Bootstrapping a 4D LiDAR Annotation Tool from Video Foundation Models

Jihun Kim[1]*, Hyun-Kurl Jang[1]*, Hyemin Yang[1]*, Jinnyeong Yang[1]*, Hyeokjun Kweon[2]*, and Kuk-Jin Yoon[1]

[1] KAIST, Visual Intelligence Lab
{jihun1998, jhg0001, hyemin0806, jinnyeong6118, kjyoon}@kaist.ac.kr
[2] Chung-Ang University, FoVLab
hyeokjunkweon@cau.ac.kr

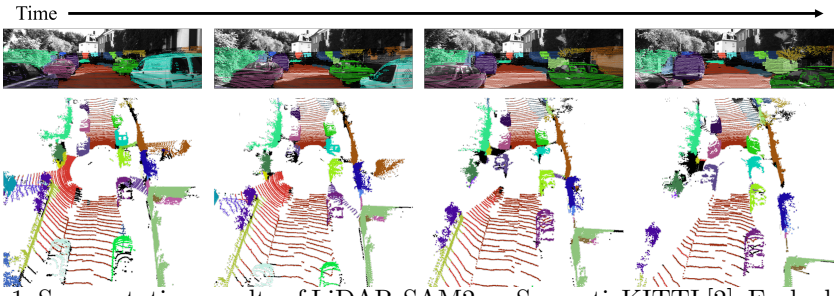


Fig. 1: Segmentation results of LiDAR-SAM2 on SemanticKITTI [2]. Each object is segmented via a user click in the first LiDAR frame, and distinct colors are assigned to different objects for visual distinction.

Abstract. Progress in 4D LiDAR segmentation is bottlenecked by data. Assigning temporally consistent labels across sparse point cloud sequences is costly and hard to scale, and every new task or domain tends to demand fresh dense annotation. This motivates a simple question of whether high-quality LiDAR training data can be produced automatically, without any human labeling. To this end, we introduce **LiDAR-SAM2**, a framework that turns a 2D video foundation model, SAM2, into a scalable source of supervision for the 4D LiDAR domain. On the data side, it automatically generates temporally coherent LiDAR-level labels from SAM2 video masks through multi-view projection and spatio-temporal aggregation. On the modeling side, a tailored modality interface and a two-stage learning objective adapt SAM2's video segmentation kernel to spatio-temporal LiDAR structure, so that a single click per object yields a consistent mask track across the sequence. Trained with no human LiDAR annotation, LiDAR-SAM2 produces semantic and panoptic labels on SemanticKITTI that approach the quality of full human annotation from only a few points, and models trained on these labels approach the performance of full ground-truth supervision. This positions LiDAR-SAM2 as a scalable labeling tool that substantially reduces the annotation burden for 3D and 4D scene understanding.

Keywords: 4D LiDAR Segmentation · Video Foundation Model · Automatic Data Annotation

* Equal contribution.

1 Introduction

Perception in autonomous driving demands a dynamic understanding of 3D scenes, and **4D LiDAR segmentation** plays a pivotal role in this process, offering spatially detailed interpretations of the scene by identifying objects and temporally associating them across point cloud sequences. The field covers a range of tasks, including semantic [32, 16, 20, 30, 35, 29, 8], instance/panoptic [13, 41, 23, 48, 24, 33, 1], and moving objects [5, 28, 43] segmentation. Each of these tasks relies on its own form of supervision, so that a separate set of annotations must be collected for every task and granularity.

Obtaining such annotations directly on LiDAR point clouds, however, is particularly difficult. Labeling sparse, irregular points is far less intuitive than labeling images, and assigning temporally consistent labels across an entire sequence makes the process even more labor-intensive and hard to scale. The practical bottleneck of 4D LiDAR segmentation therefore lies less in the model than in the data it depends on, suggesting the need to revisit the problem from a more data-centric perspective.

Interestingly, a comparable challenge has arisen in the 2D domain. Image segmentation, spanning semantic, panoptic, and instance formulations, has suffered from the same rigidity of task definition and dependence on human labeling. To address these issues, a long line of research has explored interactive segmentation, enabling users to guide models through visual prompts such as points. This idea has recently culminated in foundation models such as SAM [12] and its video counterpart SAM2 [27]. By decoupling what to segment from how to segment, these models have achieved remarkable scalability and generalization across object categories and visual domains. This paradigm has not only turned data annotation into an interactive process, but has also established a foundation for general-purpose segmentation.

In light of these parallels, it is natural to ask whether the LiDAR domain could benefit from a similar shift. Recently, Interactive4D [9] takes a step in this direction with a prompt-driven 4D LiDAR segmentation. However, this approach still relies on human-annotated panoptic labels from a specific dataset (SemanticKITTI [2]). As a result, it inherits the data limitations of prior task-specific approaches. The segmentation granularity is fixed by the training annotations, and every new domain would require re-collecting large amounts of manually labeled 4D data.

We propose **LiDAR-SAM2**, the first interactive 4D LiDAR segmentation framework trained without any manually labeled LiDAR data. Our central idea is to treat a 2D video foundation model as a scalable source of LiDAR supervision rather than relying on a fixed, human-annotated dataset. On the data side, LiDAR-SAM2 is trained entirely from pseudo-labels that are automatically generated by SAM2 through multi-view projection and our spatio-temporal 4D mask aggregation, which turns partial per-camera masks into dense, temporally coherent point-wise labels. On the structural and learning side, we design a tailored modality conversion interface and a two-stage learning objective that adapt

SAM2's RGB video segmentation kernel to 4D LiDAR geometry and temporal structure.

Across extensive experiments, LiDAR-SAM2 shows (1) strong labeling quality on SemanticKITTI [2], producing semantic and panoptic supervision without any human-annotated LiDAR labels, and (2) high downstream utility, where models trained on our automatically generated labels approach the accuracy of models trained on full manual annotation from only minimal prompts. Figure 1 further illustrates the resulting interactive 4D LiDAR segmentation, where a single initial click per object enables LiDAR-SAM2 to produce consistent, high-quality mask tracks across the entire sequence.

Together, these results show that better training data, curated automatically from a foundation model, can bring the scalability and generality of interactive segmentation into the 4D LiDAR domain, offering a practical path toward large-scale, high-quality 3D scene understanding without manual annotation.

2 Related Works

2.1 4D LiDAR Segmentation

LiDAR segmentation is fundamental to understanding dynamic 3D environment, considering temporal coherence across sequences. Early approaches employed point-based or voxelized backbones [31, 49, 45, 6] to model spatial geometry within a single scan, while later 4D methods incorporated temporal cues through recurrent units, temporal attention, or pose-aware feature warping [1, 23, 13, 25, 32, 41]. The field now covers diverse tasks—including semantic, instance, panoptic, and moving objects segmentation—each imposing different levels of label granularity and temporal consistency [1, 32, 25]. Despite these advances, most systems still depend on manually annotated point-level labels, which are expensive and labor-intensive to collect at scale. To address this challenge, we propose an interactive 4D LiDAR segmentation named LiDAR-SAM2, as a labeling assistant. It produces high-quality spatio-temporal segmentation from minimal point prompts, significantly reducing the annotation burden, and is broadly applicable across downstream LiDAR segmentation tasks.

2.2 Leveraging 2D VFMs for 3D

Vision foundation models (VFMs) have shown strong transfer to downstream tasks, with especially large gains in 2D settings where pretraining is image/video-centric. Extending 2D VFMs to 3D perception has proceeded along two primary routes. Lifting-based methods [22, 47, 40, 38, 4, 42, 10, 14] first obtain 2D predictions from a VFM and then reproject them into 3D using calibrated geometry and visibility reasoning. Distillation-based [3, 46, 26, 44, 37, 21, 39] approaches instead train 3D-native backbones under supervision from 2D VFMs via pseudo-labels or feature-level guidance. Both strategies, however, have limitations: lifting pipelines are brittle to calibration and resampling and often underutilize

native 3D geometry, while distillation often erode advantages of large-scale 2D VFM—broad category coverage and open-set generalization. Complementary to these, another line of work incorporates 2D VFMs within 3D frameworks by adding 3D-aware adapters for volumetric inputs [34, 15, 18]. We advance this line in the LiDAR domain by adapting SAM2 via a geometry-preserving interface that enables promptable, LiDAR-only inference.

3 Pseudo-Label Generation using SAM2

We begin by describing how to generate 4D pseudo-labels from synchronized multi-view RGB images. Specifically, SAM2 [27] is applied to the multi-view RGB streams to obtain per-view temporal segmentation masks, which are then transferred to the LiDAR domain via geometric projection and multi-view aggregation. This produces temporally coherent point-wise pseudo-labels across the LiDAR sequence, enabling us to train an interactive 4D LiDAR segmentation model without any manually annotated 3D data.

3.1 Obtaining View-wise Mask Proposals

We denote a time-ordered LiDAR sequence by

$$\mathcal{X} = \{X_t\}_{t=1}^{T}, \quad X_t = \{x_{t,i} \in \mathbb{R}^3\}_{i=1}^{N_t}, \tag{1}$$

where T is the length of the sequence and N_t is the number of points in frame t. We assume access to a synchronized RGB image $I_t \in \mathbb{R}^{H_I \times W_I \times 3}$ at each timestep, with known geometric calibration between the LiDAR and the camera.

We exploit the segment-everything mode[3] to obtain initial mask proposals. This yields a set of 2D binary masks $\{ m_{1,k}^{2D} \}_{k=1}^{K_1}$, where $m_{1,k}^{2D} \in \{0,1\}^{H_I \times W_I}$. K_1 denotes the number of proposals in the initial frame.

Subsequently, each mask $m_{1,k}^{2D}$ is then used as a prompt to SAM2 [27] to obtain its temporal mask trajectory:

$$\text{SAM2}(V \mid m_{1,k}^{2D}) \quad \rightarrow \quad \mathcal{M}_k^{Video} = \{m_{t,k}^{2D}\}_{t=1}^{T} \tag{2}$$

where $V = I_{1:T}$ denotes the video (*i.e.*, image sequence). In other words, SAM2 propagates each initial mask proposal forward through time, producing a consistent segmentation trajectory for each segment across the sequence.

Then, each 2D binary mask is transferred into the 3D LiDAR domain via the calibrated LiDAR-camera projection. We use a lifting function $\Phi_t : \{0,1\}^{H_I \times W_I} \rightarrow \{0,1\}^{N_t}$ to obtain 3D binary mask of k-th proposal at timestep t as

$$m_{t,k}^{3D} = \Phi_t(m_{t,k}^{2D}), \quad m_{t,k}^{3D} \in \{0,1\}^{N_t}. \tag{3}$$

Collectively performing this process on the video mask $\mathcal{M}_k^{Video}$ yields a 4D LiDAR mask track:

$$\mathcal{M}_k^{4D} = \{ m_{t,k}^{3D} \}_{t=1}^{T}. \tag{4}$$

[3] The segment everything mode is implemented in SAM but not yet in SAM2; we thus use SAM for this step without methodological differences.

3.2 Spatio-Temporal 4D Mask Aggregation

The above process assumes a single view per LiDAR scan, but practical setups often use multiple cameras. While each camera yields temporally consistent mask tracks, cross-view consistency is not guaranteed. With limited camera coverage (*e.g.* SemanticKITTI [2]), each view captures only a subset of the scene, causing the same object to split into multiple view-specific mask tracks.

To obtain a unified set of 4D pseudo-labels in the LiDAR domain, we perform spatio-temporal 4D mask aggregation. It aims to merge view-specific tracks that correspond to the same underlying 3D object and to stitch identities over time despite partial view coverage. We begin with spatial aggregation. For each object, multiple view-specific masks may correspond to the same physical instance. To identify such correspondences, we assign the same mask track ID to any set of masks that share common LiDAR points and merge them accordingly. Applying this rule across all views results in a cross-view consistent set of mask tracks.

We then perform temporal aggregation to enforce consistency across time. Since 2D masks are obtained from camera views, the initial pseudo-labels are inherently limited to camera-visible regions, leaving portions of the LiDAR sweep unlabeled due to the restricted camera FoV. To overcome this limitation, we explicitly expand pseudo-label coverage via spatiotemporal aggregation. Concretely, masks from the previous timestep are first propagated into the current LiDAR frame using the known ego-motion. We then robustly fuse the propagated and current masks by voxelizing the point cloud and assigning each voxel the majority mask ID among the points it contains; the voted IDs are broadcasted back to points. Iterating this propagation-and-fusion over time allows labels to persist and grow beyond what is labeled in the current frame alone, mitigating the limited camera coverage. More details are in the Supplementary Material.

On SemanticKITTI, temporal aggregation substantially improves pseudo-label density while preserving quality. Without aggregation, we recover 22.12 segments on average, achieving 91.90% precision but only 13.59% recall. After aggregation, the average number of segments increases to 73.84 and recall rises dramatically to 79.58%, while precision remains high at 85.93%. This yields temporally consistent dense pseudo-labels, enabling high-quality automatic supervision.

4 LiDAR-SAM2

4.1 Overview

Our goal is to build an interactive 4D LiDAR segmentation model that allows a user to specify what to segment through a minimal prompt. Following SAM2 [27], we aim to learn

$$\hat{\mathcal{Y}} = \mathcal{F}(\mathcal{X} \mid \mathcal{P}), \quad \hat{\mathcal{Y}} = \{\hat{Y}_t\}_{t=1}^{T}, \tag{5}$$

where $\mathcal{X} = \{X_t\}_{t=1}^{T}$ is the LiDAR sequence and $\mathcal{P} \subset X_t$ is a set of user-selected point prompts. The output $\hat{\mathcal{Y}}$ is a predicted 4D mask track, where $\hat{Y}_t = \{\hat{y}_{t,i} \in$

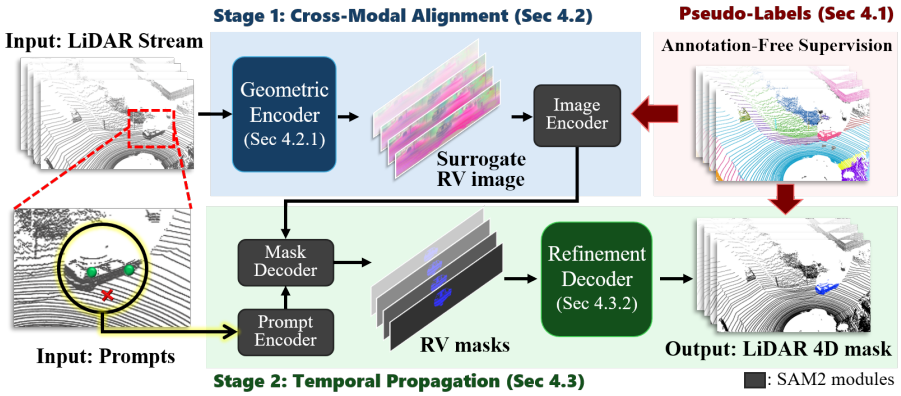


Fig. 2: Overview of the LiDAR-SAM2 framework. The training pipeline consists of two stages, both supervised by pseudo-labels.

$[0,1]\}_{i=1}^{N_t}$ denotes per-point segmentation scores indicating how likely each point $x_{t,i}$ belongs to the segment specified by $\mathcal{P}$.

On the architectural side, rather than designing a new model from scratch, we retain SAM2 as a strong segmentation kernel, and focus on enabling it to operate directly on LiDAR sequences. This preserves SAM2's interactive prompt-based behavior, but also introduces two challenges: (1) LiDAR point clouds are sparse and irregular 3D data, whereas SAM2 expects dense 2D feature maps; and (2) the temporal dynamics of 4D LiDAR sequences differ from those in videos on which SAM2 was originally trained.

To address these issues, we propose an interface-oriented architecture that maps LiDAR frames into and out of SAM2's 2D representation space while preserving their underlying 3D geometric structure. This allows SAM2 to process LiDAR inputs without modifying its architecture, maintaining spatial locality and structure in point space.

We further introduce a two-stage strategy with specialized training objectives. This design is inspired by recent VLMs [19, 17], where the vision-to-language projection layer is first aligned before full fine-tuning. In **Stage 1**, we perform cross-modal alignment so that SAM2's image encoder can effectively interpret LiDAR-derived range-view representations. This alignment is conducted at the frame level, without any temporal modeling. Then, in **Stage 2**, we enable LiDAR-SAM2 to learn temporal propagation and object-level consistency across time for 4D segmentation.

4.2 Learning Cross-Modal Alignment (Stage 1)

The goal of Stage 1 is to align LiDAR domain with SAM2 in a **frame-wise** manner. Directly learning temporal propagation and motion consistency in 4D space is challenging, and SAM2 is trained on the RGB domain only. Therefore, we first ensure that a single LiDAR scan and its information can be interpreted

by SAM2. For clarity, since Stage 1 is frame-wise, we omit the time index t in this section.

We adopt the Range-View (RV) representation to express each LiDAR scan as a dense image-like signal, allowing SAM2 to be used without modifying its architecture. $\rho(\cdot)$ denotes the standard spherical projection from 3D space to RV plane, mapping the points $x_i \in \mathbb{R}^3$ to pixel coordinates $(u, v) \in \{1, \dots, H\} \times \{1, \dots, W\}$.

In addition, the 4D pseudo-labels from Sec. 3 are also projected into the RV domain. Given a mask track $\mathcal{M}_k^{4D} = \{m_{t,k}^{3D}\}_{t=1}^{T}$, each per-frame binary point mask $m_k^{3D} \in \{0,1\}^N$ is mapped to a range-view binary image:

$$m_k^{RV} = \rho(m_k^{3D}) \in \{0,1\}^{H \times W}, \tag{6}$$

which is spatially aligned with the RV domain.

Geometry-aware Range-View Representation To preserve geometric structure while converting LiDAR into an image-like representation, we first extract per-point features using a SSL-pretrained geometric encoder E_{geo}:

$$\mathbf{h} = E_{\text{geo}}(X) \in \mathbb{R}^{N \times d}. \tag{7}$$

To make these features compatible with SAM2's image encoder trained on the RGB domain, we employ a lightweight MLP g to map them into a 3-channel embedding:

$$\mathbf{c} = g(\mathbf{h}) \in \mathbb{R}^{N \times 3}, \tag{8}$$

interpreting LiDAR features as an RGB signal. We then rasterize the point-wise embeddings into a dense RV image:

$$\hat{I}^{\text{RV}} = \rho(X, \mathbf{c}) \in \mathbb{R}^{H \times W \times 3}. \tag{9}$$

The resulting surrogate RV image is processed by the SAM2 image encoder with trainable LoRA [11]:

$$\mathbf{F} = E_{\text{SAM2}}(\hat{I}^{\text{RV}}) \in \mathbb{R}^{H \times W \times D}. \tag{10}$$

Objective Function for Interactive Segmentation We train our framework with SAM2-style loss. From the pseudo-label m^{RV}, we sample a set of foreground/background pixels to serve as positive/negative prompts. This set of prompts $\mathcal{P}$ is fed into the prompt encoder, and combined with the feature map $\mathbf{F}$ through the SAM2 decoder to predict a mask $\hat{y}^{RV}$. Accordingly, the loss is defined by a segmentation loss between the predicted mask and the pseudo-label in the RV domain:

$$\mathcal{L}_{\text{interactive}}^{\text{frame-wise}} = \text{RVSegLoss}(\hat{y}^{RV}, m^{RV}), \tag{11}$$

where the details about RVSegLoss is in *Supplementary Material*. To sum up, the full objective for Stage 1 is:

$$\mathcal{L}_{\text{Stage1}} = \mathcal{L}_{\text{interactive}}^{\text{frame-wise}}. \tag{12}$$

Note that Stage 1 updates the LiDAR encoder (E_{geo}), the lightweight MLP (g), and SAM2's image encoder (E_{SAM2} with LoRA). The other parts of SAM2 remain unchanged.

4.3 Learning Temporal Propagation (Stage 2)

Stage 2 focuses on learning how segmentation should propagate over time. The model receives a LiDAR sequence $\mathcal{X} = \{X_t\}_{t=1}^{T}$, and each scan is converted into its RV feature map: $\mathbf{F}_t = E_{\text{SAM2}}(\text{RV}_t)$.

4D Decoder for Refinement Since this track is still in the RV domain, we obtain the final 4D point-wise mask track by inverse projection:

$$\hat{\mathcal{Y}}^{4D} = \{\hat{y}_t^{4D}\}_{t=1}^{T}, \quad \hat{y}_t^{4D} = \rho^{-1}(\hat{y}_t^{RV}). \tag{13}$$

To further restore LiDAR-level geometric detail, we introduce a lightweight 3D refinement LiDAR decoder D_{ref}. Specifically, given a temporal window $t : t+\Delta t$, we place the lifted predictions[4] into D_{ref} for each frame in the window and refine them in point space. We denote this refinement over the entire window as

$$\tilde{\mathcal{Y}}_{t:t+\Delta t}^{4D} = D_{\text{ref}}(X_{t:t+\Delta t}, \hat{\mathcal{Y}}_{t:t+\Delta t}^{4D}). \tag{14}$$

This decoding stage corrects projection-induced artifacts and reinstates surface continuity in 3D space, yielding coherent and stable segmentation results.

Finally, the refined 4D mask track $\tilde{\mathcal{Y}}^{4D}$ is supervised using the 4D pseudo-label $\mathcal{M}^{4D}$ in the LiDAR domain via

$$\mathcal{L}_{\text{refine}} = \text{4DSegLoss}(\tilde{\mathcal{Y}}^{4D}, \mathcal{M}^{4D}), \tag{15}$$

where the details about 4DSegLoss are in *Supplementary Material*. To sum up, the full objective for Stage 2 is:

$$\mathcal{L}_{\text{Stage2}} = \mathcal{L}_{\text{interactive}}^{\text{temporal}} + \mathcal{L}_{\text{refine}}. \tag{16}$$

During Stage 2, we freeze the modules trained in Stage 1 (E_{geo}, g, E_{SAM2}), and train the remaining SAM2 components (prompt encoder, memory attention, and mask decoder) as well as the additional temporal modules introduced below. This allows the model to learn how to propagate segmentation consistently across time without disrupting the modality alignment established in Stage 1.

[4] In practice, we use the lifted SAM2 decoder logits before thresholding rather than binary masks. Details can be found in Supplementary Material.

5 Experiments

5.1 Settings

Datasets We evaluate LiDAR-SAM2 on SemanticKITTI [2], which consists of 64-beam LiDAR scans synchronized with two forward-facing RGB cameras. We train on the training set and report all results on the validation set.
Metric For semantic segmentation, we report the mean Intersection-over-Union (mIoU) averaged over the 19 SemanticKITTI classes. For 4D panoptic segmentation, we adopt the LiDAR Segmentation and Tracking Quality (LSTQ) [1], which factorizes into an association score (S_{assoc}) measuring temporal instance consistency and a classification score (S_{cls}). We further report semantic IoU separately for stuff (IoU_{st}) and thing (IoU_{th}) classes. All metrics are computed on the respective validation sets.
Implementation All experiments are conducted on a single NVIDIA A6000 GPU with the pretrained SAM2 *base+*. The geometric encoder is pre-trained using FCGF [7] on the same datasets used for evaluation. The refinement head D_{ref} is implemented as MinkUNet-14 [6].

5.2 LiDAR-SAM2 as Labeling Tool

We evaluate LiDAR-SAM2 as a labeling tool for all entities in the LiDAR scene by simulating human interaction on 4-frame LiDAR sweeps. At the first LiDAR frame, a fixed number of point prompts is provided for every instance to initialize the annotation, following a standard interactive segmentation protocol. The first positive click is placed near the object center, and subsequent clicks are added by inspecting the true-negative and false-positive regions, where we click on whichever region is larger at that time. For stuff categories such as road and sidewalk that are not instance-separated, we voxelize the current point cloud with a voxel size of 0.2 m and compute connected components in voxel space, treating each disconnected component as a separate pseudo-instance even when several share the same semantic label, which keeps propagation and editing localized and consistent across sweeps. The predicted masks are then propagated through the remaining frames and one additional frame to seed the next sweep. Carried masks are reused in the following sweep only if their overlap with the corresponding ground-truth mask exceeds an IoU threshold of 0.8, and otherwise, including for newly appearing objects, we discard the carried mask and introduce additional human prompts to re-initialize the instance. On average, this process corresponds to roughly **ten point prompts per frame**.

Semantic Segmentation We first assess our labels on semantic segmentation by training two standard backbones, MinkowskiUNet (MinkUNet) [6] and PointTransformer v2 (PTv2) [36], from scratch on labels of different origin. As reported in Table 3, we compare four label sources. The first is the raw point prompts used during interaction. The second is labels from SAM2 inference via RGB projection. The third is labels generated by LiDAR-SAM2. The fourth is

Table 1: Semantic segmentation results (mIoU) trained with point labels, SAM2 labels, our labels, and full GT. The number in parentheses next to each label type denotes the number of oracle-provided points per frame. Experiments are on the SemanticKITTI [2] validation set.

Methods	Label type	car	bicycle	motorcycle	truck	other-vehicle	person	bicyclist	motorcyclist	road	parking	sidewalk	other-ground	building	fence	vegetation	trunk	terrain	pole	traffic-sign	mIoU
MinkU [6]	Point labels (10)	9.7	0.5	2.4	5.6	4.0	0.8	1.6	3.4	0.8	2.5	4.1	0.2	5.3	4.0	1.8	10.4	10.4	23.1	7.4	5.2
	SAM2 labels (10)	61.0	0.6	6.5	50.3	11.9	25.1	44.7	0.1	57.8	6.0	7.4	0.6	67.6	21.8	53.5	21.3	47.8	45.7	37.2	29.8
	Our labels (10)	84.3	17.8	60.5	75.8	46.8	58.3	73.0	2.5	85.2	36.8	69.6	0.3	84.6	52.2	84.7	63.6	69.9	49.6	33.2	55.2
	Full GT (100k)	96.1	35.0	64.8	83.4	59.9	73.4	88.3	0.0	93.7	53.1	81.0	7.0	91.0	61.7	88.0	67.8	75.5	62.5	48.6	64.8
PTv2 [36]	Point labels (10)	27.6	1.9	5.8	3.2	6.6	0.9	2.0	0.0	1.2	4.8	0.2	0.1	6.2	3.7	1.8	15.0	2.7	41.0	16.6	7.4
	SAM2 labels (10)	78.7	0.1	0.0	0.0	15.3	14.6	22.3	1.9	53.9	1.4	0.7	0.0	72.1	23.1	64.5	22.8	51.5	43.2	35.9	26.4
	Our labels (10)	86.2	34.1	71.9	78.2	39.7	69.1	80.7	30.1	86.6	45.1	72.9	8.3	86.3	59.4	86.8	62.2	73.7	52.9	41.1	61.3
	Full GT (100k)	96.0	53.5	83.9	88.6	59.4	82.2	93.4	19.2	95.3	48.7	83.2	0.1	91.3	65.1	88.1	71.2	74.3	65.9	50.8	69.0

Table 2: Panoptic segmentation results trained with our labels and full GT. Experiments are evaluated on SemanticKITTI [2] validation set.

Methods	Label type	LSTQ	S_{assoc}	S_{cls}	IoU_{st}	IoU_{th}
4D-PLS [1]	Our labels	46.9	46.3	47.6	54.3	44.3
4D-PLS [1]	Full GT	62.7	65.1	60.5	65.4	61.3
4D-StOP [13]	Our labels	57.7	64.2	51.7	56.5	51.7
4D-StOP [13]	Full GT	67.0	74.4	60.3	65.3	60.9
Mask4Former [41]	Our labels	64.8	70.7	59.4	56.9	62.8
Mask4Former [41]	Full GT	70.5	74.3	66.9	67.1	66.6

full human ground truth. Training on the sparse point prompts alone yields very low accuracy, reflecting the limited supervision. SAM2 labels give a moderate improvement but remain insufficient, suffering from projection artifacts and a lack of adaptation to the LiDAR domain. In contrast, LiDAR-SAM2 labels lead to a substantial gain, roughly doubling the mIoU of naive SAM2, and bring both backbones close to models trained on full ground truth, despite using no human LiDAR annotation.

Panoptic Segmentation We further validate our labels on 4D panoptic segmentation, which additionally requires temporally consistent instance identities. We train three panoptic backbones, namely 4D-PLS [1], 4D-StOP [13], and Mask4Former [41], using labels generated by LiDAR-SAM2, and compare against the same backbones trained on full ground truth (Table 4). Models supervised by our labels achieve strong panoptic quality across all metrics, including LSTQ, S_{assoc}, S_{cls}, and the stuff and thing IoUs. They recover a large fraction of the full ground-truth performance without any manual LiDAR labels.

Table 3: Semantic segmentation results (mIoU) trained with point labels, SAM2 labels, our labels, and full GT. Experiments are on the SemanticKITTI [2] validation set.

Label type	pts/frame	MinkUNet [6]	PTv2 [36]
Point labels	10	5.2	7.4
SAM2 labels	10	24.3	26.4
Our labels	10	55.2	61.3
Full GT	100k	63.8	70.3

Table 4: Panoptic segmentation results trained with our labels and full GT. Experiments are evaluated on SemanticKITTI [2] validation set.

Method	LSTQ	S_{assoc}	S_{cls}	IoU_{st}	IoU_{th}
4D-PLS [1] + Our labels	46.9	46.3	47.6	54.3	44.3
4D-PLS [1] + Full GT	62.7	65.1	60.5	65.4	61.3
4D-StOP [13] + Our labels	57.7	64.2	51.7	56.5	51.7
4D-StOP [13] + Full GT	67.0	74.4	60.3	65.3	60.9
Mask4Former [41] + Our labels	64.8	70.7	59.4	56.9	62.8
Mask4Former [41] + Full GT	70.5	74.3	66.9	67.1	66.6

Labeling Quality Beyond downstream accuracy, we inspect the labels produced by LiDAR-SAM2 directly. fig:annotation shows instance-level and semantic-level labels generated by our tool. Object boundaries are clean and identities remain consistent across time, closely matching the underlying scene structure. Together with the strong downstream performance reported above, these labels confirm that LiDAR-SAM2 is a readily deployable labeling tool for 3D and 4D LiDAR segmentation, dramatically reducing the manual annotation burden.

6 Conclusion

We present LiDAR-SAM2, a novel framework for interactive 4D LiDAR segmentation that requires no manual LiDAR labels. By distilling temporal priors from SAM2 into the LiDAR domain, our approach leverages synchronized video–LiDAR sequences to train a LiDAR-only model capable of producing spatiotemporally consistent interactive segmentation. A tailored LiDAR-aware pipeline and specialized training objectives enable SAM2's video segmentation kernel to operate effectively on LiDAR data while preserving strong temporal coherence. Our experiments demonstrate that LiDAR-SAM2 substantially reduces annotation cost, achieving strong performance with only minimal prompting.

7 Acknowledgement

This work was supported by the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea (NRF-2025S1A5C3A04022639).

References

1. Aygun, M., Osep, A., Weber, M., Maximov, M., Stachniss, C., Behley, J., Leal-Taixé, L.: 4d panoptic lidar segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5527–5537 (2021)
2. Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., Gall, J.: SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In: Proc. of the IEEE/CVF International Conf. on Computer Vision (ICCV) (2019)

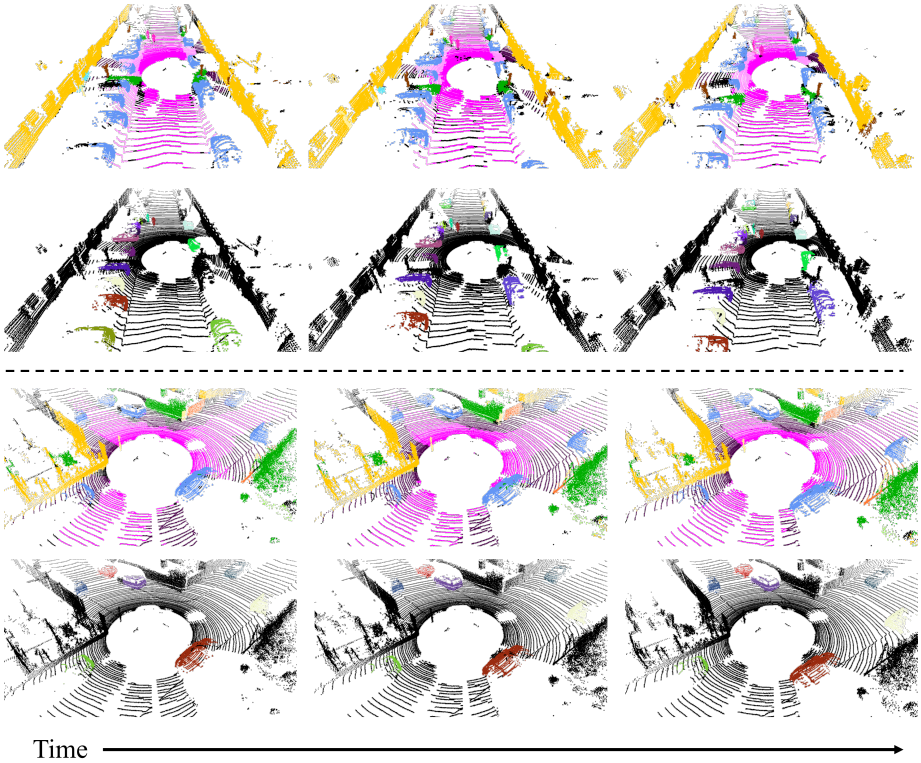


Fig. 3: Qualitative results of labels annotated with LiDAR-SAM2. The top shows instance labels, while the bottom shows semantic labels.

3. Çanakçı, A.S., Vödisch, N., Petek, K., Burgard, W., Valada, A.: Label-efficient lidar panoptic segmentation. arXiv preprint arXiv:2503.02372 (2025)
4. Cen, J., Zhou, Z., Fang, J., Shen, W., Xie, L., Jiang, D., Zhang, X., Tian, Q., et al.: Segment anything in 3d with nerfs. Advances in Neural Information Processing Systems **36**, 25971–25990 (2023)
5. Chen, X., Li, S., Mersch, B., Wiesmann, L., Gall, J., Behley, J., Stachniss, C.: Moving object segmentation in 3d lidar data: A learning-based approach exploiting sequential data. IEEE Robotics and Automation Letters **6**(4), 6529–6536 (2021)
6. Choy, C., Gwak, J., Savarese, S.: 4d spatio-temporal convnets: Minkowski convolutional neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3075–3084 (2019)
7. Choy, C., Park, J., Koltun, V.: Fully convolutional geometric features. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 8958–8966 (2019)
8. Duerr, F., Pfaller, M., Weigel, H., Beyerer, J.: Lidar-based recurrent 3d semantic segmentation with temporal memory alignment. In: 2020 International Conference on 3D Vision (3DV). pp. 781–790. IEEE (2020)
9. Fradlin, I., Zulfikar, I.E., Yilmaz, K., Kontogianni, T., Leibe, B.: Interactive4d: Interactive 4d lidar segmentation. arXiv preprint arXiv:2410.08206 (2024)
10. He, Q., Peng, J., Jiang, Z., Hu, X., Zhang, J.: Pointseg: A training-free paradigm for 3d scene segmentation via foundation models. In: Proceedings of the IEEE/CVF

International Conference on Computer Vision. pp. 2657–2667 (2025)
11. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al.: Lora: Low-rank adaptation of large language models. ICLR **1**(2), 3 (2022)
12. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4015–4026 (2023)
13. Kreuzberg, L., Zulfikar, I.E., Mahadevan, S., Engelmann, F., Leibe, B.: 4d-stop: Panoptic segmentation of 4d lidar using spatio-temporal object proposal generation and aggregation. In: European Conference on Computer Vision. pp. 537–553. Springer (2022)
14. Kweon, H., Kim, J., Yoon, K.J.: Weakly supervised point cloud semantic segmentation via artificial oracle. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3721–3731 (2024)
15. Li, C., Sultan, R.I., Khanduri, P., Qiang, Y., Indrin, C., Zhu, D.: Autoprosam: Automated prompting sam for 3d multi-organ segmentation. In: 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 3570–3580. IEEE (2025)
16. Li, E., Casas, S., Urtasun, R.: Memoryseg: Online lidar semantic segmentation with a latent memory. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 745–754 (2023)
17. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In: International conference on machine learning. pp. 19730–19742. PMLR (2023)
18. Li, S., Zhang, J., Qi, L., Shi, Y.: Ga-sam: Geometry-aware sam adaptation with sparse annotation-driven point cloud completion. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 214–224. Springer (2025)
19. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. Advances in neural information processing systems **36**, 34892–34916 (2023)
20. Liu, J., Chang, C., Liu, J., Wu, X., Ma, L., Qi, X.: Mars3d: A plug-and-play motion-aware model for semantic segmentation on multi-scan 3d point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9372–9381 (2023)
21. Liu, M., Uy, M.A., Xiang, D., Su, H., Fidler, S., Sharp, N., Gao, J.: Partfield: Learning 3d feature fields for part segmentation and beyond. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9704–9715 (2025)
22. Liu, M., Zhu, Y., Cai, H., Han, S., Ling, Z., Porikli, F., Su, H.: Partslip: Low-shot part segmentation for 3d point clouds via pretrained image-language models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 21736–21746 (2023)
23. Marcuzzi, R., Nunes, L., Wiesmann, L., Marks, E., Behley, J., Stachniss, C.: Mask4d: End-to-end mask-based 4d panoptic segmentation for lidar sequences. IEEE Robotics and Automation Letters **8**(11), 7487–7494 (2023)
24. Marcuzzi, R., Nunes, L., Wiesmann, L., Vizzo, I., Behley, J., Stachniss, C.: Contrastive instance association for 4d panoptic segmentation using sequences of 3d lidar scans. IEEE Robotics and Automation Letters **7**(2), 1550–1557 (2022)
25. Mersch, B., Chen, X., Vizzo, I., Nunes, L., Behley, J., Stachniss, C.: Receding moving object segmentation in 3d lidar data using sparse 4d convolutions. IEEE Robotics and Automation Letters **7**(3), 7503–7510 (2022)

26. Ošep, A., Meinhardt, T., Ferroni, F., Peri, N., Ramanan, D., Leal-Taixé, L.: Better call sal: Towards learning to segment anything in lidar. In: European Conference on Computer Vision. pp. 71–90. Springer (2024)
27. Ravi, N., Gabeur, V., Hu, Y.T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., et al.: Sam 2: Segment anything in images and videos. arXiv preprint arXiv:2408.00714 (2024)
28. Rozsa, Z., Madaras, A., Sziranyi, T.: Efficient moving object segmentation in lidar point clouds using minimal number of sweeps. IEEE Open Journal of Signal Processing (2025)
29. Shi, H., Lin, G., Wang, H., Hung, T.Y., Wang, Z.: Spsequencenet: Semantic segmentation network on 4d point clouds. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4574–4583 (2020)
30. Shi, H., Wei, J., Wang, H., Liu, F., Lin, G.: Learning temporal variations for 4d point cloud segmentation. International Journal of Computer Vision **132**(12), 5603–5617 (2024)
31. Thomas, H., Qi, C.R., Deschaud, J.E., Marcotegui, B., Goulette, F., Guibas, L.J.: Kpconv: Flexible and deformable convolution for point clouds. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6411–6420 (2019)
32. Wang, N., Guo, R., Shi, C., Wang, Z., Zhang, H., Lu, H., Zheng, Z., Chen, X.: Segnet4d: Efficient instance-aware 4d semantic segmentation for lidar point cloud. IEEE Transactions on Automation Science and Engineering (2025)
33. Wang, N., Shi, C., Guo, R., Lu, H., Zheng, Z., Chen, X.: Insmos: Instance-aware moving object segmentation in lidar data. In: 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 7598–7605. IEEE (2023)
34. Wang, Z., Yu, X., Rao, Y., Zhou, J., Lu, J.: P2p: Tuning pre-trained image models for point cloud analysis with point-to-pixel prompting. Advances in neural information processing systems **35**, 14388–14402 (2022)
35. Wu, X., Hou, Y., Huang, X., Lin, B., He, T., Zhu, X., Ma, Y., Wu, B., Liu, H., Cai, D., et al.: Taseg: Temporal aggregation network for lidar semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15311–15320 (2024)
36. Wu, X., Lao, Y., Jiang, L., Liu, X., Zhao, H.: Point transformer v2: Grouped vector attention and partition-based pooling. Advances in Neural Information Processing Systems **35**, 33330–33342 (2022)
37. Xu, J., Wang, S., Ni, Z., Hu, C., Yang, S., Zhu, J., Li, Q.: Sam4d: Segment anything in camera and lidar streams. arXiv preprint arXiv:2506.21547 (2025)
38. Yang, X., Gu, X., Yin, X., Gao, X.: Sa3dip: Segment any 3d instance with potential 3d priors. Advances in Neural Information Processing Systems **37**, 102568–102586 (2024)
39. Yang, Y., Huang, Y., Guo, Y.C., Lu, L., Wu, X., Lam, E.Y., Cao, Y.P., Liu, X.: Sampart3d: Segment any part in 3d objects. arXiv preprint arXiv:2411.07184 (2024)
40. Yang, Y., Wu, X., He, T., Zhao, H., Liu, X.: Sam3d: Segment anything in 3d scenes. arXiv preprint arXiv:2306.03908 (2023)
41. Yilmaz, K., Schult, J., Nekrasov, A., Leibe, B.: Mask4former: Mask transformer for 4d panoptic segmentation. In: 2024 IEEE International Conference on Robotics and Automation (ICRA). pp. 9418–9425. IEEE (2024)
42. Yin, Y., Liu, Y., Xiao, Y., Cohen-Or, D., Huang, J., Chen, B.: Sai3d: Segment any instance in 3d scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3292–3302 (2024)

43. Zeng, K., Shi, H., Lin, J., Li, S., Cheng, J., Wang, K., Li, Z., Yang, K.: Mambamos: Lidar-based 3d moving object segmentation with motion-aware state space model. In: Proceedings of the 32nd ACM International Conference on Multimedia. pp. 1505–1513 (2024)
44. Zhang, Y., Ošep, A., Leal-Taixé, L., Meinhardt, T.: Zero-Shot 4D Lidar Panoptic Segmentation. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2025)
45. Zhou, H., Zhu, X., Song, X., Ma, Y., Wang, Z., Li, H., Lin, D.: Cylinder3d: An effective 3d framework for driving-scene lidar semantic segmentation. arXiv preprint arXiv:2008.01550 (2020)
46. Zhou, Y., Gu, J., Chiang, T.Y., Xiang, F., Su, H.: Point-sam: Promptable 3d segmentation model for point clouds (2024)
47. Zhou, Y., Gu, J., Li, X., Liu, M., Fang, Y., Su, H.: Partslip++: Enhancing low-shot 3d part segmentation via multi-view instance segmentation and maximum likelihood estimation. arXiv preprint arXiv:2312.03015 (2023)
48. Zhu, M., Han, S., Cai, H., Borse, S., Ghaffari, M., Porikli, F.: 4d panoptic segmentation as invariant and equivariant field prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 22488–22498 (2023)
49. Zhu, X., Zhou, H., Wang, T., Hong, F., Ma, Y., Li, W., Li, H., Lin, D.: Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9939–9948 (2021)