. Learning to Detect Pointing Gestures from Wearable IMUs. In *AAAI Conference on Artificial Intelligence*, 2018.

[25] H. Nakagawa, S. Hasegawa, Y. Hagiwara, A. Taniguchi, and T. Taniguchi. Pointing Frame Estimation With Audio-Visual Time Series Data for Daily Life Service Robots. In *IEEE International Conference on Systems, Man, and Cybernetics*, 2024.

[26] H. Deguchi, S. Taguchi, K. Shibata, and S. Koide. Enhanced Robot Navigation with Human Geometric Instruction. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2023.

[27] J. Hu, Z. Jiang, X. Ding, T. Mu, and P. Hall. VGPN: Voice-Guided Pointing Robot Navigation for Humans. In *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2018.

[28] A. Oyama, S. Hasegawa, H. Nakagawa, A. Taniguchi, Y. Hagiwara, and T. Taniguchi. Exophora Resolution of Linguistic Instructions with a Demonstrative based on Real-World Multimodal Information. In *International Conference on Robot and Human Interactive Communication*, 2023.

[29] Y. Chen, Q. Li, D. Kong, Y. L. Kei, S.-C. Zhu, T. Gao, Y. Zhu, and S. Huang. YouRefIt: Embodied Reference Understanding with Language and Gesture. In *IEEE/CVF International Conference on Computer Vision*, 2021.

[30] L.-H. Lin, Y. Cui, Y. Hao, F. Xia, and D. Sadigh. Gesture-Informed Robot Assistance via Foundation Models. In *Conference on Robot Learning*, 2023.

[31] S. Constantin, F. I. Eyiokur, D. Yaman, L. Bärmann, and A. Waibel. Interactive Multimodal Robot Dialog Using Pointing Gesture Recognition. In *European Conference on Computer Vision*, 2022.

[32] C. Huang, O. Mees, A. Zeng, and W. Burgard. Visual Language Maps for Robot Navigation. In *IEEE International Conference on Robotics and Automation*, 2023.

[33] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman, et al. Do As I Can, Not As I Say: Grounding Language in Robotic Affordances. In *Conference on Robot Learning*, 2022.

[34] Z. Lu, Y. Pei, G. Wang, P. Li, Y. Yang, Y. Lei, and H. T. Shen. ScanERU: Interactive 3D Visual Grounding based on Embodied Reference Understanding. In *AAAI Conference on Artificial Intelligence*, 2024.

[35] K. Yilmaz, J. Schult, A. Nekrasov, and B. Leibe. Mask4Former: Mask Transformer for 4D Panoptic Segmentation. In *IEEE International Conference on Robotics and Automation*, 2024.

[36] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al. Segment Anything. In *IEEE/CVF International Conference on Computer Vision*, 2023.

[37] I. Fradlin, I. E. Zulfikar, K. Yilmaz, T. Kontogianni, and B. Leibe. Interactive4D: Interactive 4D LiDAR Segmentation. In *IEEE International Conference on Robotics and Automation*, 2025.

[38] J. Valentin, V. Vineet, M.-M. Cheng, D. Kim, J. Shotton, P. Kohli, M. Nießner, A. Criminisi, S. Izadi, and P. Torr. SemanticPaint: Interactive 3D Labeling and Learning at your Fingertips. *ACM Transactions on Graphics*, 34(5), 2015.

[39] M. Bain, J. Huh, T. Han, and A. Zisserman. WhisperX: Time-Accurate Speech Transcription of Long-Form Audio. *INTERSPEECH 2023*, 2023.

[40] I. Sárándi, T. Linder, K. O. Arras, and B. Leibe. MeTRAbs: Metric-Scale Truncation-Robust Heatmaps for Absolute 3D Human Pose Estimation. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2021.

[41] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. SMPL: A Skinned Multi-Person Linear Model. *ACM Transactions on Graphics*, 34(6), 2015.

[42] C. Choy, J. Gwak, and S. Savarese. 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[43] B. Graham, M. Engelcke, and L. van der Maaten. 3D Semantic Segmentation with Submanifold Sparse Convolutional Networks. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[44] M. Tancik, P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. Barron, and R. Ng. Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains. *Neural Information Processing Systems*, 2020.

[45] Y. Yuan, D. Chen, and L. Yan. Interactive Three-dimensional Segmentation Using Region Growing Algorithms. *Journal of Algorithms & Computational Technology*, 2015.

[46] I. Loshchilov and F. Hutter. Decoupled Weight Decay Regularization. *International Conference on Learning Representations*, 2019.

[47] L. N. Smith and N. Topin. Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, 2019.

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

Supplementary Material

A Dataset Details and Ethics

Our study involved seven participants (one female, six males) aged 25–41 (mean: 35.4). While we did not formally assess technological affinity, all participants had engineering backgrounds and were familiar with modern technologies. None reported motor impairments affecting their ability to perform pointing gestures. The ethics board of our research project has approved the consent form signed by the participants before participation.

B 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 [46] and a one-cycle learning rate schedule [47], 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 [42] 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 [46] and a one-cycle learning rate schedule [47], 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.

C DP Dataset Filtering

(a) Transitional pointing frame (b) Final pointing frame

Figure 5: DP Dataset samples.

We use the DP Dataset [11] to train our pointing head. As shown in Fig. 5, 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. 5 (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. 5 (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 [18]. 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.