RangeSAM: Promptable Segmentation of Sparse 3D Point Clouds in Outdoor Driving Scenes

by

Arya Araban (1439683)

Supervised by:

Kourosh Khoshelham, Cipher Zhang

A thesis submitted in partial fulfillment of the
Master of Computer Science (MC-CS) degree

in the
School of Computing and Information Systems
The University of Melbourne

October 2024

Abstract

Point cloud segmentation stands as a critical component in 3D scene understanding, with particular significance in autonomous driving applications. While semantic segmentation has seen substantial progress, class-agnostic segmentation remains challenging due to its inherent need for adaptability across diverse object types. This challenge has sparked interest in promptable segmentation approaches, which offer promising solutions for handling arbitrary objects through guided inputs. However, existing 3D promptable segmentation methods primarily target dense point clouds, overlooking the critical challenge of sparsity prevalent in real-world LiDAR data, especially in outdoor driving scenes. We introduce RangeSAM, a novel architecture for promptable point cloud segmentation specifically designed to address this limitation. Our approach seamlessly integrates range-based processing techniques with principles from state-of-the-art 2D promptable segmentation models, enabling effective handling of sparse and irregular LiDAR data while maintaining high segmentation fidelity. RangeSAM incorporates a unique point prompt mechanism with and leverages transformer-based components to capture long-range dependencies, allowing for intuitive user guidance and robust segmentation in complex scenes. Through extensive evaluation on multiple datasets, including KITTI360, SemanticKITTI, and ApolloScape, we demonstrate that RangeSAM achieves competitive performance against existing promptable segmentation methods while effectively handling sparse point clouds. Our model achieves a score of 56.05% IoU with 20 point prompts on KITTI360, with experiments conducted confirming the effective utilization of prompt information. Notably, RangeSAM maintains strong performance across datasets unseen during training, achieving over 46% IoU on ApolloScape despite significant domain differences, and demonstrates competitive results against non-promptable approaches on SemanticKITTI with a competetive F1 score (74.4) and segmentation association metric (50.4). These results, combined with comprehensive ablation studies validating our distinctive prompt-centric design, establish RangeSAM as a promising solution for real-world applications requiring adaptive, user-guided segmentation of sparse point cloud data.

Declaration of Authorship

I, Arya Araban, declare that this thesis and the work presented in it are my own. I confirm that:

- This thesis does not incorporate without acknowledgement any material previously submitted for degree or diploma in any university; and that to the best of my knowledge and belief it does not contain any material previously published or written by another person where due reference is not made in the text.
- Clearance for this research from the University's Ethics Committee was not required.
- The thesis is 24,704 words in length (excluding text in images, tables, bibliographies, and appendices).

Signed: *Arya Araban*

Date: *October 28, 2024*

Acknowledgements

I would like to express my heartfelt gratitude to everyone who provided guidance, encouragement, and support throughout the journey of completing this thesis. Firstly, I wish to extend my appreciation to my supervisors, A/Prof. Kourosh Khoshelham and PhD candidate Cipher Zhang, for their invaluable insights and feedback which contributed greatly to the development of this work. I extend my sincere thanks to PhD candidate Zexian Huang, whose perspective was instrumental in helping me gain confidence and clarity in my project direction at crucial stages. I am also immensely thankful to my friends and family, whose unwavering support and encouragement during challenging times enabled me to persevere and remain focused.

Each of these contributions has played a meaningful role in bringing this thesis to fruition, and for that, I am truly grateful.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

The advent of three-dimensional (3D) sensing technologies has revolutionized spatial data acquisition and analysis, with point clouds emerging as a powerful representation for capturing intricate geometric structures of real-world environments. Point clouds, composed of sets of points in three-dimensional space, offer a direct and detailed depiction of 3D geometry, preserving crucial depth information and enabling precise localization and mapping of objects and surfaces [1]. This rich representation has found widespread applications across various domains, with autonomous driving standing out as one of the most prominent examples of its impact [2].

Acquiring point cloud data has become increasingly accessible and prevalent due to the widespread adoption of Light Detection and Ranging (LiDAR) sensors, stereo cameras, and other depth-sensing devices. LiDAR, in particular, has become a cornerstone technology in autonomous driving systems, providing high-resolution 3D scans of the vehicle's surroundings [3]. These scans capture the spatial relationships between various objects in the environment, forming a crucial input for perception, localization, and decision-making algorithms.

While point clouds contain detailed spatial information, they do not inherently hold semantic meaning. To extract actionable insights, it is necessary to segment the point cloud into meaningful regions or objects. This process, known as point cloud segmentation, is a fundamental task in 3D computer vision and plays a pivotal role in scene understanding. Segmentation enables the identification and delineation of various objects within the point cloud, facilitating downstream tasks such as object detection, classification, and tracking [4].

Traditionally, point cloud segmentation has been approached through semantic segmentation, where each point in the cloud is assigned a predefined class label (e.g., car, pedestrian, building) [5]. While semantic segmentation has shown remarkable success in many applications, it faces limitations in dynamic and open-world environments where the range of possible object categories is vast and often unpredictable. This challenge is particularly acute in autonomous driving scenarios, where the ability to identify and segment arbitrary objects, including those not seen during training, is crucial for safe and robust operation.

In response to these limitations, there has been growing interest in class-agnostic segmentation approaches for point clouds. Class-agnostic segmentation aims to partition the point cloud into distinct object instances without relying on predefined class labels [6]. This approach offers greater flexibility and generalization, as it can potentially segment any object, regardless of whether it was represented in the training data. Class-agnostic segmentation aligns well with the concept of open-set recognition, which is increasingly important in real-world applications where encountering novel object types is commonplace [7]. However, while this approach removes the dependence on predefined classes, they often lack the precision and adaptability needed for nuanced segmentation of objects in complex scenes.

Recent advancements in 2D image segmentation, exemplified by models like the Segment Anything Model [8], have demonstrated the potential of a specific paradigm of class-agnostic segmentation known as promptable segmentation. These models can segment arbitrary objects in images based on user-guided inputs, also known as "prompts," which can include points, boxes, or even textual descriptions. In addition to allowing user interaction, promptable segmentation offers more nuanced object delineation by responding to specific user inputs, enabling finer-grained control over segmentation boundaries. Moreover, these models can generate segmentations automatically by internally simulating user prompts, further enhancing their adaptability.

The success of such approaches in the 2D domain has sparked interest in developing analogous capabilities for 3D point cloud data. However, extending these ideas to the 3D domain presents unique difficulties. Point clouds, unlike 2D images, are unstructured and irregular, lacking the grid-like arrangement of pixels. This makes it difficult to directly apply conventional convolutional neural network architectures, and feature extraction and object delineation become much more complex.

Furthermore, the variable density and distribution of point clouds add to the complexity. LiDAR scans typically produce dense clusters of points for close-range objects but sparse representations for distant ones. This inherent sparsity, especially in outdoor environments, complicates the development of robust segmentation algorithms.

Introducing an interactive or promptable aspect to segmentation introduces another layer of intricacy in the 3D domain. While clicking or drawing on a 2D image is intuitive and straightforward, interacting with a 3D point cloud requires more sophisticated interfaces and visualization techniques. The ability to efficiently and accurately provide prompts in 3D space is crucial for the practical application of interactive segmentation in real-world scenarios.

1.2 Problem Statement

While progress has been made in 3D promptable segmentation of point clouds, a major challenge remains entirely unaddressed in current research: the lack of consideration for sparsity, a key characteristic of LiDAR scans. This issue is particularly important in practical applications like autonomous driving, where LiDAR plays a crucial role.

Recent research has emphasized the potential of promptable segmentation for point clouds, with these methods being typically derived from indoor environments or close-range 3D scans [9, 10]. Consequently, they face challenges when dealing with the varying point densities found in real-world outdoor LiDAR data. The performance gap between dense and sparse point cloud segmentation remains largely unexplored, as these methods are primarily developed and tested on dense datasets.

In contrast, in the field of 2D image segmentation, Transformer-based architectures and attention mechanisms have demonstrated great potential in capturing complex spatial relationships [11]. These attention mechanisms enable models to focus on relevant input features, effectively capturing long-range dependencies. Meanwhile, range projection has emerged as a promising technique for bridging 2D and 3D domains, offering an efficient way to represent sparse 3D data in a more structured format which preserves depth information [12].

Despite these advancements, the combination of range projection with attention mechanisms for promptable, class-agnostic segmentation in sparse point clouds remains unexplored. Such an approach could potentially address the sparsity challenge by leveraging the efficiency of range projection in handling 3D data and the capacity of attention mechanisms to capture long-range dependencies, even in the irregular data distributions of outdoor LiDAR scans. This integration presents a promising avenue for improving segmentation performance on sparse point clouds, thereby advancing the field towards more robust and versatile segmentation solutions.

This research gap highlights a significant opportunity in the field of point cloud segmentation. The lack of consideration for sparsity of LiDAR scans has led to a dearth of

methods that can effectively handle this type of data in real-world scenarios [13]. Addressing this gap is essential for advancing the state-of-the-art in point cloud segmentation and enabling practical applications, particularly in the context of autonomous driving and other outdoor robotics applications.

By focusing on the development of a novel approach that combines range projection and attention mechanisms for promptable segmentation in sparse outdoor scenes, this thesis seeks to bridge the gap between the capabilities of current segmentation methods and the demands of real-world outdoor LiDAR data processing.

1.3 Research Objectives and Scope

The primary objective of this research is to develop and evaluate a promptable, class-agnostic segmentation model for 3D point clouds, with a particular emphasis on sparse outdoor driving scenes. This focus is motivated by the critical importance of robust perception in autonomous driving applications and the unique challenges posed by outdoor LiDAR data.

Specifically, the research encompasses the following key areas:

- **Model Architecture**: The core of this research involves designing a novel architecture for promptable 3D point cloud segmentation in sparse environments. This architecture will utilize range projection techniques and attention mechanisms, adapting and extending these concepts to address the unique challenges of sparse, unstructured 3D data from LiDAR scans.
- **Prompt Mechanism**: This research will focus on investigating and adapting point-based prompts as the primary mechanism for encoding semantic information in 3D point clouds, as these techniques have proven to be effective in 3D space.
- **Sparsity Handling**: A significant focus will be on effectively handling the sparsity inherent in outdoor LiDAR scans. This includes exploring novel mechanisms that can operate effectively on sparse, irregular point distributions.
- **Generalizability**: While the focus is on outdoor driving scenes, the research will also examine the model's ability to generalize to scene types from distinct outdoor geographical locations that were not explicitly represented in the training data
- **Evaluation**: The research will include a comprehensive evaluation of the proposed model on standard benchmarks for interactive point cloud segmentation. This will

involve both quantitative metrics and qualitative analysis to assess the model's performance across various scenarios and object types.

To guide this research, we address the following research question:

RQ: Can a 3D promptable segmentation model that integrates range projection and attention mechanisms effectively mitigate the challenge of sparsity in outdoor LiDAR scans, thereby enabling promptable segmentation across varying point densities?

By addressing this question, this research aims to bridge the gap between the capabilities demonstrated in 2D promptable segmentation and the unique challenges posed by sparse 3D point cloud data.

To the best of our knowledge, this work represents the first attempt to adapt 3D promptable segmentation techniques to sparse LiDAR data, addressing a significant limitation in current methods. The findings from this study will contribute to the development of more robust and versatile segmentation methods for autonomous driving and potentially other applications involving sparse 3D data.

Although this research centers on the development of a novel segmentation model for sparse point clouds, there are certain considerations that fall outside its immediate scope. Specifically:

1. While the model aims for class-agnostic segmentation, the primary evaluation and optimization will be conducted on object types commonly found in outdoor driving scenes.

2. The research will focus on LiDAR-based point clouds, although the developed methods may be applicable to point clouds from other sources.

3. Following established practices, our evaluation will simulate prompts based on ground truth instance annotations. This research will not explore prompt simulation techniques for automated scene segmentation without such guidance.

4. Real-time performance, while desirable, is not a primary focus of this research. The emphasis is on developing effective segmentation techniques for sparse data, with optimization for speed being a potential area for future work.

5. The research will not extensively explore multi-modal approaches combining point clouds with other sensor data (e.g., camera images), even though there may exist potential for such extensions.

6. The study will primarily focus on the architectural aspects of combining range projection with attention mechanisms, rather than exploring novel training paradigms or data augmentation techniques.

By maintaining this focused scope, the research intends to make significant contributions to the field of 3D point cloud segmentation while remaining feasible within the constraints of a master's research project.

1.4 Thesis Outline

This thesis is structured to provide a comprehensive exploration of promptable, class-agnostic segmentation for 3D point clouds, with a focus on addressing the challenges posed by sparse outdoor environments. The organization of the thesis is as follows:

- **Chapter 1: Introduction**
  This chapter provides the background and context for the research, outlines the problem statement and research questions, defines the scope of the study, and presents this outline of the thesis structure.

- **Chapter 2: Literature Review**
  This chapter offers a comprehensive review of relevant literature, covering point clouds and their acquisition, traditional 3D semantic segmentation approaches, recent advancements in class-agnostic and promptable segmentation. The review identifies gaps in current knowledge and highlights areas where this research aims to make contributions, specifically promptable segmentation in 3D point clouds.

- **Chapter 3: Methodology**
  This chapter introduces RangeSAM, our innovative method for promptable segmentation of sparse outdoor LiDAR point clouds. We begin with data collection and processing using three dominant datasets. Next, we detail the key components of the architecture of RangeSAM, with each component being explained, focusing on their innovations. The chapter concludes with implementation details, covering our training setup, loss functions, data augmentation strategies, hyperparameter tuning, and evaluation metrics for model performance.

- **Chapter 4: Performance Evaluation**
  This chapter presents a comprehensive analysis of RangeSAM's performance. We start by outlining key evaluation preliminaries, including our prompt sampling strategy and metric prioritization. The chapter features four main quantitative

experiments: examining prompt influence, cross-dataset generalization, comparisons with other promptable segmentation models, and evaluations against class-agnostic segmentation models. Additionally, We provide qualitative assessments for a comprehensive visual of the model's capabilities. Finally, we explore how training datasets affect generalization and conduct an ablation study on novel architectural components.

- **Chapter 5: Conclusion**
  This chapter synthesizes the key findings of our research and their broader implications. We begin with an overview that summarizes the main results and their interpretation, addressing how they answer our research question. We then discuss the contributions of our work to the field of computer vision, particularly in 3D point cloud segmentation. Finally, we identify the limitations of our study and propose future research directions to address these constraints and further advance the field.

This structure is designed to offer a logical flow of information, from the theoretical foundations and existing work, through the development and evaluation of the proposed approach, to a critical discussion of its implications and potential future directions. Each chapter builds upon the previous ones, culminating in a comprehensive exploration of promptable, class-agnostic segmentation for 3D point clouds in sparse outdoor environments.

Chapter 2

Literature Review

2.1 Introduction to Point Clouds

Understanding the intricacies of point cloud data is crucial for developing effective segmentation methods, particularly in the context of outdoor environments. This section explores point cloud characteristics and acquisition techniques, providing essential context for the subsequent discussion on segmentation approaches.

2.1.1 Definition and Characteristics

Point clouds are a fundamental representation of 3D data, consisting of a set of points in three-dimensional space. Each point is typically defined by its spatial coordinates (X, Y, Z) and can include additional attributes such as color (R, G, B), intensity, or surface normal information [14].

This versatile and efficient way to represent complex 3D structures and environments captures intricate geometric details that might be lost in other 3D representations. Their unstructured nature allows for a more nuanced representation of spatial relationships and surface details, making them particularly valuable in applications ranging from autonomous driving and robotics to architecture and archaeology. [15].

While point clouds excel at capturing geometric information, they also present unique challenges for processing and analysis. Unlike the grid-like structure of 2D images, point clouds are inherently unordered and irregular. This characteristic, while allowing for flexible representation, complicates the application of traditional computer vision techniques and necessitates specialized algorithms for tasks such as segmentation and object detection [16].

Another important characteristic of point clouds, particularly relevant to outdoor environments, is that they may have varying point density across different regions. Point clouds can range from dense representations with high spatial resolution to sparse representations where points are more dispersed, with the density often decreasing at greater distances from the sensing device. This uneven distribution of points poses significant challenges for segmentation algorithms and is a key focus of current research. Figure 2.1 illustrates the contrast between dense and sparse point cloud representations.

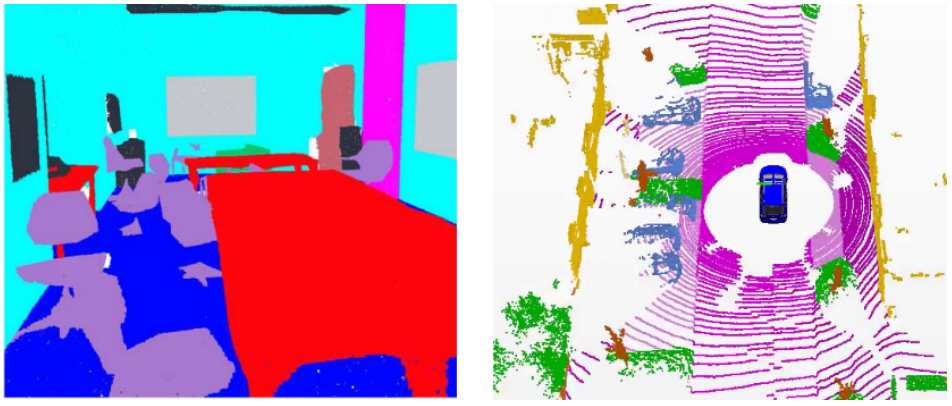

FIGURE 2.1: Comparison of Dense and Sparse Point Clouds [17, 18]

In this figure, the left image visualizes an indoor scene with uniformly high point density throughout the captured space, enabling precise geometric detail of objects and surfaces. In contrast, the right image demonstrates an outdoor scene where the point density varies considerably - objects and surfaces at increasing distances exhibit progressively lower spatial resolution, presenting fundamental challenges for consistent object recognition and segmentation across the scene.

2.1.2 Acquisition Methods

Point clouds can be acquired through various methods, each with its own strengths and limitations. The choice of acquisition method often depends on the specific application requirements, such as accuracy, scale, and speed of data collection.

2.1.2.1 LiDAR Technology

Light Detection and Ranging (LiDAR) has become one of the most prominent technologies for point cloud acquisition, especially in applications like autonomous driving and urban mapping [19]. LiDAR systems emit laser pulses and measure the time it

takes for the light to return after reflecting off surfaces in the environment. This time-of-flight (ToF) measurement is then employed to construct a precise 3D point cloud representation of the environment.

To expand on this, a typical LiDAR system consists of three main components:

1. A laser emitter that sends out rapid pulses of light
2. A receiver that detects the reflected pulses
3. A precise timing system to measure the time between emission and reception

The distance to each point in the environment is calculated using the formula:

$$\text{Distance} = \frac{\text{Speed of Light} \times \text{Time of Flight}}{2} \tag{2.1}$$

By rapidly scanning in multiple directions, a LiDAR system can build up detailed point clouds of its surroundings. Modern LiDAR systems can emit hundreds of thousands of pulses per second, allowing for the rapid acquisition of dense point clouds. Figure 2.2 illustrates the basic principle of LiDAR operation.

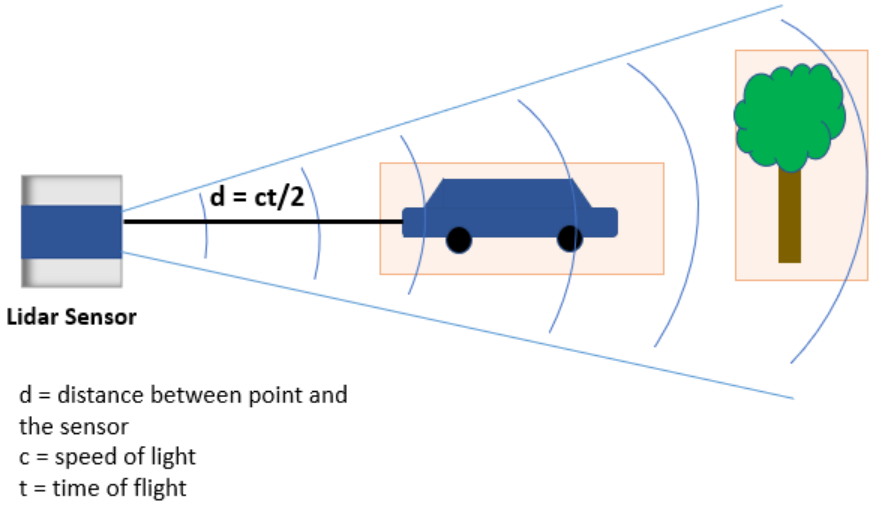

FIGURE 2.2: LiDAR Principle of Operation [20]

LiDAR systems can be broadly categorized into three types based on their deployment:

- **Terrestrial LiDAR**: Stationary systems typically used for high-resolution scanning of buildings, infrastructure, or archaeological sites. These systems offer high accuracy but are limited in their coverage area [21].

- **Airborne LiDAR**: Mounted on aircraft, these systems are used for large-scale mapping of terrain, forests, and urban areas. They are well-suited for topographic mapping and landscape modeling due to their elevated vantage point [22].
- **Mobile LiDAR**: Mounted on vehicles or handheld devices, these systems are particularly useful for mapping roads, urban environments, and indoor spaces. They offer the advantage of rapid data collection over large areas [23].

Each of these LiDAR types has its own characteristics in terms of point density, accuracy, and coverage area. Terrestrial LiDAR typically provides the highest point density but is limited in coverage area, while airborne LiDAR can cover vast areas but with lower point density. Mobile LiDAR provides an effective balance between point density and coverage area, ensuring that detailed data is captured across a wide geographical range.

One of the key advantages of LiDAR technology is its ability to penetrate vegetation to some extent, allowing for the mapping of ground surfaces even in forested areas. This capability, combined with its high accuracy and rapid data acquisition, has made LiDAR a preferred choice for many 3D mapping applications [24].

However, LiDAR also has limitations. The technology can be affected by atmospheric conditions such as rain or fog, which can scatter or absorb the laser pulses. Additionally, highly reflective or absorbent surfaces can pose challenges for accurate distance measurement [25].

2.1.2.2 Other Acquisition Techniques

While LiDAR is a dominant technology for point cloud acquisition, especially in outdoor environments, several other techniques are worth noting:

- **Photogrammetry:** this technique reconstructs 3D geometry from multiple 2D images by relying on the principle of triangulation, where the 3D position of a point can be determined by analyzing its projection in multiple images taken from different viewpoints [26]. Modern photogrammetry techniques, such as Structure from Motion (SfM), can automatically detect and match features across hundreds or thousands of images to reconstruct detailed 3D point clouds [27]. While photogrammetry can produce dense point clouds with accurate color information, it typically struggles with uniform or reflective surfaces and can be computationally intensive for large datasets.

- **Structured Light Scanning:** This technique projects a known pattern of light onto an object and analyzes the deformation of this pattern to reconstruct the 3D shape. Structured light scanners can produce high-resolution point clouds and are often used for small to medium-sized objects in controlled environments [28].
- **Time-of-Flight (ToF) Cameras:** These devices emit infrared light pulses and measure the time taken for the light to return to the sensor for each pixel. While similar in principle to LiDAR, ToF cameras capture depth information for an entire scene simultaneously, resulting in a depth image that can be converted to a point cloud. ToF cameras are often used in indoor environments and for applications requiring real-time 3D sensing [29].
- **Stereo Vision:** This technique mimics human binocular vision by using two cameras separated by a known distance. By analyzing the disparities between the two images, depth information can be calculated and used to generate a point cloud. Stereo vision systems are often used in robotics and autonomous vehicles, complementing other sensing modalities [30].

Each of these methods has its strengths and limitations in terms of accuracy, range, speed, and cost. The choice of acquisition method often depends on the specific requirements of the application, the scale of the environment, and the desired level of detail.

Photogrammetry excels in capturing detailed color information and can cover large areas, but struggles with uniform or reflective surfaces. Structured light scanning offers high resolution for small objects but is limited to controlled environments where lighting and movement can be managed effectively. ToF cameras provide real-time depth sensing but have limited range and resolution compared to LiDAR. Finally, Stereo vision is passive and can work in various lighting conditions, but its accuracy decreases with distance.

Understanding the characteristics of these acquisition techniques is valuable for developing robust segmentation algorithms that can potentially generalize across different types of point cloud data. In the context of this research, which focuses on outdoor environments and particularly driving scenes, LiDAR remains the primary technology of interest due to its ability to rapidly acquire accurate, long-range 3D data in dynamic environments.

2.2 3D Semantic Segmentation

While our research endeavors to address the challenge of promptable class-agnostic segmentation of arbitrary objects in sparse 3D point clouds, it is important to understand the substantial and relevant body of work dedicated to semantic segmentation, which involves assigning class labels to individual points or groups of points within a 3D point cloud. While our task differs from semantic segmentation in that we do not rely on predefined object classes, understanding the foundations and advancements of this segmentation approach is essential, as it provides valuable insights and techniques that inform our approach.

Figure 2.3 illustrates an example of semantic segmentation applied to a point cloud of an urban scene. Different colors represent distinct object classes, such as vehicles, roads, and vegetation.

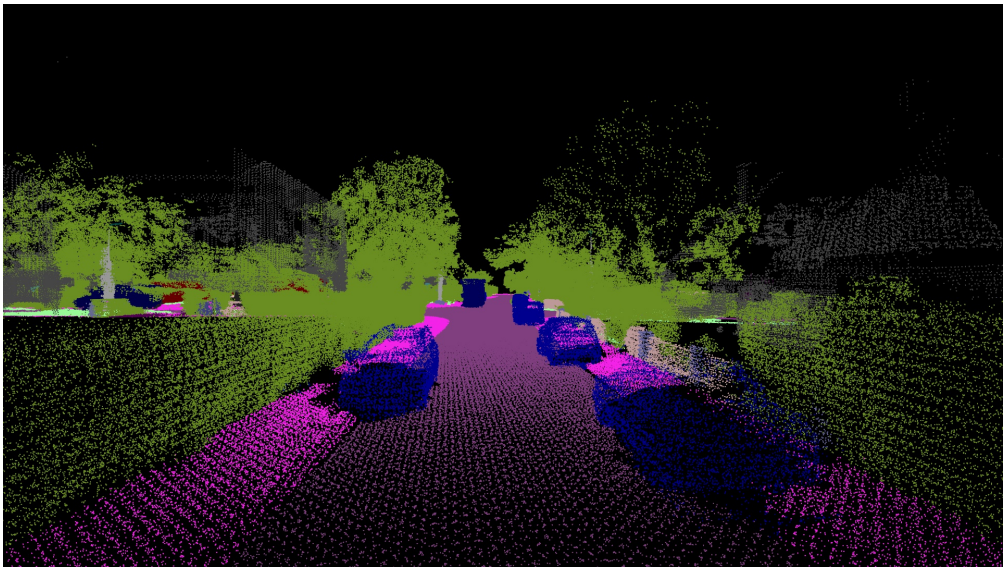

FIGURE 2.3: Semantic Segmentation of an Urban Point Cloud Scene [31]

The approaches to 3D semantic segmentation can be broadly categorized into three main paradigms: image-based, voxel-based, and point-based methods [32]. Additionally, range-based methods have emerged as a hybrid approach [33], combining elements of image-based and point-based techniques.

Each of these paradigms addresses the challenges of processing unstructured point cloud data in different ways. Understanding the strengths and limitations of each is crucial for developing effective segmentation techniques.

2.2.1 Image-Based Segmentation

Image-based methods address the challenge of unstructured point cloud data by projecting three-dimensional information onto two-dimensional image planes. This approach facilitates the application of established two-dimensional convolutional neural network (CNN) architectures, thereby leveraging the substantial advancements achieved in 2D image processing.

One of the initial works in this domain is Multi-View CNN (MVCNN) [34]. MVCNN projects the point cloud from multiple viewpoints, generating a set of 2D images. These images are then processed through a shared CNN to extract features, which are subsequently pooled to create a global descriptor. This approach effectively captures the 3D structure of objects through multiple 2D representations.

Building upon MVCNN, MVCNN-MultiRes [35] extends the multi-view concept by incorporating multi-resolution analysis. This approach uses sphere rendering of 3D shapes at different volume resolutions, alongside standard rendering. Features are extracted from these multi-resolution renderings and concatenated. The combined features are then used to train a classifier, allowing the model to leverage both detailed local information and broader contextual cues.

Another notable image-based method is SnapNet [36] which introduced a more sophisticated approach for semantic segmentation. SnapNet uses an MVCNN-like architecture to project the point cloud onto multiple virtual camera views, generating both depth and RGB images. These images are then processed by separate CNN streams, and the resulting features are fused for semantic segmentation. This multi-modal approach allows SnapNet to leverage both geometric and color information for improved segmentation accuracy.

While image-based methods benefit from the rich feature representations learned by 2D CNNs and the extensive research in 2D computer vision, they face several challenges:

1. Information loss: The projection from 3D to 2D inevitably leads to some loss of spatial information, particularly in complex scenes with occlusions.
2. View dependency: The segmentation results can be sensitive to the chosen projection views, potentially leading to inconsistencies across different viewpoints.
3. Empty Regions: In scenarios with sparse point clouds, common in long-range LiDAR scans, the projected images may contain significant empty regions, making feature extraction challenging.

Despite these limitations, image-based methods remain popular due to their computational efficiency and the ability to leverage pre-trained 2D CNN models. They are particularly effective in scenarios where the point cloud can be consistently projected onto a 2D plane.

2.2.2 Voxel-Based Segmentation

Voxel-based methods represent the 3D point cloud as a structured 3D grid of volumetric elements (voxels). Each voxel encodes information about the points falling within its volume, such as occupancy or point density. This representation allows the direct application of convolutional neural networks, enabling their success in the 3D domain.

One of the early works in voxel-based segmentation is VoxNet [37], which introduced a 3D CNN architecture for object detection and classification using voxelized point cloud data. VoxNet demonstrates the effectiveness of 3D convolutions in capturing spatial relationships within point clouds. However, it also highlighted the computational challenges associated with processing large 3D grids, especially for high-resolution or large-scale point clouds.

To address the computational limitations of dense 3D grids, OctNet [38] proposed an innovative octree-based encoding scheme. OctNet uses a hierarchical data structure that efficiently represents sparse 3D data by allocating computational resources to non-empty voxels while bypassing empty regions. This adaptive partitioning scheme allows OctNet to capture features at multiple scales, enhancing its ability to segment complex scenes.

SEGCloud [39] is another technique which further advanced voxel-based methods. It introduces an end-to-end trainable pipeline that combines voxelization, 3D CNN processing, and a novel trilinear interpolation step to map CNN outputs back to the original point cloud. This approach addresses the challenge of information loss during voxelization and ensures fine-grained segmentation results.

While voxel-based methods offer several advantages, including the ability to capture spatial context and the direct application of 3D CNNs, they also face significant challenges:

1. Quantization artifacts: The voxelization process can introduce quantization artifacts, particularly at voxel boundaries, potentially affecting segmentation accuracy.

2. Computational complexity: Processing dense 3D grids can be computationally expensive, especially for large-scale or high-resolution point clouds.

3. Memory requirements: Voxel representations, particularly at high resolutions, can have substantial memory requirements, limiting their applicability to large scenes.

Despite these challenges, voxel-based methods continue to be an active area of research, with ongoing efforts to develop more efficient and accurate architectures for processing 3D volumetric data.

2.2.3 Point-Based Segmentation

Point-based methods aim to process point clouds directly in their native, unstructured format. These approaches avoid the need for intermediate representations like 2D projections or voxel grids, preserving the inherent spatial relationships within the point cloud data. Point-based methods can be further categorized into three main subcategories: Multi-Layer Perceptron (MLP) based methods, Graph Convolutional Network (GCN) based methods, and Transformer-based methods.

- **Multi-Layer Perceptron (MLP) based Methods:** MLP-based methods use multi-layer perceptron networks to process individual points or local neighborhoods within the point cloud. These methods aim to learn point-wise features that can be aggregated to perform segmentation tasks.

  The seminal work in this category is PointNet [40], which revolutionized point cloud processing by directly operating on raw point cloud data. PointNet uses a series of MLPs to extract features for each point independently, followed by an operation which aggregates these features into a global feature vector. This architecture ensures permutation invariance, meaning the order in which points are processed does not affect the final result.

  Building on the success of PointNet, PointNet++ [41] introduced a hierarchical feature learning approach. PointNet++ addresses the limitation of PointNet in capturing local structures by applying PointNet recursively on nested partitions of the input point set. This allows the network to capture both fine-grained patterns and global context, leading to improved segmentation performance.

  RandLA-Net [42] further advanced MLP-based methods by introducing an efficient random sampling strategy coupled with local feature aggregation. This approach enables the processing of large-scale point clouds with millions of points, addressing the scalability issues faced by previous methods.

- **Graph Convolutional Network (GCN) based Methods:** GCN-based methods represent the point cloud as a graph, where each point is treated as a node,

and the relationships between points are encoded as edges. These methods apply graph convolutional operations to aggregate and propagate features across the graph, capturing both local and global contextual information.

A notable example of GCN-based methods is the Dynamic Graph CNN (DGCNN) [43]. DGCNN introduces the concept of EdgeConv, a novel convolution-like operation on graphs. Unlike fixed graph convolutions, DGCNN dynamically computes the graph at each layer of the network, allowing it to capture semantic as well as geometric neighborhoods. This dynamic graph updating mechanism enables DGCNN to adapt to varying structural patterns within the point cloud and learn more discriminative features.

Another significant contribution in this category is the Hierarchical Point-Edge Interaction Network (HPEIN) [44]. HPEIN employs a hierarchical graph convolutional network structure to process point clouds by progressively constructing a graph from coarse to fine layers. The network integrates point and edge features across multiple scales, effectively capturing geometric relationships throughout the scene. This approach incorporates point-edge interactions and edge feature supervision, leading to improved segmentation performance on multiple benchmarks.

- **Transformer-Based Methods:** Transformer-based methods, inspired by their success in natural language processing and 2D computer vision, have gained traction in point cloud segmentation due to utilizing attention mechanisms. These mechanisms enable models to dynamically focus on relevant parts of the input data, weighing the importance of different points or features.

  Self-attention is a key component of transformers which allows each point to interact with all other points, capturing global relationships and long-range dependencies, which is crucial for understanding overall structure and context in point cloud segmentation.

  Point Transformer [45] is a pioneering work in this category, adapting the transformer architecture to point cloud data. It introduces a vector self-attention layer that adaptively modulates feature channels by computing attention weights based on point-wise geometric relationships. The multi-head attention mechanism enables Point Transformer to effectively process geometric features at varying scales, leading to robust segmentation performance.

  Point Cloud Transformer (PCT) [46] further refines the transformer-based approach for point clouds. PCT introduces a novel offset-attention module that explicitly models geometric relationships between points. This module, combined with a neighbor embedding layer, allows PCT to better capture detailed structural

patterns while leveraging the broad receptive field characteristic of transformer architectures.

Transformer-based methods have shown remarkable performance in point cloud segmentation tasks, often surpassing previous state-of-the-art models. Their ability to capture both local and global context with high accuracy makes them particularly effective for complex scenes with varying scales of objects.

Figure 2.4 illustrates an example architecture of a generic point-based semantic segmentation network, showcasing how these methods process raw point cloud data to produce segmentation results.

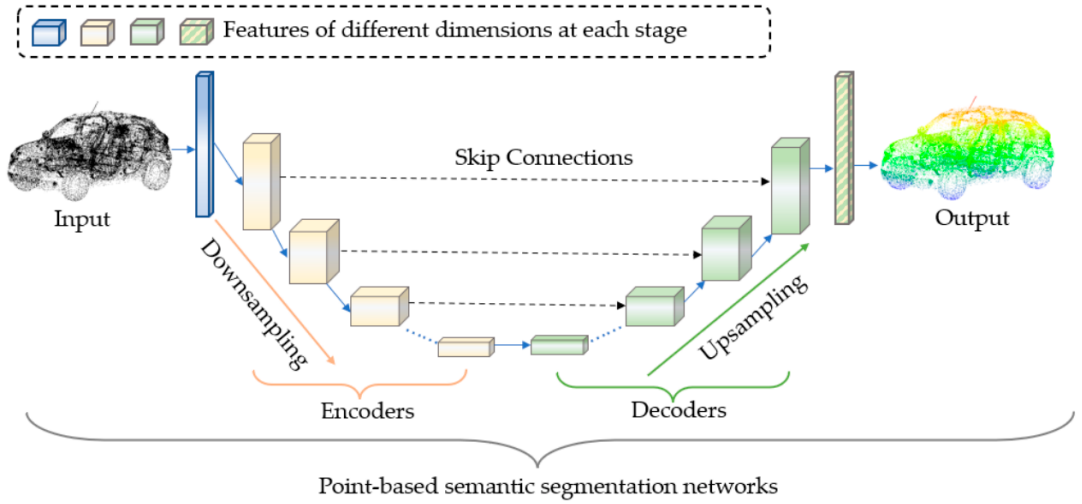

FIGURE 2.4: General Structure of Point-Based Networks [32]

Point-based networks typically process an input point cloud through multiple stages, transforming and analyzing the data to produce a semantically segmented output. The architecture enables the network to understand and label the 3D structure of the input data.

While point-based methods have shown impressive results, they also face certain challenges:

1. Computational complexity: Processing large point clouds can be computationally expensive, especially for methods that consider all pairs of points, which is especially prevalent in many Transformer-based approaches.

2. Handling varying point densities: Point clouds often have non-uniform densities, which can affect the performance of methods that rely on fixed-size local neighborhoods.

3. Feature aggregation: Integrating local and global features remains a major challenge due to the unstructured nature of point clouds, making it difficult to maintain spatial relationships while seamlessly combining information across different scales.

Despite these challenges, point-based methods continue to be at the forefront of 3D semantic segmentation research, with ongoing efforts to improve their efficiency, scalability, and accuracy.

2.2.4 Range-Based Segmentation

Range-based segmentation methods offer a hybrid approach that merges elements of both image-based and point-based techniques by utilizing range sensors such as LiDAR or RGB-D cameras. These methods produce a structured 2D representation of the 3D scene while retaining depth information. This approach is highly relevant to our research, as it forms the basis of our architecture and supports our work on promptable, class-agnostic segmentation.

The key advantage of range-based methods is their ability to maintain a structured representation of the point cloud, similar to image-based methods, while preserving the full 3D information. This approach allows for the utilization of well-established 2D processing techniques, benefiting from the extensive research in 2D computer vision, while still retaining crucial 3D data. It is especially effective for processing LiDAR data in autonomous driving scenarios, where the sensor's native output can be directly utilized without the need for complex point cloud transformations, streamlining the segmentation pipeline.

SqueezeSeg [47] exemplifies this approach, projecting LiDAR point clouds onto a spherical surface to create a 2D range image. Each pixel in this image contains depth, intensity, and coordinate information, preserving the LiDAR scan's geometry. SqueezeSeg processes this representation using a lightweight 2D CNN, enabling real-time segmentation crucial for autonomous driving. It also incorporates a Conditional Random Field (CRF), a probabilistic graphical model that considers contextual information, for post-processing, refining results by considering spatial relationships between points.

Building upon this concept, RangeNet++ [48] introduced several improvements to the range-based segmentation approach. It employs a more sophisticated CNN architecture and incorporates a novel post-processing step called kNN-based label transfer, which leverages the k-Nearest Neighbors algorithm to refine segmentation results by propagating labels from the 2D range image back to the 3D point cloud, addressing some of the information loss inherent in the projection process.

RangeViT [49] is a network which represents a significant advancement in range-based segmentation by incorporating transformer architectures. It uniquely combines efficient range image processing with Vision Transformers (ViT) [11], which are neural network architectures that excels at learning image features through self-attention mechanisms. RangeViT operates by first projecting 3D point clouds to 2D range images, then applying ViT to encode these images, followed by processing the obtained image features and finally re-projecting these features to the 3D space for refined segmentation.

This hybrid approach allows RangeViT to effectively capture both local geometric details and global contextual information in LiDAR point clouds. By leveraging self-attention mechanisms, RangeViT can model long-range dependencies across the entire scene, a crucial ability for understanding complex urban environments. Moreover, its architecture is designed to handle the sparsity inherent in LiDAR data, making it particularly effective for long-range sensing scenarios.

Figure 2.5 illustrates the overall architecture of RangeViT, showcasing its innovative fusion of range-based processing and transformer technology.

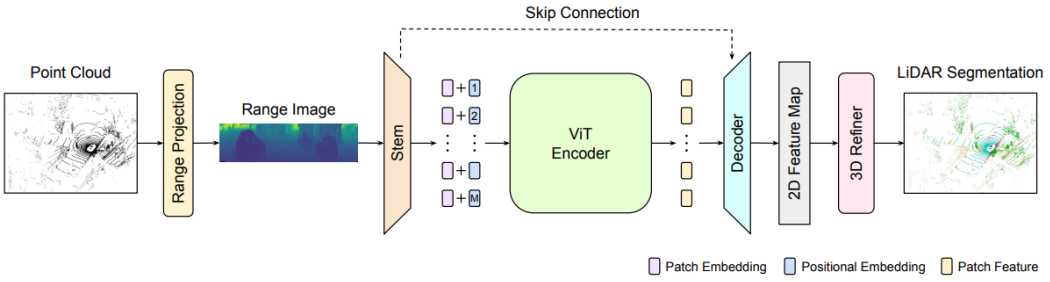

FIGURE 2.5: Architecture of RangeVIT [49]

As shown in this figure, RangeViT consists of several key components:

1. Range Projection: The point cloud is projected onto a 2D plane, creating a range image that preserves depth, intensity, and coordinate information.

2. Convolutional Stem: Initial feature extraction is performed using convolutional layers. This stem processes the range image to create a lower-resolution feature map, reducing spatial dimensions while increasing channel depth. This step is crucial for preparing the input for the ViT encoder.

3. ViT Encoder: A Vision Transformer (ViT) processes the feature map from the convolutional stem. It divides the feature map into patches, applies positional encodings, and uses self-attention mechanisms to capture long-range dependencies and global context across the entire scene.

4. Decoder: The decoder progressively upsamples the features from the ViT encoder. It combines these with skip connections from the convolutional stem, preserving fine-grained spatial information. This produces a detailed 2D feature map that captures both global context and local details.

5. 3D Refiner: This layer maps the 2D features back to the original 3D point cloud. It uses the preserved coordinate information to accurately place features in 3D space, refining the segmentation results and ensuring they align with the original point cloud structure.

RangeViT offers several key advantages over previous range-based methods. Its transformer architecture effectively addresses the sparsity common in LiDAR data, enabling efficient processing of large-scale point clouds. The self-attention mechanisms in the ViT encoder enhances global context awareness, allowing the model to capture long-range dependencies across the scene. Furthermore, the integration of range projection and 3D refinement ensures that the 3D structural information is preserved and utilized throughout the segmentation process, maintaining the integrity of the original point cloud data.

The success of RangeViT in semantic segmentation tasks establishes a robust foundation for addressing LiDAR data in 3D point cloud segmentation. Although generally range-based methods may experience challenges with occlusions and potential loss of fine-grained 3D details, their efficiency in processing LiDAR data makes them ideal for enhancing segmentation performance in outdoor driving environments. Our methodology, detailed later, advances beyond the RangeViT framework by introducing significant architectural refinements and novel mechanisms, creating a highly effective promptable model capable of accurately segmenting diverse objects in complex 3D outdoor scenes.

2.3 3D Class-Agnostic Segmentation

Class-agnostic segmentation represents a significant paradigm shift in the field of computer vision and 3D point cloud processing. Unlike traditional semantic segmentation, which relies on predefined object categories, class-agnostic approaches aim to segment arbitrary objects without the constraints of a fixed set of classes. This capability is particularly crucial in dynamic and unpredictable environments, such as those encountered in autonomous driving scenarios, where the variety of objects can range from pedestrians and vehicles to unexpected obstacles.

As briefly touched upon in the background chapter, the importance of class-agnostic segmentation lies in its potential to handle unknown or novel objects—an essential requirement for robust perception systems. Real-world applications, especially in outdoor environments, are fraught with challenges, making it impossible to anticipate and train for every possible object category. Class-agnostic methods address this limitation by focusing on the fundamental task of object delineation, allowing systems to identify and segment objects based solely on their shape, size, and spatial context, rather than their predefined labels.

The shift towards class-agnostic segmentation is driven by several key factors:

1. Generalization: Class-agnostic models can potentially handle a wider range of scenarios and object types, including those not seen during training. This broad generalization is particularly valuable in situations where environmental conditions change rapidly, or where new object types emerge unexpectedly.

2. Reduced Annotation Burden: Traditional segmentation methods often require extensive, class-specific labeled datasets, which are time-consuming and expensive to create. Class-agnostic approaches alleviate this need, allowing for effective segmentation without the overhead of exhaustive labeling, thus streamlining the data preparation process.

3. Adaptability: In rapidly changing environments, class-agnostic models can more easily adapt to new object types without requiring retraining. This flexibility is crucial in applications like robotics and autonomous vehicles, where the operational landscape can evolve in real time.

4. Open-set Recognition: Class-agnostic segmentation aligns well with the concept of open-set recognition, where systems must handle unknown classes during inference. This capability ensures that the model can effectively manage uncertainties, essential for applications where safety and reliability are paramount.

In this section, we will examine the evolution of the field towards class-agnostic approaches in 3D. Followed by this, we will analyze recent advancements in promptable segmentation and their role in expanding the capabilities of point cloud processing. These developments facilitate more sophisticated interactions and enhance performance across a variety of operational contexts.

2.3.1 Evolution from Semantic to Class-Agnostic Segmentation

The transition from semantic to class-agnostic segmentation in 3D point clouds has been gradual, with researchers exploring various approaches to overcome the limitations of traditional semantic segmentation methods. This evolution has been driven by the need for more flexible and generalizable models capable of handling the complexity and diversity of real-world 3D data.

Early attempts at class-agnostic segmentation in 3D point clouds often extended ideas from 2D image segmentation. The concept of instance segmentation, where individual object instances are identified regardless of their class, provided a stepping stone towards fully class-agnostic approaches.

SGPN (Similarity Group Proposal Network) [50] marked an important early development in this direction. While not fully class-agnostic, SGPN introduced the idea of grouping points based on their similarity in learned feature space rather than predefined classes. This approach uses a novel similarity matrix to indicate whether pairs of points belong to the same object instance, allowing for the segmentation of individual object instances without initially relying on a fixed set of categories.

Building on these foundations, more recent methods have pushed further towards true class-agnostic segmentation. LRGNet [51] introduced a novel framework that combines deep learning with traditional region growing techniques. The key idea behind LRGNet is to learn a function that directly predicts which points to add or remove from the current region at each step of the region growing process. This approach allows the model to segment objects of any class using a single neural network, without making assumptions about their shapes or sizes.

Advancing the field further, 3DUIS [6] proposed an unsupervised class-agnostic instance segmentation method specifically designed for LiDAR data. This approach leverages self-supervised learning to extract point-wise features, which are then used to build a graph representation of the point cloud. By applying an efficient graph-based optimization technique to this representation, 3DUIS achieves instance segmentation without requiring any labels or semantic background removal. This method demonstrates the potential of combining unsupervised learning with classical graph-based techniques for class-agnostic segmentation in challenging real-world scenarios.

The Region Transformer [52] further advanced the concept of learnable region growing by introducing a transformer-based architecture for class-agnostic 3D segmentation. This method leverages self-attention mechanisms to capture long-range dependencies within point clouds. By iteratively expanding or contracting a region through the addition

or removal of points, the Region Transformer is guided by learned attention weights, allowing for more flexible and adaptive segmentation. This approach bridges the gap between traditional region growing methods and modern transformer architectures, capturing complex spatial relationships within the point cloud and leading to more accurate and flexible segmentation results.

These methods mark key steps in 3D class-agnostic segmentation. SGPN introduced similarity-based grouping, LRGNet developed adaptive region growing, 3DUIS pioneered unsupervised graph-based segmentation, and Region Transformer applied attention mechanisms. Each approach tackled different aspects of the challenge, advancing the field's capabilities and pushing towards more robust and versatile class-agnostic segmentation methods.

Despite these advances, these approaches still face limitations in terms of generalization to truly arbitrary objects and the ability to handle guided interactions, which are particularly critical in dynamic environments. This led to increased interest in promptable segmentation methods, which aim to provide more flexible and interactive ways of specifying segmentation targets.

2.3.2 Promptable Segmentation

Promptable segmentation represents a significant advancement in the field of computer vision, offering a more flexible and interactive approach to image and point cloud segmentation. This method allows users to guide the segmentation process through various forms of input prompts, such as points, bounding boxes, or even natural language descriptions [8]. The ability to interactively specify objects of interest makes promptable segmentation particularly valuable in scenarios where the target objects may not be known in advance or where fine-grained control over the segmentation process is desired.

The key motivation behind promptable segmentation is to create more versatile and user-friendly segmentation systems while reducing annotation burden. Traditional semantic segmentation models are limited to predefined object categories and struggle with novel or ambiguous objects. In contrast, promptable segmentation models can adapt to user intent on-the-fly, segmenting arbitrary objects based on minimal input. This approach often allows for training with weaker supervision, potentially reducing the need for exhaustive per-point labels in training data.

It is important to note that while user input is a key feature of promptable segmentation, these models can also be designed to operate autonomously. By simulating user prompts

or leveraging other contextual information, promptable models can function seamlessly without direct human guidance. This capability integrates promptable segmentation into automated systems, combining flexibility and efficiency.

Promptable segmentation works by encoding user-provided or simulated prompts into a format that can be processed alongside the input image or point cloud. The model then uses this information to guide its attention towards the relevant regions or objects. This approach allows for a more dynamic and interactive segmentation process, allowing for refining results in real-time by providing additional prompts.

The success of promptable segmentation in 2D image processing, has sparked significant interest in extending these capabilities to 3D point cloud data. However, this extension presents unique challenges due to the unstructured nature of point clouds and the increased complexity of 3D spatial relationships. Despite these challenges, the potential benefits of promptable segmentation in 3D are substantial, particularly for applications in robotics, autonomous driving systems, and virtual reality.

2.3.2.1 Segment Anything Model (SAM) and its Utilization in 3D

The Segment Anything Model (SAM), introduced by Meta AI Research [8], represents a significant breakthrough in the field of promptable, class-agnostic segmentation. While SAM was primarily designed for 2D image segmentation, with it leveraging 2D transformer-based architectures and self-attention mechanisms, its innovative approach and impressive performance have inspired numerous adaptations and extensions to 3D point cloud segmentation. To fully grasp SAM's potential for point cloud segmentation, it is essential to understand its inner workings and how it is being utilized for this purpose.

Figure 2.6 illustrates an overview of the model's architecture.

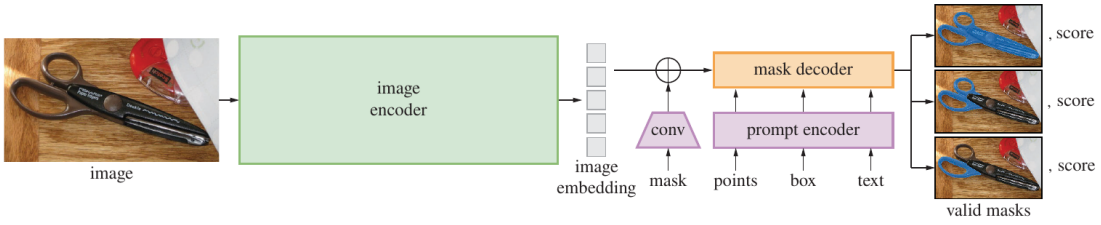

FIGURE 2.6: Segment Anything Model Architecture

The architecture of SAM consists of three main components: the image encoder, prompt encoder, and mask decoder. The image encoder, utilizes a Vision Transformer (ViT) to

process the entire input image to create a rich, high-dimensional feature representation. This process, while computationally intensive, allows for a comprehensive understanding of the image content that can be flexibly queried by various prompts.

The prompt encoder transforms user prompts (e.g. points and bounding boxes) into vector embeddings, enabling the model to comprehend and identify the designated object of interest. For instance, a point is represented as the sum of a positional encoding of its location and a learned embedding indicating whether it belongs to the foreground (object of interest) or background. This process is carried out by passing the prompt through a small neural network that maps the prompt elements to their corresponding vector representations.

Finally, The mask decoder is a module that efficiently combines the image embedding from the encoder and the prompt embeddings to produce an output mask. It takes inspiration from Transformer segmentation models [53], featuring a modified Transformer decoder with cross-attention mechanisms. The decoder updates both the image embedding and prompt tokens through cross-attention, allowing for effective integration of the two sources of information. After upscaling the updated image embedding, the output token is used to dynamically predict the final mask through a small multi-layer perceptron.

SAM has been trained on a large and diverse dataset of 11 million images and 1.1 billion masks, generated using its own data engine, which enables it to demonstrate strong zero-shot performance on various segmentation tasks, accurately segmenting objects in new images it has never seen before.

It is important to note that user prompts, such as points and boxes, are not part of the dataset, and are dynamically created based on ground truth instance masks while training the model. To elaborate further, during training, SAM simulates an interactive setup by sampling prompts in multiple rounds per mask, allowing it to learn from increasingly refined inputs that mimic user interactions. For inference, users can input their own prompts to segment specific objects of interest. However, if the goal is to segment everything in the image, SAM overlays a grid of point prompts on the image to attempt to segment all existing objects.

The success of SAM in 2D image segmentation has naturally led researchers to explore its potential applications in 3D point cloud segmentation. Several innovative approaches have emerged, each attempting to leverage SAM's capabilities in the 3D domain while addressing the unique challenges of point cloud data.

One notable approach, SAM3D [54], extends SAM to 3D point cloud segmentation. In the context of a 3D scene, complete with a collection of RGB images and corresponding

pose data, SAM3D leverages the base SAM model to generate segmentation masks for each individual RGB image. These 2D masks are then projected onto the 3D point cloud, resulting in a series of 3D masks. In the final step, these 3D masks are merged together using a bottom-up approach, iteratively combining the masks to produce a comprehensive, final mask for the entire point cloud.

SAMPro3D [55] presents another approach which also leverages the base SAM model for 3D point cloud segmentation without further training. It initializes 3D prompts within the input scene and projects them onto 2D frames to generate pixel prompts for SAM. This is followed by employing a 2D-guided prompt filtering mechanism to retain high-quality prompts based on their performance across all frames. A prompt consolidation strategy further refines the results by merging prompts likely segmenting the same object. Finally, all input points are projected onto the segmented frames, and their mask predictions are aggregated to produce the final 3D segmentation.

Segment3D [56] takes a different approach from the previous two by incorporating SAM into its training pipeline rather than using it directly for inference. It employs a two-stage training approach to generate class-agnostic 3D segmentation masks without manual annotations. Initially, Segment3D pre-trains a 3D segmentation model on partial RGB-D point clouds, using SAM-generated 2D masks as pseudo ground-truth labels. Then, the pre-trained model is fine-tuned on complete 3D point clouds in a self-supervised manner, using its own high-confidence mask predictions as the training signal. This approach avoids the merging process and directly infers segmentation masks for entire 3D scenes using a native 3D model.

These adaptations of SAM to 3D point cloud segmentation demonstrate the model's potential impact beyond its original 2D domain. However, they also highlight some of the challenges in directly utilizing pre-trained 2D methods for 3D data. These challenges include handling occlusions, ensuring consistency across multiple views, capturing the full 3D structure of objects, and bridging the gap between 2D image-based training and the unique properties of 3D point clouds. Additionally, These methods typically require more input than just the point clouds, such as RGB images or camera pose data, which can be a significant limitation in scenarios where only point cloud data is available.

These limitations motivate the development of 3D promptable segmentation methods that directly operate on point cloud data while maintaining the flexibility and generalizability demonstrated by SAM in the 2D domain.

2.3.2.2 Native 3D Promptable Architectures

Recognizing the limitations of adapting 2D models such as SAM to 3D data, researchers have begun developing native 3D promptable segmentation methods. These approaches aim to combine the flexibility and generalization capabilities of promptable models with architectures specifically designed for point cloud data. Generally, these methods use points as prompts, as points are the natural atomic unit of 3D point cloud data and provide a simple, intuitive way for users to indicate regions of interest by clicking in the 3D space.

InterObject3D [13] is a 3D promptable segmentation approach that refines results through direct interaction with 3D point clouds. It encodes user clicks as 3D volumes, representing positive and negative feedback as additional input channels. This enables seamless integration with existing 3D segmentation architectures, such as convolutional networks based on the Minkowski Engine [57]. InterObject3D generalizes well to new environments and object classes not seen during training, making it suitable for annotating new datasets or adapting to novel scenarios.

In InterObject3D, click simulation is utilized for training and evaluation, mimicking realistic user behavior. During inference, the model targets the largest error region, ensuring reproducible evaluation. Additionally, InterObject3D features online adaptation, treating user corrections as training examples to update the model during testing, further improving generalization to unseen classes and datasets.

Agile3D [10] takes a different approach to 3D promptable segmentation by focusing on efficient and adaptive processing of point cloud data. Unlike previous methods that segment objects sequentially, Agile3D can segment multiple objects simultaneously within a 3D scene. The model employs a click-as-query module that converts user clicks into spatial-temporal query vectors, encoding both the spatial location and temporal order of user interactions. This is combined with a click attention module that enables explicit interactions between click queries and the 3D scene features.

A key innovation of Agile3D is its ability to pre-compute backbone features for a scene, allowing for faster inference during interactive segmentation. The model only needs to run a lightweight decoder for each iteration of user input, significantly reducing computation time compared to methods that require a full forward pass for each interaction. Agile3D also introduces an iterative training strategy that more closely mimics real user behavior, improving the model's performance in practical scenarios.

PointSAM [9] is another 3D promptable segmentation method that draws significant inspiration from the SAM architecture, with it being specifically tailored for 3D point

clouds. Unlike methods that rely on projecting 2D SAM outputs to 3D, PointSAM operates directly on point cloud data. Similar to SAM, its architecture consists of three main components, as illustrated in Figure 2.7.

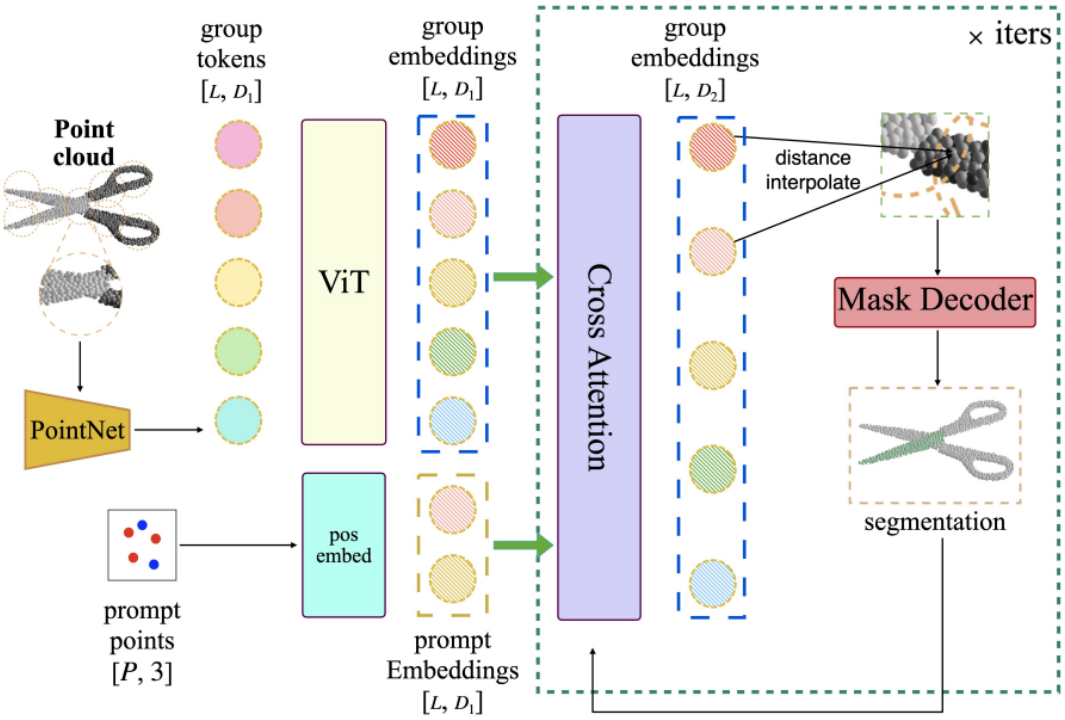

FIGURE 2.7: Overview of Point-SAM Architecture

In PointSAM, The Point Cloud Encoder first samples a fixed number of centers using Farthest Point Sampling (FPS) and groups k-nearest neighbors into patches. These patches are processed through a PointNet [40] and a pre-trained transformer from Uni3D [58] to generate the point-cloud embedding. The Prompt Encoder handles both point and mask prompts, encoding them into consistent representations. The Mask Decoder, a key component, uses transformer decoder blocks with prompt self-attention and cross-attention in both directions between prompts and point-cloud embeddings. It then upsamples the point-cloud embedding and applies a dynamic linear classifier to generate the final segmentation mask. PointSAM can handle variable input sizes and produce multiple output masks, addressing many limitations of 2D projection methods. This approach allows for more accurate and consistent 3D segmentation results, including the ability to segment internal structures of 3D objects.

While these native 3D promptable segmentation methods represent significant advancements in the field, they generally focus on dense point clouds and often use datasets that feature relatively complete object representations. This focus on dense data is understandable given the prevalence of such datasets in indoor scenarios and many computer

vision benchmarks. However, in real-world applications, particularly in outdoor driving scenes, point clouds are often sparse due to the nature of LiDAR sensor data acquisition. This sparsity poses unique challenges that are not fully addressed by current approaches. The varying point density, especially at long ranges, and the potential for occlusions in LiDAR scans create additional complexities for segmentation tasks.

As we transition to our methodology, we propose a novel approach that specifically targets these challenges associated with sparse outdoor LiDAR point clouds. By adapting promptable segmentation techniques to work effectively with sparse data, we aim to bridge a critical gap in the field and provide a more robust solution for real-world autonomous driving applications.

Chapter 3

Methodology

Our research introduces a novel approach to promptable segmentation for sparse outdoor LiDAR point clouds, addressing key challenges and building on insights identified in the literature review. At the core of our methodology is RangeSAM, a framework that combines the efficiency of range-based processing with the flexibility of promptable segmentation, specifically designed to address the unique challenges posed by sparse data in autonomous driving scenarios.

RangeSAM leverages insights from the RangeViT architecture while introducing novel promptable segmentation capabilities inspired by SAM for interactive and class-agnostic segmentation. Through a unique adaptation of these concepts to the 3D domain and specialized handling of sparse LiDAR data, RangeSAM addresses a critical gap in 3D point cloud segmentation.

In this chapter, we detail our methodological approach, beginning with how the data used in RangeSAM is gathered and processed, followed by a comprehensive description of RangeSAM's architecture and functionality. We will then discuss the implementation details, including our training setup, the metrics utilized for evaluation, and the hyperparameter tuning process.

3.1 Data Collection and Processing

The development of a promptable, class-agnostic segmentation model for 3D point clouds necessitates the use of outdoor LiDAR datasets with instance-level annotations. These instance labels are crucial for training the model, as they allow for the creation of point cloud object-instance pairs where the object of interest is designated as the "foreground" and the rest of the scene as the "background". Additionally, point prompts, categorized

as either foreground points or background points, serve as an essential input to guide the segmentation process. The specifics of how these prompts are utilized and encoded within the RangeSAM model will be elaborated upon in subsequent sections.

To facilitate this research, we conducted a comprehensive analysis of available outdoor LiDAR datasets, focusing on those that provide instance-level annotations. Table 3.1 presents a comparison of three prominent datasets in this domain: SemanticKITTI [18], KITTI-360 [31], and ApolloScape[59].

Dataset	**SemanticKITTI**	**KITTI-360**	**ApolloScape**
Location	Karlsruhe, Germany	Karlsruhe, Germany	Multiple cities, China
Total number of scans	43,552	80,000	10,995
Provides raw scan labels	Yes	No	Yes
Number of labeled scans	20,351	-	5,594
Number of semantic classes	28	45	6
Provides instance labels	Yes	Yes	Yes
Number of instance classes	8	21	6
Total driving distance (km)	39.2	73.7	-
360° coverage	Yes	Yes	Yes
Temporal consistency	Yes	Yes	No

TABLE 3.1: Dataset Comparison - SemanticKITTI, KITTI-360, and ApolloScape

This table offers a comparative overview of key characteristics for each dataset, including the total number of scans, availability of raw scan labels, number of semantic and instance classes, and other relevant features. While all three datasets provide valuable information, they each possess unique attributes that influence their suitability for our specific research objectives.

SemanticKITTI, derived from the original KITTI dataset [3], captures semi-urban environments in Karlsruhe, Germany. It offers a substantial number of labeled scans (20,351) and provides instance labels for 8 classes. The dataset's strength lies in its temporal consistency and raw scan labeling, making it particularly useful for tasks requiring frame-to-frame coherence.

Figure 3.1 illustrates a typical SemanticKITTI scan, where different colors represent distinct object instances. This visualization underscores the dataset's capability to differentiate between individual objects within the same class, a crucial feature for instance-level segmentation tasks.

ApolloScape, collected across multiple cities in China, offers a diverse range of urban environments. While it provides fewer labeled scans (5,594) compared to SemanticKITTI, it still offers instance labels for 6 classes. The dataset's strength lies in its geographical diversity, potentially enhancing the robustness of models across different urban layouts.

FIGURE 3.1: SemanticKITTI Scan With Object Instances Color-Coded

KITTI-360, which can be considered a successor to the original KITTI dataset, stands out for its comprehensive and detailed annotations. Captured in suburban environments of Karlsruhe, Germany, it provides the largest number of scans (80,000) among the three datasets. KITTI-360 offers instance labels for an impressive 21 classes, significantly more than SemanticKITTI or ApolloScape. This extensive class coverage includes detailed instance annotations for common objects found in urban datasets, such as cars and pedestrians, while also incorporating labels for less frequently represented classes such as poles, traffic lights, and buildings. This diversity makes it particularly suitable for fine-grained segmentation tasks in complex urban scenes.

However, KITTI-360 presents a unique challenge: it creates dense scenes from raw LiDAR scans rather than providing labels for individual raw frames. The dataset aggregates multiple LiDAR scans to create denser point clouds, which are then annotated. While this approach results in more comprehensive and detailed annotations, it also means that KITTI-360 does not directly provide labels for raw LiDAR scans, as reflected in our dataset comparison table. This characteristic necessitates an additional step to recover labels for individual raw LiDAR frames.

To address this challenge, we utilized a partial re-implementation of the raw 3D scan accumulation algorithm from the KITTI-360 development kit [60]. This tool facilitates the recovery of instance labels for individual raw LiDAR scans from the aggregated and annotated point clouds. By leveraging this method, we successfully extracted labels for 64,640 individual LiDAR scans, significantly exceeding the number of labeled scans available in SemanticKITTI or ApolloScape datasets. This approach allows us to harness the comprehensive annotations of KITTI-360 while retaining the granularity of individual LiDAR frames.

To effectively process point cloud data for model training and evaluation, we developed a preprocessing pipeline detailed in Algorithm 1. The pipeline transforms raw point clouds into binary instance masks, efficiently encoding which points belong to each unique object instance.

Algorithm 1 Data Preprocessing for 3D Point Clouds

Input: Point cloud data with instance labels

Output: Sparse matrix files containing binary masks of each instance

Step 1: Identify Unique Instances

For each scan in the dataset:

Extract distinct instance labels from the point cloud.

Step 2: Create Sparse Binary Masks

For each unique instance in the scan:

Generate a binary mask indicating the presence (1) or absence (0) of the instance at each point of the scan.

Step 3: Save Instance Masks

For each binary mask:

Save the mask as a compressed sparse matrix for storage effeciency and quick loading.

This preprocessing step is crucial as it transforms the raw point cloud data, into a format more suitable for training our promptable segmentation model. By separating instances and creating binary masks, we enable the model to focus on individual objects during training, facilitating the learning of class-agnostic segmentation.

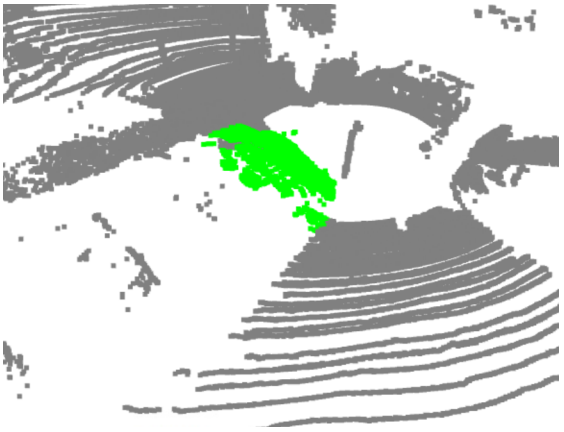

FIGURE 3.2: Preprocessed KITTI-360 Scan With Instance Highlight

Figure 3.2 illustrates the outcome of this preprocessing on a KITTI-360 scan. The highlighted green points represent a single object instance (e.g., a car) isolated from the

scene, while the gray points constitute the background. This visualization demonstrates the effectiveness of our instance separation process in complex urban environments, showcasing the refined data structure that will serve as input for our model during the training process.

While we applied this preprocessing pipeline to all three datasets (KITTI-360, SemanticKITTI, and ApolloScape), we selected KITTI-360 as our primary training dataset due to its larger scale, more diverse instance classes, and higher level of detail in annotations. Following standard machine learning practices [61], we partitioned the KITTI-360 dataset using an 80-10-10 split: 80% for training, 10% for validation to monitor performance and guide hyperparameter tuning, and 10% for final testing on unseen data.

To evaluate our model across different urban environments, we leveraged the preprocessed SemanticKITTI and ApolloScape datasets as strategic external test sets. SemanticKITTI's semi-urban environments allow us to assess performance across varied urban densities, while ApolloScape's Chinese cityscape data enables evaluation across distinctly different urban planning paradigms and architectural styles, allowing for more nuanced assessment of model generalizability.

This comprehensive data collection and processing approach provides a solid foundation for the development and evaluation of our model. By leveraging the strengths of multiple datasets and employing advanced preprocessing techniques, we aim to create a robust and generalizable promptable segmentation model capable of handling the complexities of outdoor LiDAR point clouds.

3.2 RangeSAM

RangeSAM introduces a transformative approach to promptable, class-agnostic segmentation designed specifically for sparse outdoor LiDAR point clouds. While incorporating proven concepts from RangeViT and SAM, RangeSAM presents distinct innovations that address the fundamental challenges of processing sparse LiDAR data in promptable settings. This section details RangeSAM's unique architecture and key components that enable effective promptable segmentation of 3D point cloud data.

At its core, RangeSAM redefines the semantic segmentation framework of RangeViT, adapting it to the task of promptable, class-agnostic segmentation. This adaptation involves re-framing the multi-class semantic segmentation task as a binary classification problem. In this context, the model is trained to distinguish between "foreground" points, which represent the object of interest, and "background" points, which encompass all other points in the scene. By adopting this binary classification approach,

RangeSAM is capable of segmenting arbitrary objects based on user-provided prompts. This key feature enables class-agnostic segmentation, allowing the model to effectively isolate and identify objects of interest regardless of their specific class.

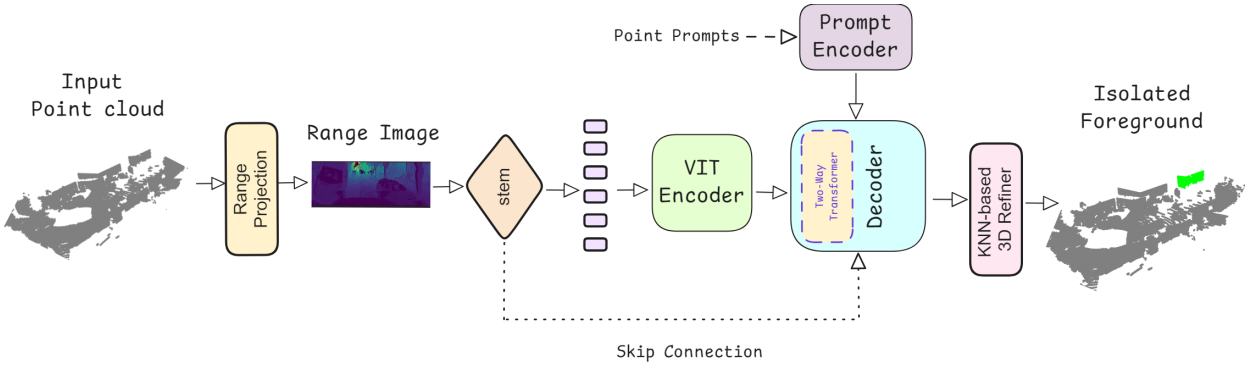

FIGURE 3.3: RangeSAM Architecture

Figure 3.3 provides a high-level illustration of the architecture of RangeSAM. The diagram showcases how an input point cloud is processed through various stages to produce an isolated foreground object based on prompts which are provided to the model. For example, the model can employ these prompts to indicate the intention of isolating a building within the scene, directing the model to accurately separate the points of that specific building from the rest of the scene. This visual representation underscores RangeSAM's ability to perform targeted, prompt-guided segmentation on arbitrary objects within a point cloud.

Our RangeSAM model can be conceptually divided into five main components: (1) Instance-Selection and Range Projection, (2) Image Encoder (comprising the Convolutional Stem and ViT Encoder), (3) Prompt Encoder, (4) Bi-Directional Transformer-Integrated Decoder, and (5) KNN-based 3D Refiner. Each of these components plays a crucial role in transforming the input point cloud and prompts into an accurate segmentation mask. A detailed explanation of each component will follow, highlighting their specific functions and contributions to the overall model.

3.2.1 Instance Selection and Range Projection

The RangeSAM architecture begins with two essential steps: instance selection and range projection, which is accompanied by a cropping mechanism. These processes transform raw 3D point cloud data into a focused, 2D representation that forms the foundation for segmentation tasks.

The instance selection process identifies specific objects to focus on within each point cloud scan. For each scan, all object instances present are identified. A class is then

randomly selected from those available, followed by a random instance from that class. This selection forms a scan-instance pair, serving as a single training example. The range projection step subsequently transforms this 3D instance into a 2D representation, which is fed into the segmentation network.

This approach ensures that our model learns to segment objects effectively by focusing on individual instances. A significant benefit of this method is its ability to address class imbalance, where one class may dominate others in the dataset. By randomly selecting a class for each scan, we adopt a well-established technique for handling class imbalance [62], ensuring that all classes have an equal opportunity to be represented in the training process, regardless of their frequency in the original scans.

While this method ensures that each training sample contains a significant object of interest, it does present a potential limitation, in that a considerable amount of data from each scan may be overlooked. However, we mitigate this concern through two key factors. First, we utilize an extensive dataset with a large number of scans. Second, our training regimen, which will be detailed in the subsequent training setup section, involves many epochs. This combination ensures that over the course of training, the model is exposed to a diverse and comprehensive set of object instances and contexts, despite the focused nature of each individual training example.

Following the instance selection, we employ range projection to transform the 3D point cloud into a structured 2D format. This projection technique converts unstructured point cloud data into a dense image-like representation where each pixel encodes spatial and intensity information from the original 3D scene. This approach not only preserves essential geometric relationships but also enables the use of powerful 2D convolutional architectures.

The range projection process is based on the spherical projection model [48], which maps 3D points onto a 2D grid based on their angular positions relative to the LiDAR sensor. For each point $p = (x, y, z, i) \in P$ in the point cloud, where (x, y, z) are the Cartesian coordinates and i is the intensity, we compute its projected coordinates (h, w) on the range image as follows:

$$\begin{pmatrix} h & w \end{pmatrix} = \begin{pmatrix} \left(1 - \frac{\arcsin(zr^{-1}) + |\text{fov}_{\text{down}}|}{\text{fov}_v}\right) H \\ \frac{1}{2}\left(1 - \frac{\arctan(y,x)}{\pi}\right) W \end{pmatrix} \tag{3.1}$$

In this equation, $r = \sqrt{x^2 + y^2 + z^2}$ is the range of the point, H and W are the height and width of the range image, fov_v is the vertical field of view, and fov_{down} is the

downward angle of the field of view. The two field of view parameters are constant, depending on the specific model of LiDAR sensors utilized.

The resulting range image is a multi-channel 2D representation where each pixel corresponds to a projected 3D point. In our implementation, we use five channels for each pixel: range (r), x-coordinate, y-coordinate, z-coordinate, and intensity (i). This rich representation allows our model to leverage both spatial and intensity information during the segmentation process.

It is important to note that this projection can result in information loss due to occlusions and varying point densities. Multiple 3D points may project to the same pixel, in which case we retain only the point with the smallest range. Conversely, some pixels may have no corresponding 3D points, resulting in "holes" in the range image. This challenge will be addressed later in our pipeline through a final 3D refinement step, which helps recover and refine the spatial information lost during projection.

Following the range projection, we implement a cropping strategy to further focus our model's attention and manage computational resources. This strategy extracts a focused portion of the range image, thereby enhancing the efficiency and effectiveness of the training process. Following RangeVIT, The cropping is performed horizontally, retaining 1/5th of the original image width while maintaining the full vertical resolution.

While our cropping method draws inspiration from the cropping technique used in RangeVIT, a significant difference exists. RangeVIT employs a random cropping strategy, selecting a crop randomly across the width of the range image for each scan, which effectively serves its multi-class segmentation objectives. RangeSAM, However, specifically targets the region containing the object instance of interest. This focused approach is essential for our class-agnostic task, as the majority of a typical outdoor LiDAR range image corresponds to background points.

To determine the precise location of the crop, we employ a bounding box detection algorithm on the selected instance. The algorithm identifies the minimum and maximum coordinates of the instance in both the vertical and horizontal directions. These coordinates are then used to center the crop window around the instance, ensuring that the object of interest is fully captured within the cropped region. If the instance is near the edge of the range image, the crop is adjusted to ensure it remains within the image boundaries, maintaining the crop size while shifting its position as needed.

Figure 3.4 provides an example of a range image with a highlighted instance and the corresponding crop boundaries. The instance is shown in green, while two vertical red lines indicate the boundaries of the crop width centered on the instance.

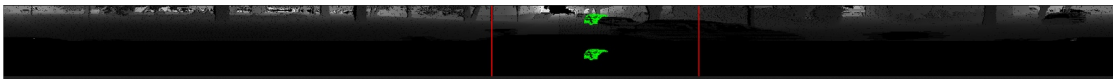

FIGURE 3.4: Range Image With Instance Highlight and Crop Boundaries

This cropping strategy offers several advantages. First, it allows us to focus computational resources on regions containing objects of interest. Second, it helps balance the dataset by ensuring that each training example contains a significant instance, rather than being dominated by background or empty space. Finally, by consistently centering the crop on the selected instance, we ensure that the model learns to segment objects in a consistent spatial context, which may improve its ability to identify and delineate object boundaries.

By combining instance selection with range projection and the aforementioned cropping strategy, we create a foundation for efficient and effective processing in the subsequent stages of our RangeSAM model. This approach allows us to leverage the strengths of 2D convolutional architectures while maintaining the crucial depth and spatial information inherent in the original 3D data, all while focusing on specific objects of interest within each scan.

3.2.2 Image Encoder: Stem and ViT Encoder

The image encoder in RangeSAM, consisting of a convolutional stem and a Vision Transformer (ViT) encoder, allows for effective processing of the range image representation of the point cloud. This component is adapted from RangeViT [49], with minor modifications designed around our promptable segmentation task.

The convolutional stem serves as a bridge between the range image input and the ViT encoder, addressing the domain gap between range images and the natural images typically processed by ViTs. As demonstrated in RangeVIT, the standard tokenization process of ViTs, which involves dividing the image into patches and linearly embedding them, is suboptimal for range image data. To overcome this limitation, A non-linear convolutional stem is employed, consisting of layers that have been shown to enhance optimization stability and predictive performance in ViT architectures [63].

This convolutional stem consists of two main components:

1. Context Module: This module, adapted from the SalsaNext architecture [64], comprises four residual blocks. It captures short-range dependencies of the 3D points projected in the range image, producing pixel-wise features with D_h channels at the same resolution as the input range image. The resulting tensor of context

features (t_c) has dimensions ($D_h \times H \times W$), where H and W are the height and width of the range image, respectively. It is important to note that these features are also passed to the Decoder (detailed later) via a skip connection, helping to preserve fine-grained spatial information throughout the network.

2. Projection Layer: To produce tokens compatible with the ViT input, an average pooling layer is applied which is followed by a 1x1 convolutional layer. This process reduces the spatial dimensions of t_c from $H \times W$ to $\frac{H}{P_H} \times \frac{W}{P_W}$, where P_H and P_W are respectively the patch height and patch width. The final 1x1 convolution produces D_i output channels.

The convolutional stem thus yields $M = \frac{HW}{P_H P_W}$ visual tokens $(v_1, \ldots, v_M)$, each of dimension D_i, matching the input requirements of a standard ViT.

To capture global contextual information, the ViT encoder processes an input sequence, t_0, which is prepared by stacking all visual tokens $(v_1, \ldots, v_M)$ with a special learnable classification token ($v_{\text{class}} \in \mathbb{R}^{D_i}$) which aggregates information from the entire sequence. Positional embeddings ($E_{\text{pos}} \in \mathbb{R}^{(M+1)\times D_i}$) are then added to the stacked tokens, providing spatial context to the encoder:

$$t_0 = [v_{class}, v_1, ..., v_M] + E_{pos} \tag{3.2}$$

This input sequence is then processed through L transformer blocks, resulting in an updated sequence of tokens ($t_L \in \mathbb{R}^{(M+1)\times D_i}$). After removing the classification token, we obtain the final deep patch representations ($t'_L \in \mathbb{R}^{M\times D_i}$).

Following RangeVIT, We use pre-trained VIT weights. Specifically, we initialize our ViT encoder with weights from a model pre-trained on supervised image segmentation on the 2D urban driving Cityscapes dataset [65]. This model, in turn, was initialized with weights from a model pre-trained on ImageNet-21k [66]. Despite the apparent domain shift between natural images and range images, this transfer learning approach has been shown to significantly improve performance and training convergence in LiDAR segmentation tasks [49].

The benefits of this pre-training strategy are twofold:

1. Feature Transferability: The pre-trained model has learned to extract general visual features that are, to some extent, transferable to range image data. This provides a strong starting point for our task-specific fine-tuning.

2. Optimization Stability: The pre-trained weights offer a favorable initialization point in the parameter space, potentially leading to faster convergence and better local optima during fine-tuning.

By leveraging this image encoder architecture, RangeSAM effectively processes the range image representation of the point cloud, capturing both local geometric details through the convolutional stem and global contextual information via the ViT encoder. This rich representation forms the foundation for the subsequent prompt-guided segmentation process, enabling our model to effectively isolate and segment objects of interest in sparse LiDAR point clouds.

3.2.3 Sampling and Encoding Prompts

Building upon the previous two components, RangeSAM incorporates a prompt encoder, enabling interactive segmentation. This component facilitates the transformation of user-provided or dynamically generated prompts into a format that effectively guides the segmentation process.

RangeSAM's Prompt Encoder is inspired by the Prompt Encoder in SAM, but is tailored specifically for point cloud data and the unique challenges of LiDAR-based segmentation. Unlike SAM, which supports multiple prompt types including points, bounding boxes and text, RangeSAM focuses exclusively on point prompts (hereafter referred to simply as "prompts"). This design choice is motivated by the natural compatibility of points with 3D data and the intuitive nature of point-based interaction in 3D space, building on the successes of previous work in 3D promptable segmentation [10, 13].

Our prompt encoder consists of two main components: a positional encoding layer and a set of learnable embeddings. The positional encoding layer employs a technique based on random Fourier features to encode the spatial information of each prompt [67]. This encoding is crucial for capturing the relative positions of prompts within the range image.

The positional encoding for a given prompt at location (x, y) is computed as follows:

$$\begin{aligned} PE(x,y) = [&\sin(f_1 x), \cos(f_1 x), \ldots, \sin(f_d x), \cos(f_d x), \\ &\sin(f_1 y), \cos(f_1 y), \ldots, \sin(f_d y), \cos(f_d y)] \end{aligned} \tag{3.3}$$

In this formulation, the frequencies f_i are generated by randomly sampling from a standard normal distribution, followed by scaling the sampled values by 2π. This random sampling strategy allows the positional encoding to capture a wide range of spatial relationships by incorporating a diverse set of frequencies. These frequencies are used to

generate the features that comprise the positional encoding of a prompt, allowing the model to learn to associate spatial relationships with semantic meaning effectively. The resulting positional encoding $PE(x, y)$ has a dimension of $D_{pe} = 2d$, where d is the number of frequency components used.

In addition to the positional encoding, we employ learnable embeddings to differentiate between foreground and background prompts. Foreground prompts specify the area of the object of interest, while background prompts are used to exclude irrelevant areas from the segmentation mask. Inspired by SAM, We define two learnable embedding vectors, E_{fg} for foreground prompts and E_{bg} for background prompts, each with a dimension D_p matching the positional encoding dimension D_{pe}. These embeddings are initialized randomly and are fine-tuned during training to learn optimal representations for each of the foreground and background prompt types.

The final embedding for a prompt is computed as:

$$E_p = PE(x, y) + E_{type} \tag{3.4}$$

where E_{type} is the learnable prompt embedding corresponding to either foreground or background.

Another key aspect of our prompt encoder is its ability to handle both 2D and 3D prompts seamlessly. While the prompt encoding process occurs in the 2D space of the range image, the model maintains compatibility with 3D interaction. This is achieved by projecting 3D prompts onto the range image, employing the projection method specified in equation 3.1 of the range projection section. This approach allows users to interact with the point cloud in 3D space while the model operates on the range image representation internally.

To facilitate training and evaluation without relying on user interactions, which is impractical, we employ a random sampling strategy, a commonly used technique in interactive segmentation within the 2D domain [13, 68]. In this strategy, within the range image, we randomly select between 1 and 12 foreground prompts from the target object and between 0 and 8 background prompts from an expanded bounding box around the target object. The expanded bounding box is created by enlarging the object's bounding box by 20% in each dimension, ensuring that background points are sampled from relevant areas close to the object.

In contrast to our method, SAM utilizes a more complex iterative refinement process for prompt generation during training. This approach simulates a series of user interactions, progressively refining the segmentation mask by adding new points based on the current

prediction's errors. While this method allows for more realistic simulation of user interactions and enhances prompt emphasis due to repeated prompt-feature interactions, it is very computationally intensive. Considering our limited computational resources (which will be detailed later), we chose a simpler, non-iterative prompt sampling approach in RangeSAM. Our method strikes a good balance between training efficiency and the ability to learn from diverse prompt configurations while ensuring the model is exposed to a wide range of prompts during training.

Figure 3.5 illustrates the result of our prompt sampling strategy on a range image. The red points represent foreground prompts sampled from the green target object, while the blue points indicate background prompts sampled from the surrounding area.

FIGURE 3.5: Random Prompt Sampling on Range Image

Figure 3.6 shows the same prompts re-projected into 3D space, demonstrating how 2D range image prompts correspond to locations in the original point cloud. This visualization highlights the model's ability to bridge 2D and 3D representations effectively.

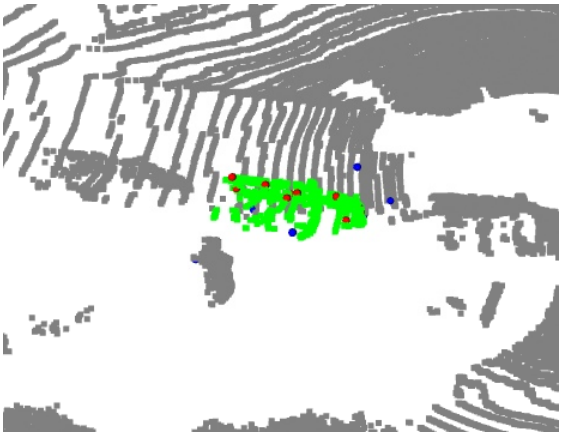

FIGURE 3.6: Reprojection of 2D Prompts onto the 3D Point Cloud

The prompt encoder's output is a set of embedded points that encapsulate both spatial and semantic information. These embeddings are then passed to the subsequent stages of the RangeSAM architecture, guiding the segmentation process by providing information about the object of interest and its surroundings.

3.2.4 Bi-Directional Transformer Integrated Decoder

The decoder component is the most crucial aspect of RangeSAM's architecture, as it enables transforming the encoded representations into a segmentation mask, while effectively integrating the information contained in the prompts. Our decoder incorporates elements from the decoders used in both SAM and RangeVIT, optimizing it for our 3D promptable segmentation task. The adapted components include an advanced upsampling mechanism, skip connections, and a bi-directional transformer for processing prompts.

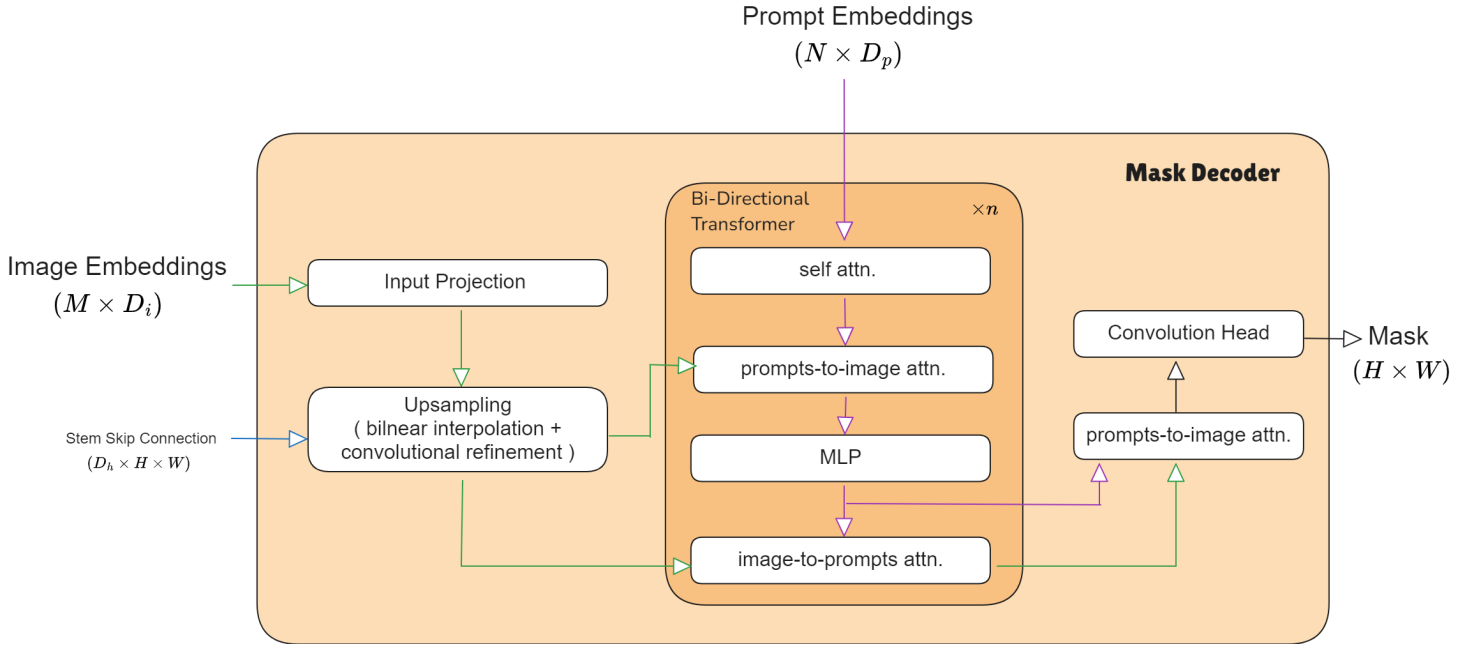

FIGURE 3.7: RangeSAM Decoder Architecture

As illustrated in Figure 3.7, the RangeSAM decoder integrates several key components to process the image embeddings and prompt embeddings effectively. The decoder takes as input the image embeddings $\mathbf{x} \in \mathbb{R}^{M \times D_i}$, where M is the number of visual tokens and D_i is the image embedding dimension, and the prompt embeddings $\mathbf{p} \in \mathbb{R}^{N \times D_p}$, where N is the number of prompts and D_p is the prompt embedding dimension. It produces an output mask $\mathbf{m} \in \mathbb{R}^{H \times W}$, where H and W are the height and width of the cropped range image, respectively. Following RangeViT's design, the working dimension of our decoder is set to D_h, matching the dimension of the context features from the convolutional stem.

The architecture can be broadly divided into three main stages: (1) the input projection and upsampling, (2) the bi-directional transformer for prompt-image interaction, and (3) the final mask prediction and refinement. Each of these components will be examined in detail.

Our decoder begins by projecting the encoded features from the ViT encoder to the decoder's working dimension D_h. This is accomplished through a learnable linear projection, ensuring that the encoder's output is compatible with the decoder's architecture while allowing for adaptive feature transformation.

Following the projection, the decoder employs an upsampling mechanism that combines bilinear interpolation with convolutional refinement. This approach, adapted from RangeViT, allows for efficient upscaling of the feature maps while preserving spatial information. The upsampling process can be expressed as:

$$x_u = f_u(x_p, t_c) \tag{3.5}$$

where f_u represents the upsampling function, $x_p \in \mathbb{R}^{D_h \times H_p \times W_p}$ is the projected input, t_c is the context features tensor passed through a skip connection, and $x_u \in \mathbb{R}^{D_h \times H \times W}$ is the resulting upsampled output. The function f_u internally uses bilinear interpolation followed by convolutional layers to refine the upsampled features.

As seen in RangeViT, The skip connection from the convolutional stem enhances spatial accuracy by directly supplying the pixel-wise context features t_c. These features are concatenated with the upsampled features along the channel dimension and refined through convolutional layers. This approach allows the decoder to utilize both high-level semantic information from the encoder and fine-grained spatial details from earlier layers, thereby improving segmentation quality.

While the upsampling process play crucial roles in refining spatial details, the most significant aspect of the RangeSAM decoder is the integration of a bi-directional transformer to process prompts alongside image features. This transformer, inspired by SAM but adapted for range image data, allows for effective interaction between the prompt embeddings and image features. More specifically, the transformer processes the prompt embeddings, image features, and their positional information, and produces refined outputs that have been enriched through multiple layers of interaction.

The bi-directional transformer has a depth of n layers, each containing self-attention mechanisms for the prompts, cross-attention between prompts and image features in both directions, and feed-forward networks. This bi-directional attention flow allows for rich interaction between the prompts and image features, enabling the model to effectively utilize the provided prompts for segmentation.

For each layer of the bi-directional transformer, we perform the following operations:

1. Self-attention on prompt embeddings:

$$p_i' = (p_i + \text{MHA}(p_i, p_i, p_i)) \tag{3.6}$$

2. Cross-attention from prompts to image features:

$$p_i'' = (p_i' + \text{MHA}(p_i', x_i, x_i)) \tag{3.7}$$

3. Cross-attention from image features to prompts:

$$x_i' = (x_i + \text{MHA}(x_i, p_i'', p_i'')) \tag{3.8}$$

4. Feed-forward networks for both prompt and image embeddings:

$$\begin{aligned} p_{i+1} &= (p_i'' + \text{FFN}(p_i'')) \\ x_{i+1} &= (x_i' + \text{FFN}(x_i')) \end{aligned} \tag{3.9}$$

In these operations, $x_i \in \mathbb{R}^{M \times D_h}$ represents the input image features to layer i, $p_i \in \mathbb{R}^{N \times D_p}$ are the prompt embeddings, MHA is multi-head attention, and FFN is a feed-forward network.

To further refine the decoder's output, we introduce a learnable scaling factor $\alpha \in \mathbb{R}$ for prompt embeddings. This scaling factor is applied to the image features $x_i \in \mathbb{R}^{M \times D_h}$ before their interaction with prompt embeddings $p_i \in \mathbb{R}^{N \times D_p}$ in the cross-attention mechanism, modifying the operation to $\text{MHA}(p_i', \alpha x_i, \alpha x_i)$. This allows the model to learn to balance the influence of the prompt information and the visual data, ensuring that neither source dominates and both contribute meaningfully to the final segmentation. This is particularly valuable in RangeSAM, as it compensates for the absence of an iterative refinement process found in SAM, helping to maintain sufficient emphasis on prompts throughout the decoding process.

After these bi-directional transformer operations, the updated image features and prompt embeddings are processed through an additional attention mechanism, following SAM's design. This enables a prompt-guided refinement of the image features, enhancing the segmentation based on the provided prompts. The attended features are then passed through a convolutional layer that projects them into the output space, generating the segmentation mask.

Unlike SAM, which produces multiple mask hypotheses and uses an IoU prediction head, RangeSAM opts for a simpler approach that generates a single mask output. This design

choice is motivated by the need for computational efficiency and the specific requirements of range image segmentation.

The final output of the decoder is a segmentation prediction at the same resolution as the cropped range image. However, To ensure that the prompts are accurately reflected in the final prediction, RangeSAM includes a logit modification step we title as "prompt label enforcement". This step adjusts the logits—the model's raw prediction values before conversion to probabilities—by modifying them at each spatial location (i, j) containing a prompt as follows:

$$M'(i,j) = \begin{cases} +\infty & \text{for the logit of the class corresponding to the prompt label} \\ -\infty & \text{for the logit of the class not corresponding to the prompt label} \\ M(i,j) & \text{if location } (i,j) \text{ contains no prompt} \end{cases} \tag{3.10}$$

This enforcement step ensures that the network's output matches the prompt labels, either foreground or background, at the specified locations. During training, this modification influences both the loss computation and backpropagation. The prompted locations will always produce minimal loss values, therefore the network learns to be more confident in its predictions around prompt locations while relying on these points as anchors for segmenting ambiguous regions. This effectively acts as a hard constraint that ensures the model's output respects the user-provided prompts.

Ultimately, the decoder of RangeSAM effectively processes range image data, producing a segmentation mask aligned with the LiDAR point cloud's range image representation. This completes the trainable portion of the network. While the output at this stage is capable of segmenting objects in the range image space, the segmentation mask is still in the 2D range image domain. This will be addressed through a final 3D refinement stage, which will create an entirely 3D mask.

3.2.5 KNN-Based 3D Refinement

The final component of RangeSAM is the KNN-based 3D refiner, designed to address potential information loss during the range projection process and refine the segmentation results in 3D space. This component builds upon the KNN post-processing technique introduced in RangeViT, adapting it for binary segmentation rather than multi-class segmentation.

The primary purpose of the KNN 3D Refiner is to overcome the limitations of range image representation, particularly the occurrence of "holes" or missing points in the projected 2D space. While the range image effectively preserves most of the 3D spatial information, some points may be lost due to occlusions or varying point densities. The KNN refiner helps recover these points and ensures a more accurate 3D segmentation.

Before the KNN refinement begins, RangeSAM projects the 2D segmentation results back onto the original 3D point cloud. This projection is achieved using stored indices that map each pixel in the range image to its corresponding 3D point. However few 3D points may not have a corresponding 2D pixel and thus lack a segmentation label. These unlabeled points become the focus of the KNN refinement. For these points, the KNN algorithm operates by examining the K nearest neighbors of each point in the 3D space.

Figure 3.8 provides a general illustration of the KNN algorithm, demonstrating how neighboring points are selected based on distance. The green point represents an unlabeled point, with $K = 3$ selecting the three nearest neighbors, classifying it as red, and $K = 5$ expanding the neighborhood to five points, classifying it as blue. This visual helps clarify how KNN aggregates information from nearby labeled points, a key aspect of the refinement process in RangeSAM.

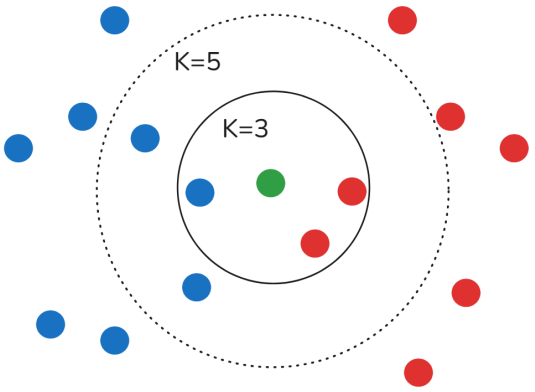

FIGURE 3.8: KNN Algorithm Illustration

The KNN refinement process of RangeSAM can be broken down into several steps:

1. Identification of Unlabeled Points: Points in the 3D point cloud that did not receive a label through the initial 2D-to-3D mapping are identified.

2. Neighborhood Sampling: For each unlabeled point in the 3D point cloud, the algorithm identifies its K nearest neighbors within the specified search radius. This step utilizes the original 3D coordinates of the points, allowing for accurate spatial relationships to be considered.

3. Distance Weighting: The algorithm applies a Gaussian weighting to the distances between the unlabeled point and its neighbors. This weighting ensures that closer points have a stronger influence on the final classification. The Gaussian kernel is defined as $W(d) = \exp\left(-\frac{d^2}{2\sigma^2}\right)$, with d being the distance between points and σ being the standard deviation.

4. Label Aggregation: The algorithm aggregates the binary labels (0 for background, 1 for foreground) from the labeled neighbors. These labels are weighted by their corresponding distance weights.

5. Final Classification: The final 3D label for each unlabeled point is determined by thresholding the weighted average of its neighbors' labels. If the average exceeds 0.5, the point is classified as foreground; otherwise, it is classified as background.

This process can be formulated as:

$$L_{3D}(p) = \begin{cases} 1 & \text{if } \frac{\sum_{i=1}^{K} W(d_i) \cdot L(n_i)}{\sum_{i=1}^{K} W(d_i)} > 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{3.11}$$

where $L_{3D}(p)$ is the final 3D label for the unlabeled point p, $L(n_i)$ is the label of the i-th neighbor (either from the initial projection or previously refined), and $W(d_i)$ is the Gaussian weight based on the distance to the i-th neighbor.

Overall, by considering neighboring points, the KNN 3D Refiner recovers information for occluded points and smooths out noise and inconsistencies in the segmentation. Additionally, it helps preserve fine geometric details that might have been lost in the 2D projection process.

By incorporating this final refinement step, RangeSAM effectively bridges the gap between 2D range image processing and 3D point cloud segmentation, aiming to produce high-quality, and spatially consistent results by leveraging the strengths of both representations.

3.3 Implementation Details

The implementation of RangeSAM involves careful consideration of various components, from data preprocessing to model architecture and training strategies. This section outlines the key aspects of our implementation, including the training setup, hyperparameter tuning process, and evaluation metrics.

3.3.1 Training Setup

Our training setup is designed to effectively leverage the architecture of RangeSAM for promptable, class-agnostic segmentation of sparse LiDAR point clouds. We build upon established practices from related works while introducing novel elements to address the unique challenges of our task. Here, we detail the key components of our training process, including model architecture specifics, data preparation, and optimization strategies.

Instance Selection and Range Projection. The input point clouds are projected onto a 2D range image with dimensions 2048×64. To focus on specific object instances, we crop this range image to a width of 384 pixels (1/5th of the original width), centered around the object of interest. This cropping strategy allows the model to concentrate on relevant areas while maintaining computational efficiency. The height of 64 pixels is preserved to retain vertical context, which is crucial for understanding the 3D structure of objects in the scene.

Image Encoder. Following RangeViT's approach, we use a ViT-S/16 model as our encoder, with 12 layers, 6 attention heads, and 384 channels (D_i=384), totaling 21 million parameters. The ViT encoder is initialized with ImageNet21k pre-trained weights and fine-tuned on Cityscapes for semantic segmentation. This transfer learning leverages rich feature representations from diverse outdoor driving images, making them suitable for our task. For the convolutional stem, we set the working dimension D_h to 128, with this value also being in common with the working dimension of the decoder.

Given the similarities between our sparse outdoor driving scenes and RangeViT's data domain, and both following the conceptual similar task of 'semantic' segmentation (even though the end result of our task is class-agnostic), we directly adopt these parameters. By maintaining these key architectural elements, we ensure a strong baseline for feature extraction, upon which we can build our promptable segmentation capabilities.

Prompt Encoder. The prompt encoder in RangeSAM plays a crucial role in integrating user-provided or dynamically generated prompts into the segmentation process. We set the initial prompt embedding dimension D_p to 128. This value was determined through our hyperparameter tuning process, allowing for a rich representation of prompt information while maintaining computational efficiency. The prompt embeddings are subsequently projected to match the decoder's working dimension through a linear projection layer in the decoder, enabling seamless integration with the image features.

Decoder. Our decoder builds upon the RangeViT architecture while incorporating key elements inspired by SAM. The upsampling process in the decoder uses a combination of 1×1 convolutions and Pixel Shuffle convolutions to efficiently recover fine-grained spatial information. This is followed by feature concatenation with the context features from the convolutional stem and further refinement through 3×3 and 1×1 convolutional layers, all maintaining the working dimension D_h=128.

A significant addition to our decoder is the bi-directional transformer, which enables effective interaction between prompt embeddings and image features. To manage computational complexity, we set this transformer to use an embedding dimension that matches our decoder's working dimension D_h. MLP blocks are set to a large internal dimension of 2048, but the limited number of prompts ensures that computational demands remain manageable. The transformer employs 8 attention heads and has a depth of 3 layers. All of these transformer-related parameters were subject to tuning in our experiments to optimize the prompt-image interaction.

3D Refinement. The 3D refinement stage employs a K-Nearest Neighbors algorithm to refine the segmentation results in 3D space. The tuning process, detailed later, revealed that using only the single nearest neighbor within a search radius of 3 units is most effective. This configuration effectively balances local detail preservation and computational efficiency.

Data Augmentation. We adopt a data augmentation strategy similar to RangeViT, applying transformations directly to the point clouds before range projection. This includes random flips over the y-axis (vertical axis in the range image), random translations, and random rotations between $\pm 5^\circ$ (applying roll, pitch, and yaw). Each augmentation is applied with a probability of 0.5. We do not apply additional augmentations to the range images themselves, as the inherent variability in LiDAR scans and our cropping strategy provide sufficient diversity.

Loss Functions. RangeSAM employs a combination of three loss functions to guide the training process:

1. Focal Loss: We use focal loss as our primary segmentation loss, defined as $L_{focal} = -\alpha(1-p)^\gamma y \log(p) - (1-\alpha)p^\gamma(1-y)\log(1-p)$, where p is the model's prediction, y is the ground truth, $\gamma = 2$ is the focusing parameter, and $\alpha = 0.95$ is the class balancing factor. This loss helps address class imbalance by down-weighting the loss contribution of well-classified examples, allowing the model to focus on harder,

misclassified examples. The high α value of 0.95 is chosen to account for the typical class distribution in our data, where foreground (object) points usually constitute around 5% of the total points.

2. Dice Loss: To complement the focal loss, we incorporate dice loss, defined as $L_{dice} = 1 - \frac{2\sum_i p_i y_i}{\sum_i p_i^2 + \sum_i y_i^2}$, where p_i and y_i are the predicted and ground truth values for pixel i. This loss is particularly effective for segmentation tasks as it directly optimizes the overlap between the predicted and ground truth masks, helping the model achieve a better balance between precision and recall.

3. Prompt Loss: We introduce a novel prompt loss to encourage the model to pay special attention to the areas around provided prompts. This loss computes the binary cross-entropy $L_{prompt} = -\frac{1}{N}\sum_{i\in\Omega}[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)]$ between the predicted and target segmentations within a 3×3 kernel Ω centered on each prompt point, where N is the number of valid points in the kernel. By emphasizing accurate predictions near prompt locations, this loss helps the model learn to more effectively utilize the prompt information.

The total loss is computed as a weighted sum of these components:

$$L_{total} = w_1 \cdot L_{focal} + w_2 \cdot L_{dice} + w_3 \cdot L_{prompt} \tag{3.12}$$

This weighting scheme uses values of $w_1 = 20$ for the focal loss, $w_2 = 1$ for the dice loss, and $w_3 = 0.3$ for the prompt loss. Following SAM, we maintain a 20:1 ratio between focal and dice loss to bring them to comparable scales, as focal loss typically produces smaller values. For the prompt loss, empirical testing indicated that a weight of 0.3 brought the term to a similar scale, albeit with less emphasis than the focal and dice losses, allowing them to serve as the primary contributors to overall segmentation.

Training Procedure. Our RangeSAM model is trained on the KITTI360 dataset, which we split into train/validation/test sets using an 80/10/10 ratio. This results in approximately 52,000 scans for training and 12,000 scans for evaluation. We train the model over 200 epochs using a single NVIDIA L40 GPU with 24GB of VRAM, which provides sufficient computational resources while maintaining a practical and accessible hardware setup. Given the complexity of our model and the challenging nature of the task, we opt for a batch size of 8 to allow for more frequent updates and potentially better generalization, all while remaining within the limits of our GPU memory.

We use the AdamW optimizer, an adaptive learning rate optimization algorithm that extends the Adam optimizer with decoupled weight decay. We set $\beta_1 = 0.9$ and $\beta_2 = 0.999$,

which control the exponential decay rates for the first and second moment estimates respectively, and a weight decay of 0.01 to regularize the model. Our learning rate schedule employs a linear warm-up phase followed by cosine annealing. The learning rate increases linearly from 0 to a peak value of 0.0004 over the first 10 epochs, then undergoes a cosine decay to 0 over the remaining epochs. This peak learning rate was determined through our tuning process to optimize the training dynamics for our specific task.

This training process is designed to effectively optimize the RangeSAM model over a large number of iterations. The combination of a carefully chosen optimizer, learning rate schedule, and batch size allows us to balance between computational efficiency and model performance. This approach enables handling the complexities of our task while maintaining a training setup that is both effective and feasible on commonly available hardware.

The specific values for several mentioned key parameters were determined through a systematic hyperparameter tuning process. The details of this process will be described later in this section.

3.3.2 Evaluation Metrics

To assess the performance of RangeSAM, we employ a diverse set of evaluation metrics. Given the novel nature of our work in promptable segmentation for sparse 3D point clouds, we utilize metrics that allow for comparison with various related approaches, including other promptable methods and non-promptable class-agnostic segmentation techniques.

Intersection over Union (IoU). The primary metric for evaluating segmentation quality is the Intersection over Union (IoU), also known as the Jaccard index. IoU measures the overlap between the predicted foreground segmentation mask and the ground truth foreground mask. It is calculated as:

$$\text{IoU} = \frac{|P \cap G|}{|P \cup G|} = \frac{TP}{TP + FP + FN} \tag{3.13}$$

Where P is the set of points in the predicted mask and G is the set of points in the ground truth mask. Additionally TP, FP, and FN are the true positive, false positive, and false negative foreground point assignments, respectively. IoU ranges from 0 to 1, with higher values indicating better segmentation accuracy. In practice, IoU scores are typically multiplied by 100 to express results as percentages.

To facilitate comparison with other promptable segmentation methods, we employ IoU@K, which measures the IoU achieved after K prompt inputs. This metric allows us to evaluate the model's performance as a function of the number of user interactions, providing insights into the efficiency and effectiveness of the prompting mechanism.

Precision, Recall, and F1 Score. These metrics provide a nuanced view of segmentation performance, particularly useful for imbalanced datasets common in point cloud segmentation tasks. Precision measures the proportion of correctly identified foreground points among all points predicted as foreground, highlighting the accuracy of the model's positive predictions. Recall measures the proportion of correctly identified foreground points among all actual foreground points, indicating the model's ability to capture relevant instances. The F1 score combines precision and recall into a single metric, providing a balanced measure that reflects both aspects of performance. These metrics are defined as:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN}, \quad \text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.14}$$

We also utilize Average Precision (AP), which summarizes the precision-recall curve as the weighted mean of precisions achieved at various confidence score thresholds. These thresholds represent the decision boundaries used to classify points as foreground or background, allowing for an evaluation of model performance at different levels of certainty. This metric is defined as:

$$\text{AP} = \sum_{n} (R_n - R_{n-1}) P_n \tag{3.15}$$

where P_n and R_n are the precision and recall at the nth threshold. By summarizing performance across these varying confidence levels, AP offers a single-number representation that captures the effectiveness of the segmentation model in real-world scenarios.

Segmentation Association Quality (S_{assoc}). To evaluate our model's performance in a class-agnostic setting, we adopt the Segmentation Association Quality (S_{assoc}) metric introduced in [6]. Unlike the standard IoU metric, which focuses solely on overlap, S_{assoc} accounts for how well points are assigned to a single ground truth instance. It weights the IoU with the number of true positives while normalizing by the instance size, effectively penalizing cases with fewer correctly associated points. This provides

a slightly more nuanced view of the instance segmentation quality. Its formulation is given as follows:

$$S_{assoc} = \frac{\text{TP} \cdot \text{IoU}}{|t|} \tag{3.16}$$

where $|t|$ represents the size of the ground truth instance. This formulation captures the quality of point-to-instance assignments, enhancing the evaluation of open-world instance segmentation tasks across varying instance sizes and shapes.

By employing this diverse set of metrics, we aim to provide a comprehensive evaluation of RangeSAM's performance, enabling comparisons with various segmentation approaches and offering insights into different aspects of the model's capabilities. The results and analysis based on these metrics will be presented in detail in the subsequent chapter.

3.3.3 Hyperparameter Tuning

The performance of RangeSAM is significantly influenced by the choice of hyperparameters. To optimize these critical parameters, we conducted a comprehensive hyperparameter tuning process. This process focused on key components of our architecture, including the learning rate, transformer depth, number of attention heads, MLP dimension in the transformer, and the prompt embedding dimension.

While our model involves numerous hyperparameters, we prioritized these specific ones due to their substantial impact on model performance and computational requirements. Other parameters, such as the image encoder architecture and certain decoder dimensions, were adopted from established baselines like RangeViT, allowing us to focus our limited computational resources on tuning the most critical aspects of our novel promptable segmentation approach.

Our tuning process employed a Bayesian optimization approach, which efficiently explores the hyperparameter space by building a probabilistic model of the objective function. This method allows for more intelligent sampling of hyperparameters compared to grid or random search, leading to faster convergence on optimal values. We conducted 100 trials, each evaluating the model's performance over 5 epochs without learning rate warm-up. To manage computational resources and time constraints, we used a subset of our KITTI360 dataset for this process, comprising 2000 scans for training and 1000 for validation, with the objective of maximizing average Intersection over Union (IoU) over the 5 epochs. It's important to note that the IoUs obtained during this tuning process

do not reflect the fully-trained model's performance in 3D promptable segmentation and are used solely for comparative purposes.

The hyperparameters we tuned and their respective search spaces were as follows: learning rate (1e-5 to 1e-2), transformer depth (1 to 4 layers), number of attention heads (2, 4, 8, or 16), MLP dimension in the transformer (512, 1024, 1536, or 2048), and prompt embedding dimension (64, 128, 256, or 384). These ranges were chosen based on common practices in transformer architectures and our hardware requirements.

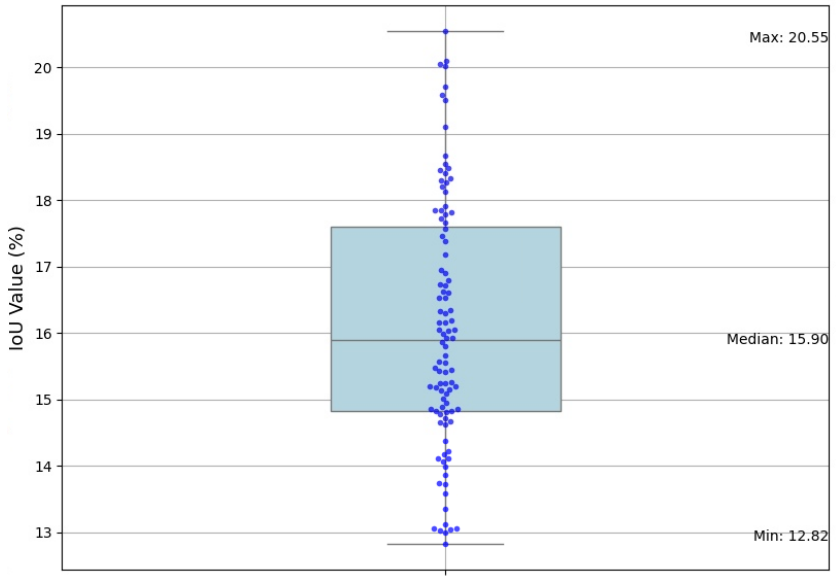

FIGURE 3.9: IoU Values Across Tuning Trials

Figure 3.9 presents the distribution of IoU values across our hyperparameter tuning trials. The box plot, complemented by a swarm overlay, reveals a performance range from 12.82 to 20.55, with a median of 15.90. This substantial spread of nearly 8% between the minimum and maximum IoU values underscores the critical impact of hyperparameter tuning on our model's performance. The concentration of points around the median and upper quartile indicates that many configurations achieved moderate success, while the outliers near 20.0 highlight the potential for significant improvement through optimal parameter selection.

Following this, in Figure 3.10 we illustrate the relationships between hyperparameters in the top 10 performing trials using a parallel coordinates plot. Each line represents a trial, with its color indicating the IoU value achieved. This visualization allows us to observe how different hyperparameter combinations contribute to model performance.

Notably, the best-performing configurations cluster around a learning rate of 0.0004 , with a consistent transformer depth of 3 layers. While both 8 and 4 attention heads

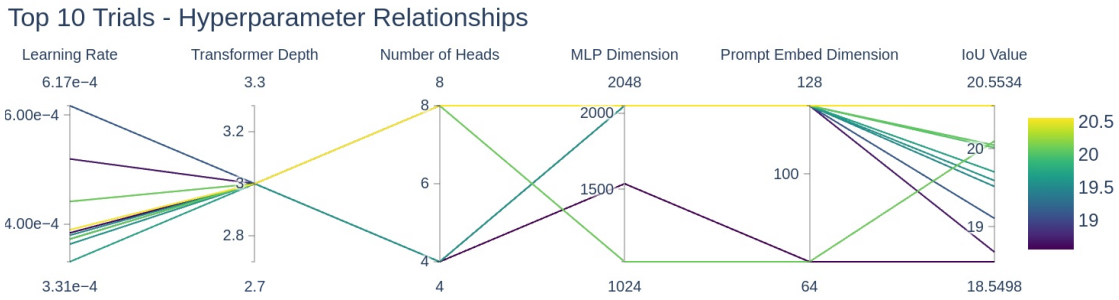

FIGURE 3.10: Hyperparameter Relationships of Top 10 Performing Trials

are present, the majority of top trials favor 8 heads. The MLP dimension is predominantly set at 2048, indicating a preference for higher model capacity. Interestingly, the prompt embedding dimension is consistently low (mostly 128) compared to the highest possible value (384), suggesting that smaller prompt representations are sufficient. The top-performing trial (IoU 20.55) uses 8 attention heads, the largest MLP dimension (2048), and a prompt embedding size of 128. Most other high-performing trials differ only slightly from this configuration. Based on these insights, we selected the hyperparameters from the top-performing trial for our final model, with the learning rate set to 0.0004, as it represents the optimal value in the continuous range observed across trials. As previously seen, the selection of these values is reflected in the training setup.

Following the optimization of our main model architecture, we conducted a separate hyperparameter tuning process for the KNN-based 3D refiner. This post-processing step is not part of the trainable network but plays a crucial role in refining the final 3D segmentation output. We evaluated various combinations of the number of neighbors (k), search radius, and kernel standard deviation (σ) on our fully trained model. Given the relatively few parameters, we performed a grid search over the parameter space: number of neighbors k (1, 3, 5, 7, 9), search radius (3, 5, 7), and σ (0.5, 1, 2). These combinations were assessed to identify the optimal configuration.

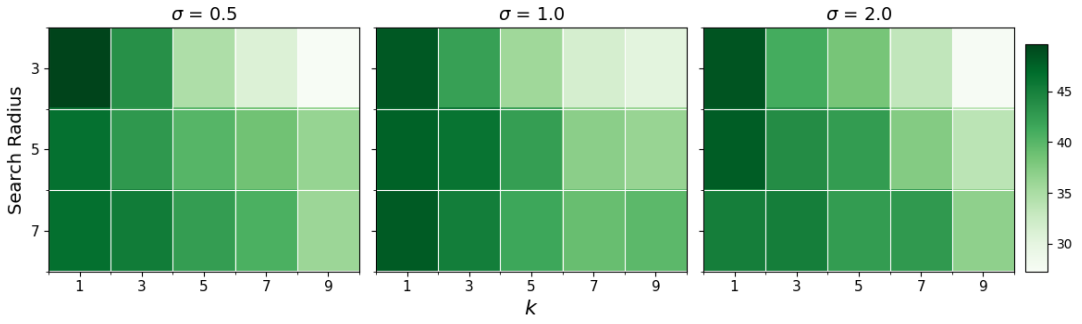

FIGURE 3.11: KNN Refiner Hyperparameter IoU Score Heatmap

Figure 3.11 presents a heatmap of the IoU scores achieved with different hyperparameter combinations for the KNN refiner at the points which it is applied to, with darker cells

representing higher IoU. Interestingly, the best results were obtained with k=1 and a search radius of 3, which effectively renders the kernel standard deviation parameter irrelevant. This configuration achieved a high IoU score, significantly outperforming other combinations.

A smaller neighborhood and moderate search radius indicate that relying on the nearest neighbor within a reasonable range offers the most precise refinement. This result can be explained by the prevalence of background points in LiDAR scans. Including more neighbors can inadvertently cause foreground points to be misclassified as background, particularly in areas with sparse object representation.

Ultimately, by carefully exploring the hyperparameter space for both our main architecture and the 3D refiner, we have ensured that our model is well-configured to tackle the challenges of promptable, class-agnostic segmentation in sparse LiDAR point clouds.

Chapter 4

Performance Evaluation

Building upon the methodological framework outlined in the previous chapter, we now present a comprehensive evaluation of RangeSAM's performance in promptable, class-agnostic segmentation of LiDAR point clouds. Since our approach uniquely addresses the challenge of sparsity which is overlooked in existing research, direct comparison with prior methods is challenging. To contextualize our results, we have devised appropriate baselines.

Our evaluation strategy is designed to provide a thorough understanding of RangeSAM's capabilities, strengths, and limitations. The analysis encompasses quantitative metrics for assessing segmentation accuracy and efficiency, qualitative examinations to visualize the model's output in various scenarios, and additional experiments which help understand the contribution of training datasets and key architectural components. This multi-faceted approach offers meaningful insights into RangeSAM's performance both in absolute terms and relative to the established baselines, highlighting its potential impact in the field of point cloud segmentation.

4.1 Evaluation Preliminaries

Prior to discussing our specific experiments, it is essential to establish some preliminaries for our evaluations. This section outlines key aspects of our evaluation methodology, ensuring clarity and consistency across subsequent analyses.

Prompt Sampling Strategy. Central to our evaluation process is the point sampling strategy used to simulate user prompts. While our training process used flexible

random sampling, our evaluation requires a more controlled approach to ensure fair comparisons. For quantitative evaluations, we employ a consistent point sampling strategy that mirrors our training methodology with a fixed number of prompts.

When sampling K points for evaluation, we target a ratio of approximately 65% foreground to 35% background points. This ratio, determined through empirical testing, provides a balance between object information and context. Points are sampled from within an expanded bounding box around the target object, enlarged by 20% in each dimension. For $K = 1$, we select a single foreground point; for $K > 1$, at least one background point is always included. If the number of available points is insufficient, we adjust the sampling while maintaining the total count of K points.

This strategy enables us to evaluate our model's performance consistently across experiments, providing a standardized framework for assessing RangeSAM's capabilities. In the following experiments, this strategy will be utilized as our prompt selection strategy unless explicitly stated otherwise.

Promptable Segmentation Comparisons. The novel nature of RangeSAM, particularly its focus on sparse LiDAR scans, presents unique challenges in comparative evaluation with other existing 3D promptable segmentation models. Ideally, we would evaluate these models on sparse data similar to ours. However, practical constraints such as unavailable model weights, proprietary code, and the substantial modifications required make such direct comparisons infeasible within our research timeline.

To bridge this gap and facilitate a more meaningful comparison, we've introduced a distinction between "sparse" and "dense" objects within our evaluation framework. This categorization is based on the observation that points closer to the LiDAR sensor tend to be denser, while those farther away are sparser. Specifically, we define objects as "dense" if they meet two criteria: (1) their center point is within 15 meters of the sensor, and (2) they contain at least 50 foreground points. Objects that don't meet both criteria are classified as "sparse." These thresholds were determined through visual examination of the point clouds, striking a balance that reasonably differentiates between noticeably denser and sparser object representations.

While this approach doesn't fully equate our task's difficulty with that of existing methods, it allows us to contextualize RangeSAM's performance more effectively. Importantly, this strategy ensures we don't overestimate our model's capabilities relative to others. Instead, it provides a conservative basis for comparison, acknowledging that even our "dense" objects may present more challenges than the data typically used in prior work, due to them in actuality still being sparse.

Evaluation Models. For the majority of our evaluations, including all experiments in the Quantitative and Qualitative Analysis sections, we employ the model outlined in Section 3.3.1, which is trained on the KITTI360 dataset over 200 epochs. This model serves as our primary benchmark for assessing RangeSAM's performance.

Additionally, we will explore specific scenarios requiring models trained on alternative datasets or under varied conditions, detailed in a dedicated section containing additional experiments. These experiments are designed to examine specific aspects of RangeSAM's architecture and cross-dataset performance. In each case, the exact model configurations will be specified, allowing us to maintain consistency in the main analysis while providing insights into the model's behavior across diverse training regimes.

Metric Prioritization. While our methodology introduced a comprehensive set of evaluation metrics, we prioritize the Intersection over Union (IoU) of the foreground mask as our primary performance indicator. IoU effectively captures many aspects of segmentation quality that other metrics individually represent. It inherently balances precision and recall, as it considers both false positives and false negatives in its denominator. Moreover, IoU's sensitivity to spatial overlap aligns well with the S_{assoc} metric's goal of assessing instance-level segmentation quality.

For our experiments which focus on RangeSAM's promptable segmentation capabilities, we will predominantly use the IoU variant we introduced as IoU@K, as is the standard practice across both 2D and 3D Promptable Segmentation techniques. However, to facilitate meaningful comparisons with non-promptable, class-agnostic segmentation models, and to gain further insights into the model's behavior, we will employ the full range of metrics introduced. This approach ensures a fair and comprehensive evaluation across different segmentation paradigms, allowing us to contextualize RangeSAM's performance within the broader landscape of 3D point cloud segmentation techniques.

Object Coverage. Unless stated otherwise, our evaluation experiments process all object instances present in each processed point cloud scan, in contrast to our training methodology where one random object is sampled per scan. This comprehensive evaluation protocol allows for thoroughly assessing model performance across a full range of object characteristics, providing robust insights into the model's segmentation capabilities under diverse operating conditions.

4.2 Quantitative Analysis

4.2.1 Prompt Influence

Setup. To evaluate the impact of prompts on RangeSAM's segmentation performance, we conducted an experiment consisting of two complementary components. The first component assessed the model's segmentation quality as a function of the number of prompts provided, while the second examined the model's sensitivity to prompt labels.

In the first component, we evaluated RangeSAM's performance using the IoU@K metric, where K represents the number of prompts provided. We varied K from 1 to 100, applying our consistent point sampling strategy to select prompts for object instances in a representative subset (approximately a quarter) of the KITTI360 test dataset. This sampling approach allowed us to efficiently explore a wide range of K values while maintaining statistical significance, providing insights into the model's prompt utilization efficiency and the potential trade-offs between user interaction and segmentation quality. Additionally, we conducted separate evaluations using only foreground prompts and only background prompts to understand their individual contributions to the segmentation process. This comprehensive analysis facilitates in revealing the model's ability to leverage different types of prompt information and the synergistic effects of combining foreground and background cues.

The second component aimed to verify the model's reliance on prompt labels by deliberately inverting the prompt assignments. We selected $K = 20$ as it represents a practical upper limit for user input in real-world scenarios, balancing between segmentation accuracy and user effort. For each object instance, we assigned background labels to prompts within the object and foreground labels to prompts in the background, effectively creating an "inside-out" prompting scenario. This test serves to confirm whether RangeSAM genuinely learns to leverage the semantic information provided by the prompts or if it relies more heavily on the spatial distribution of the points of the point cloud.

These components collectively provide a detailed view of RangeSAM's prompt-driven segmentation capabilities, assessing both its performance scaling with prompt quantity and its ability to correctly interpret prompt semantics.

Results and Discussion. The results of the first component, illustrating RangeSAM's segmentation performance as a function of the number of prompts, are presented in Figure 4.1.

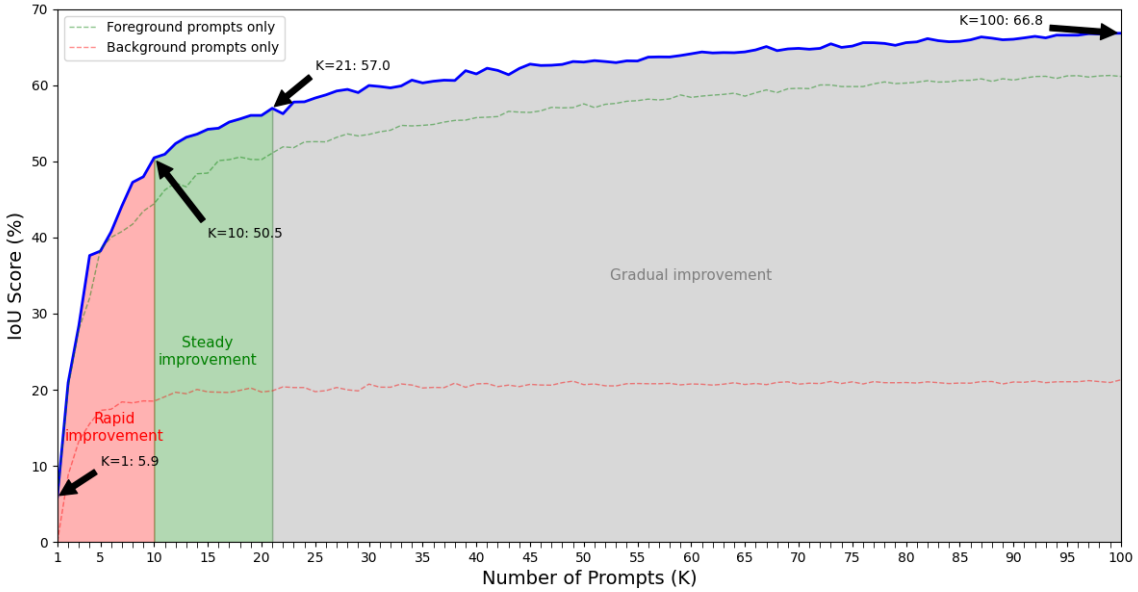

FIGURE 4.1: IoU Performance Across Varying Prompt Counts on KITTI360

This figure reveals several key insights into RangeSAM's behavior. As represented, the performance curve can be divided into three distinct regions, each characterizing a different phase of the model's response to increasing prompt inputs.

In the initial phase ($K = 1$ to $K \approx 10$), we observe rapid improvement, with IoU rising sharply from 5.9 at $K = 1$ to about 50.5 at $K = 10$, demonstrating the model's ability to quickly leverage even a small set of prompts. The second phase ($K \approx 10$ to $K \approx 21$) shows steady improvement, with IoU reaching approximately 57.0 at $K = 21$, aligning with our training process which used a maximum of 20 prompts. Beyond $K = 21$, performance gradually improves, despite minor fluctuations potentially caused by the model encountering prompt configurations outside its training range. Overall, the trend remains positive, with the IoU reaching approximately 66.9 at $K = 100$.

Notably, the figure also illustrates the model's performance when provided with only foreground or only background prompts. For the first few prompts ($K \leq 5$), the foreground-only and all-prompts curves are nearly identical, while the background-only curve shows a rapid initial increase from near-zero to about 15.0 IoU. Beyond this point, the curves diverge into a more consistent pattern. The background-only curve, while consistently lower than the all-prompts curve, shows non-trivial segmentation capability, maintaining IoU values close to 20.0. This suggests that background prompts alone provide some context for object delineation, possibly by helping the model identify object boundaries through contrast. The foreground-only curve shows a pattern similar to the all-prompts curve, albeit with consistently lower IoU values. This indicates that while foreground prompts are crucial for object delineation, the inclusion of background prompts provides valuable additional context for refinement, enabling more accurate segmentation. The

relatively constant gap between the all-prompts and foreground-only curves beyond the initial prompts suggests that the contribution of background prompts, while significant, remains fairly consistent as the number of prompts increases.

RangeSAM's performance with a single foreground prompt, while limited, shows some capability in object segmentation. The steep initial improvement with just a few additional prompts demonstrates the model's efficient use of minimal user input, potentially streamlining real-world manual segmentation tasks. This rapid improvement suggests RangeSAM quickly builds a robust understanding of object boundaries and spatial relationships within point clouds, even with sparse input.

The continued, gradual improvement beyond the training range showcases RangeSAM's ability to generalize and handle more complex segmentation scenarios. This indicates the model has learned flexible, generalizable features and strategies rather than simply memorizing specific prompt configurations. However, diminishing returns at higher prompt counts point to limitations in the current architecture and the nature of the task itself. As more prompts are added, they likely provide increasingly redundant information, suggesting a practical limit to the useful data extractable from prompts alone.

Additionally, the inherent sparsity of LiDAR point clouds may impose a ceiling on achievable accuracy, regardless of the number of prompts provided. This observation highlights the challenging nature of segmentation in sparse 3D data and suggests that future improvements might require novel approaches to information integration or alternative input modalities to complement prompts.

Proceeding to the next analysis, Table 4.1 presents the results of the second component, which compares normal and inverted prompt labeling at $K = 20$.

Prompt Labeling	**IoU**
Normal	56.05
Inverted	17.53

TABLE 4.1: Effect of Prompt Label Inversion

The comparison between normal and inverted prompt labeling reveals interesting insights into RangeSAM's behavior. While the performance drop with inverted prompts is substantial, the resulting IoU of 17.53 is not negligible, suggesting that the model still extracts useful information even from incorrectly labeled prompts. This resilience is noteworthy, as it indicates that RangeSAM's segmentation capability surpasses merely relying on prompt labels in their entirety.

The persisting performance with inverted prompts can be attributed to several factors. Firstly, even with incorrect labels, the prompts still provide valuable spatial information about the object's location and extent within the point cloud. This suggests that RangeSAM leverages both the semantic labels and the spatial distribution of prompts in its segmentation process. Secondly, the model may be identifying patterns or features in the point cloud that correlate with object boundaries, allowing it to partially overcome the label inversion.

However, the significant performance gap between normal and inverted prompts (56.05 vs 17.53) underscores the importance of correct labeling. It demonstrates that RangeSAM does indeed rely heavily on the semantic information provided by the prompts, using it to guide its attention and refine its segmentation decisions. The fact that performance with inverted prompts falls below that achieved with just two correctly labeled prompts (IoU@2 = 20.94) further emphasizes the value of accurate prompt information.

Overall, the two components of this experiment yield several key insights into RangeSAM's capabilities and limitations. It demonstrates the model's ability to effectively utilize prompts for guided segmentation, with performance improving as more prompts are provided. The model shows robustness in handling prompts beyond its training range and exhibits a degree of resilience to mislabeled inputs. However, it also highlights areas for potential improvement, such as enhancing single-prompt performance, improving performance with a higher number of prompts, and developing strategies to better manage noisy or incorrect user inputs.

4.2.2 Cross-dataset Generalization

Setup. This evaluation assesses RangeSAM's ability to generalize across different datasets and urban environments. We evaluate the model's performance on three distinct datasets: the KITTI360 test set (Karlsruhe, Germany, semi-urban environments), a subset of SemanticKITTI (Karlsruhe, suburban environments), and the entire ApolloScape dataset (various cities in China). We use IoU@K as our primary metric, evaluating performance at various K values. The KITTI360 test set serves as our baseline, while we select a random subset of SemanticKITTI with an equivalent number of scans to ensure a balanced comparison. Due to its smaller size, we use the entire ApolloScape dataset.

This analysis aims to provide insights into RangeSAM's adaptability to different geographical contexts and urban layouts, demonstrating the model's potential for real-world applications across diverse global settings.

Results and Discussion. The results of our cross-dataset evaluation reveal intriguing insights into RangeSAM's generalization capabilities across different urban environments and geographical contexts. Figure 4.2 presents a comparison of IoU@K values for KITTI360, SemanticKITTI, and ApolloScape datasets.

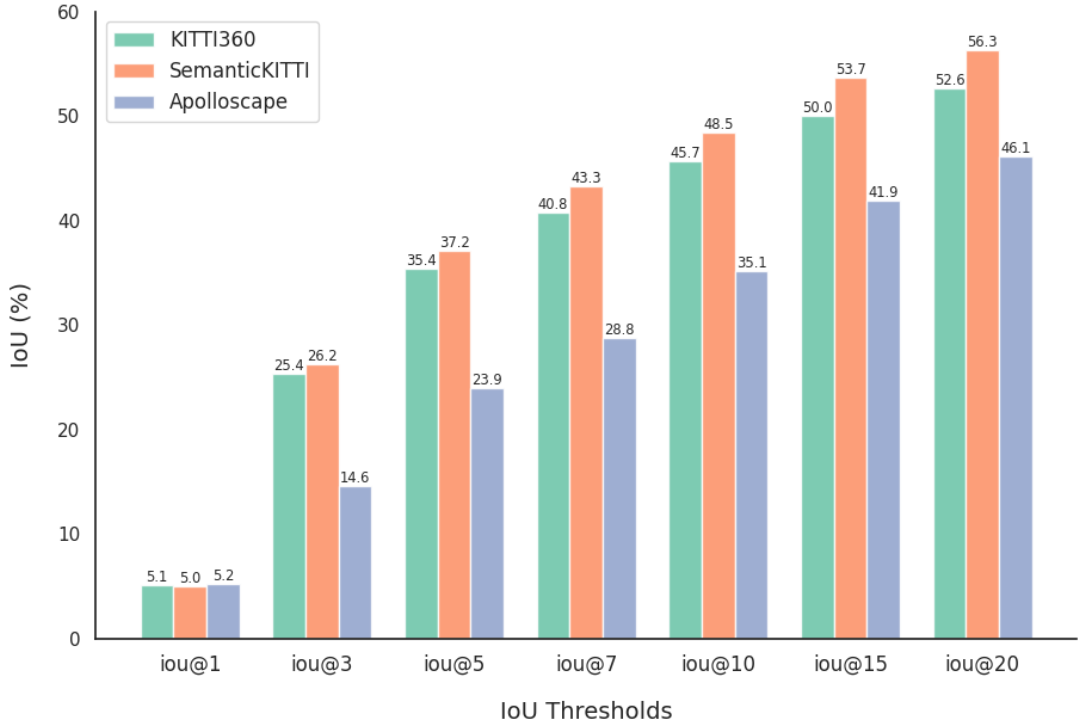

FIGURE 4.2: Cross-Dataset IoU@K Comparison

As evident, RangeSAM's performance at $K = 1$ is consistently low across all datasets, with IoU values of approximately 5. This suggests that a single prompt, regardless of the dataset, provides insufficient information for accurate segmentation. However, as the number of prompts increases, we observe distinct performance patterns across the datasets.

Interestingly, RangeSAM exhibits comparable or even slightly superior performance on the SemanticKITTI dataset compared to KITTI360, despite being trained on the latter. At $K = 20$, the model achieves an IoU of 56.3 on SemanticKITTI, noticeably outperforming its 52.6 IoU score on KITTI360. This result can be attributed to several factors. Firstly, SemanticKITTI encompasses a more focused set of object types in suburban environments, potentially simplifying the segmentation task. In contrast, KITTI360's semi-urban scenes present a wider variety of object types and more complex spatial relationships, challenging the model with a broader range of segmentation scenarios. Additionally, the suburban nature of SemanticKITTI scenes may feature less cluttered environments, facilitating easier object delineation.

The model's performance on ApolloScape, while lower than on the other datasets, remains reasonably strong, achieving an IoU of 0.461 at $K = 20$. This drop in performance is expected given the significant differences in urban layouts and object appearances between Chinese cities and the German environments used for training. Factors such as distinct architectural styles, varying vehicle types, and different road infrastructure likely contribute to this performance gap. Nevertheless, RangeSAM's ability to maintain substantial segmentation accuracy on ApolloScape demonstrates a reasonable degree of robustness and potential for cross-cultural applicability.

It is worth noting that the performance gap between datasets narrows as K increases, highlighting RangeSAM's adaptability across different environments. For example, at $K = 3$, there's a noticeable difference in IoU, with SemanticKITTI at 0.262 and ApolloScape at 0.146. By $K = 20$, however, this gap has proportionally decreased, with SemanticKITTI at 0.563 and ApolloScape at 0.461. This convergence suggests that the more prompts given to RangeSAM, it becomes better equipped to handle unfamiliar environments. Furthermore, the consistent improvement in IoU across all datasets underscores the model's ability to effectively integrate prompt information with point cloud features, reflecting learned generalizable strategies rather than overfitting to the training data.

Overall, these findings paint a nuanced picture of RangeSAM's generalization capabilities. The model demonstrates robust performance across datasets from similar geographical contexts (KITTI360 and SemanticKITTI) and maintains reasonable accuracy in substantially different urban environments (ApolloScape). While there remains a performance gap between cross-dataset and within-dataset results, this disparity highlights the challenge of generalization across significantly different urban layouts and object appearances. Nevertheless, RangeSAM's ability to maintain strong performance across these diverse datasets showcases its potential for adaptability in real-world applications across varied global settings.

4.2.3 Comparative Prompt-Driven Segmentation

Setup. To contextualize RangeSAM's performance within the landscape of 3D promptable segmentation models, we conduct a comparative evaluation against three state-of-the-art approaches: PointSAM, InterObject3D, and Agile3D. This experiment aims to assess RangeSAM's efficacy in handling LiDAR data segmentation relative to these established models, which primarily operate on dense point clouds.

Our evaluation framework mirrors the methodology of referenced studies, assessing performance at specific prompt counts on the KITTI360 dataset, but a crucial distinction

lies in the nature of the data used. While comparison models are evaluated on dense point cloud sequences from KITTI360, RangeSAM operates on the original sparse LiDAR scans from the KITTI360 test set, ensuring evaluation on unseen data and highlighting its capability to handle sparser, more challenging data typical in real-world autonomous driving scenarios. To address inherent data density differences and ensure a balanced comparison, we focus on objects classified as "dense" within the sparse LiDAR scans, using our consistent point sampling strategy to generate prompts and maintain the established foreground-to-background ratio across all prompt counts.

This evaluation strategy, while not perfectly equating the challenges faced by RangeSAM with those of the comparison models, provides a conservative basis for assessing our model's performance. It allows us to examine how segmentation quality evolves with increasing user input across different approaches, while acknowledging the additional complexity RangeSAM handles in working with sparser data.

Results and Discussion. The comparative evaluation of RangeSAM against state-of-the-art 3D promptable segmentation models reveals interesting insights into its performance and potential. Figure 4.3 illustrates the IoU values achieved by RangeSAM, PointSAM, InterObject3D, and Agile3D across different prompt counts on the KITTI360 dataset.

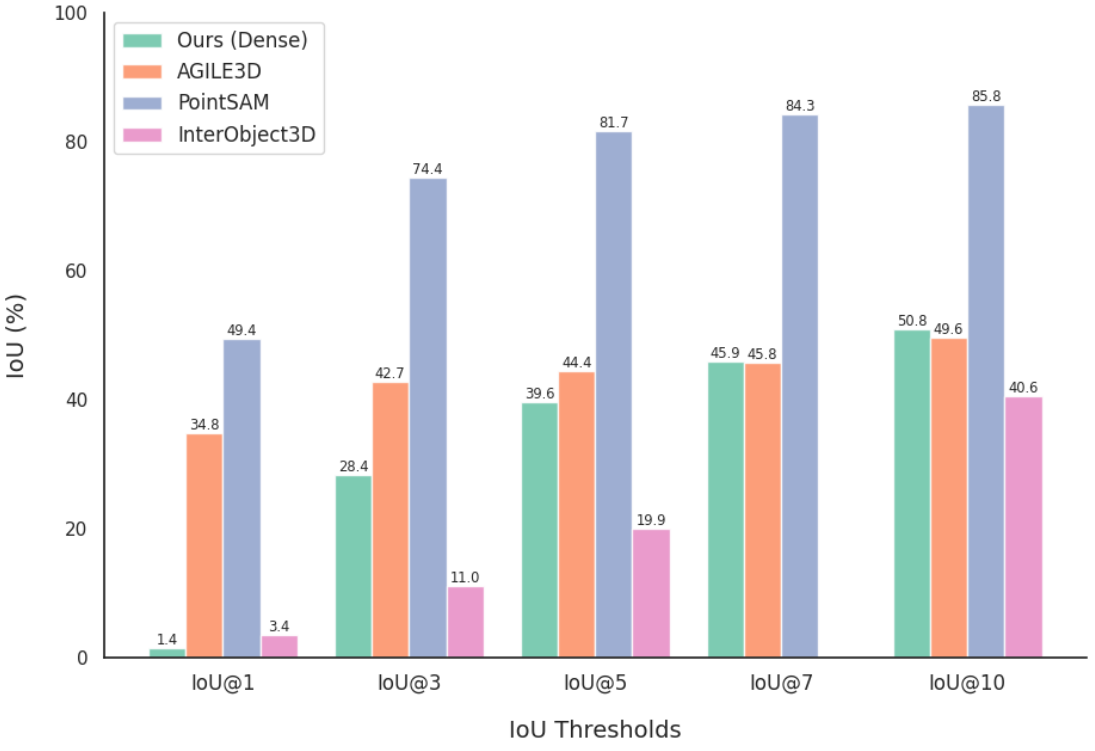

FIGURE 4.3: 3D Promptable Models IoU@K Comparison

RangeSAM demonstrates competitive performance, particularly when compared to InterObject3D and Agile3D, though it struggles with single-point prompts as observed in previous experiments. At $K = 1$, RangeSAM's performance (IoU = 1.36) is lower than all comparison models, reflecting the challenges of inferring object boundaries from a single point in sparse LiDAR data. However, RangeSAM shows rapid improvement as the number of prompts increases. By $K = 3$, it achieves an IoU of 28.36, surpassing InterObject3D (11.0) and closing the gap with Agile3D (42.7). This trend continues, with RangeSAM outperforming InterObject3D across all measured prompt counts and approaching Agile3D's performance at higher K values. By $K = 10$, RangeSAM (50.82) slightly edges out Agile3D (49.6). PointSAM, however, consistently outperforms all other models, achieving significantly higher IoU values across all prompt counts (85.8 at $K = 10$). This superior performance suggests PointSAM benefits from more sophisticated architectures and training strategies, potentially leveraging dense point cloud data more effectively. Nevertheless, RangeSAM's performance trajectory demonstrates its effectiveness in leveraging multiple prompts to refine segmentation results in sparse point clouds.

The performance trajectory of RangeSAM is particularly noteworthy. While it starts with the lowest IoU at $K = 1$, it shows the steepest improvement curve among all models as K increases. This rapid improvement indicates that RangeSAM efficiently utilizes additional prompts to refine its segmentation, a valuable characteristic for interactive segmentation tasks where users can iteratively provide more input.

It is crucial to interpret these results within the context of our evaluation framework. Our focus on "dense" objects in the KITTI360 dataset, while aiming for a fair comparison, still presents RangeSAM with more challenging data than the comparison models typically handle. These objects in our LiDAR scans are sparser than the point clouds used to train and evaluate the other models, meaning that RangeSAM is handling a more difficult task, making its competitive performance even more noteworthy. The comparison models, originally designed for truly dense point clouds, may struggle with the relative sparsity of our data. RangeSAM's ability to maintain competitive performance in this challenging scenario underscores its potential as a robust segmentation tool capable of handling varying degrees of point cloud sparsity.

4.2.4 Class-Agnostic Performance Comparison

Setup. This experiment aims to compare RangeSAM's performance against conventional high-performing class-agnostic instance segmentation models in outdoor driving scenes. Despite the inherent differences in approach—RangeSAM being prompt-driven

while others segment entire LiDAR scans without explicit user input—we conduct this comparison to assess our model's effectiveness in scenarios with limited user interaction. For this evaluation, we fix the number of prompts (K) to 20 per object instance, which we argue represents a reasonable balance between user effort and segmentation quality.

We evaluate RangeSAM against two prominent models we previously introduced in our literature review: SAM3D [54], which adapts SAM directly for 3D segmentation, and 3DUIS [6], the current state-of-the-art in outdoor driving scene instance segmentation. All models are evaluated on the SemanticKITTI dataset, which is the standard benchmark for outdoor driving scenes in non-promptable class-agnostic scenarios. We employ a comprehensive set of evaluation metrics to provide a holistic view of performance: Precision, Recall, F1 score, Average Precision at 25% and 50% overlap thresholds (AP@25, AP@50), global Average Precision (AP), and Segmentation Association Quality (S_{assoc}). These metrics offer insights into various aspects of segmentation quality, from point-level accuracy to instance-level coherence.

Results and Discussion. Table 4.2 presents the performance comparison between RangeSAM, 3DUIS, and SAM3D across all evaluation metrics on the SemanticKITTI dataset.

Model	Precision	Recall	F1	AP@25	AP@50	AP	S_{assoc}
RangeSAM	79.7	**70.5**	**74.4**	57.2	53.6	**73.3**	50.4
SAM3D	24.7	21.1	22.1	27.5	8.1	1.3	14.3
3DUIS	79.9	67.0	72.2	63.5	56.1	46.6	62.9

TABLE 4.2: Class-Agnostic Model Performance on SemanticKITTI

The results reveal intriguing insights into RangeSAM's performance relative to these established models. First and foremost, RangeSAM significantly outperforms SAM3D across all metrics, demonstrating the superiority of our native 3D approach over methods that directly adapt 2D models to 3D data. This substantial performance gap underscores the importance of architectures specifically designed for 3D point cloud data, particularly in handling the complexities of sparse LiDAR scans.

Comparing RangeSAM to 3DUIS yields a more nuanced picture. RangeSAM achieves comparable or superior performance in most key metrics. Our model's precision (79.7) is nearly identical to 3DUIS (79.9), indicating a similar ability to accurately identify foreground points. Notably, RangeSAM achieves higher recall (70.5 vs. 67.0) and F1 score (74.4 vs. 72.2), suggesting it is effective at identifying a larger proportion of relevant points while maintaining better overall segmentation quality. The substantially higher global Average Precision (73.3 vs. 46.6) is particularly noteworthy. This large gap in AP likely stems from different functionality: while 3DUIS discovers and segment

instances simultaneously, RangeSAM benefits from prompts that effectively narrow down the region of interest. This focused attention allows our model to make more confident and precise predictions within these regions, leading to higher quality confidence scores that translate into better AP performance.

However, 3DUIS maintains an edge in certain areas. It achieves higher AP@25 and AP@50 scores, indicating better performance at stricter overlap thresholds. The most significant difference lies in the Sassoc metric, where 3DUIS (62.9) significantly outperforms RangeSAM (50.4).

This disparity in S_{assoc} is particularly intriguing. While RangeSAM demonstrates strong point-level accuracy, its lower S_{assoc} score suggests challenges in instance-level grouping. This indicates that while our model excels at identifying which points belong to the foreground, it may struggle with accurately grouping these points into coherent object instances. A possible explanation is that RangeSAM might be prone to oversegmentation, splitting single object predictions across multiple instances, or failing to properly separate closely spaced objects.

The prompt-driven nature of RangeSAM may explain both its strengths and limitations. While user prompts enable focused, confident predictions leading to high global AP, this same mechanism might affect instance grouping, particularly in complex scenes with multiple nearby objects. Despite these challenges, RangeSAM's performance remains impressive, especially considering its ability to operate on sparse LiDAR data and provide user-guided segmentation, demonstrating strong potential for practical applications where accurate point-level identification and localization are crucial.

To gain deeper insights into RangeSAM's performance characteristics, particularly its instance grouping behavior, a qualitative analysis of segmentation results would be invaluable. Visual inspection of the model's outputs could reveal how the distribution of prompts affects segmentation coherence and how the model handles object boundaries in various scenarios. This analysis, which we will conduct in the following section, may provide clearer explanations for the observed quantitative results, and guide potential refinements to our prompt sampling strategy or model architecture.

4.3 Qualitative Analysis

To complement our quantitative evaluation, we conduct a thorough qualitative analysis of RangeSAM's performance. This visual inspection provides valuable insights into the model's behavior across various scenarios, helping to elucidate the strengths and limitations observed in our quantitative metrics. We examine the impact of prompt

count on segmentation quality, assess the model's performance on different datasets, and investigate potential areas for improvement in our prompt sampling strategy.

We begin by examining the impact of increasing prompt counts on segmentation quality for a typical object in the SemanticKITTI dataset. Figure 4.4 illustrates this progression, showcasing segmentation results for $K = 3$, $K = 5$, and $K = 10$ prompts, with red spheres representing foreground prompts and blue spheres representing background prompts.

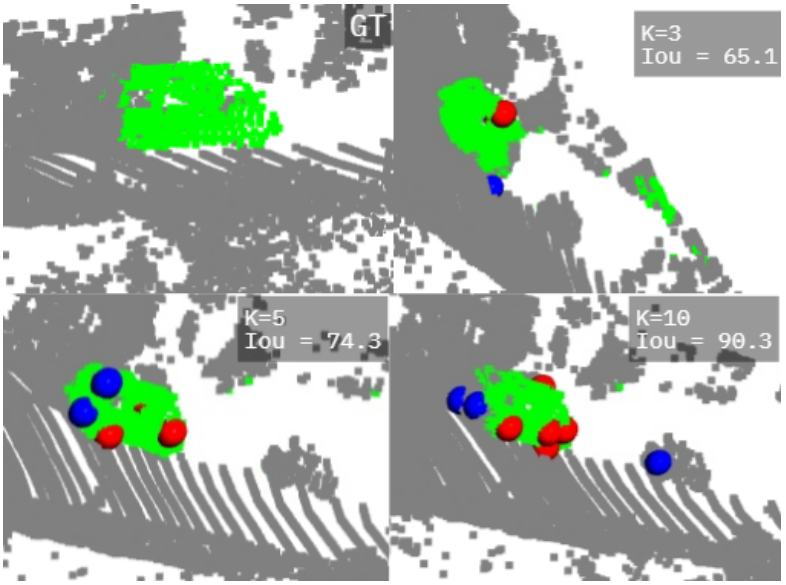

FIGURE 4.4: Object Instance Segmentation Example in SemanticKITTI

As evident in this figure, the segmentation quality improves markedly as the number of prompts increases. At $K = 3$ (IoU 65.1), a tendency for over-segmentation can be observed, with the predicted mask extending beyond the ground truth boundaries. This suggests that with limited prompts, RangeSAM exhibits an over-inclusive bias, potentially capturing contextual information or nearby objects. As the prompt count increases to $K = 5$ (IoU 74.3), the segmentation becomes more refined, with fewer extraneous points included. However, some points in close proximity to the object are still incorrectly classified as foreground. At $K = 10$ (IoU 90.3), the model produces a highly accurate segmentation that closely aligns with the ground truth, with minimal false positives or negatives.

This progression aligns with our quantitative findings, demonstrating RangeSAM's ability to effectively utilize additional prompt information to refine its segmentations. The significant improvement from $K = 3$ to $K = 10$ in the example underscores the value of relevant prompts in achieving high-quality segmentations, particularly in challenging or ambiguous scenarios. It also highlights the model's capacity to integrate multiple spatial cues to form a better understanding of object boundaries.

Moving beyond individual objects, Figure 4.5 presents a comparison between ground truth and RangeSAM's predictions for an entire KITTI360 scan. Each object instance is segmented using 20 prompts ($K = 20$) and assigned a unique color. The overall IoU averaged across all instances in this scan is 59.8, which is impressive considering the scene's complexity and the challenges posed by varying object sizes and point densities.

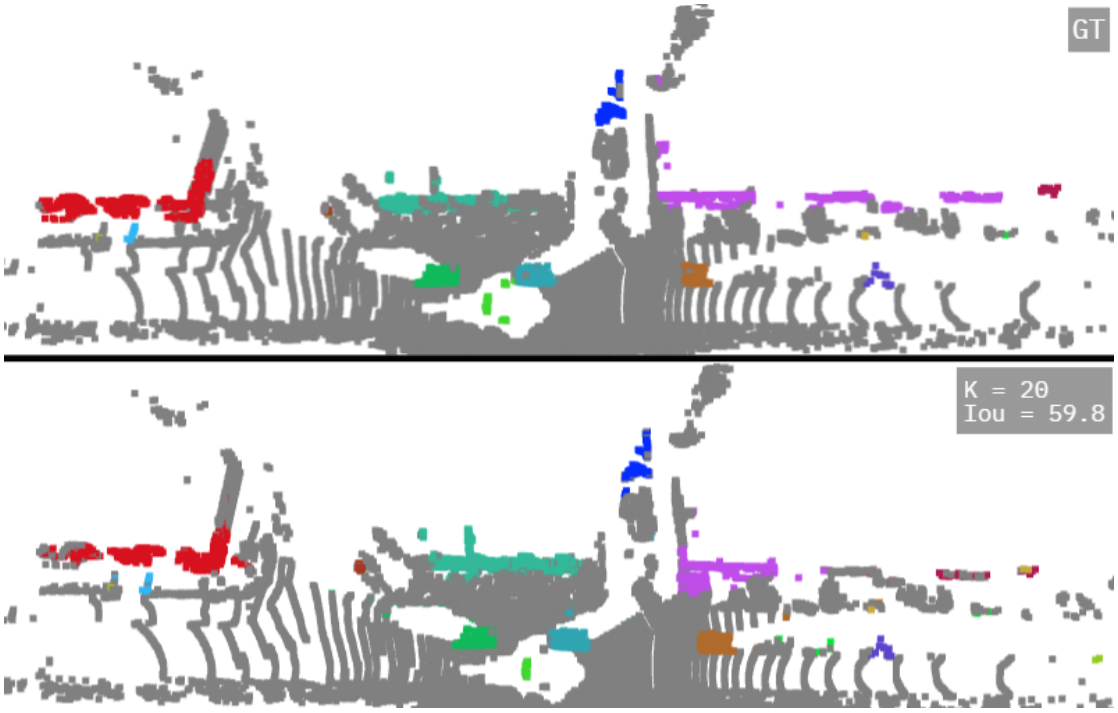

FIGURE 4.5: Prediction Visualization of a Complete KITTI360 Scan

In this figure, the predicted segmentation generally mirrors the ground truth. However, upon closer inspection, we notice that some objects in the prediction are missing points that are present in the ground truth, while others include extra points, with them being scattered onto other objects. This discrepancy could be attributed to several factors. For distant or partially occluded objects, the model may tend to struggle with accurately defining boundaries due to sparser point representation. Conversely, for larger or closer objects, the model can include nearby points not present in the ground truth, likely due to it aiming to mimic the contextual understanding gathered from training.

Nonetheless, the overall coherence of the predicted segmentation demonstrates RangeSAM's ability to maintain consistent performance across diverse object types and spatial arrangements within a scene. This is particularly noteworthy given the varying point densities and occlusions present in LiDAR scans.

To assess RangeSAM's generalization capabilities, we also examined its performance on the ApolloScape dataset, testing the model on a distinct urban environment with

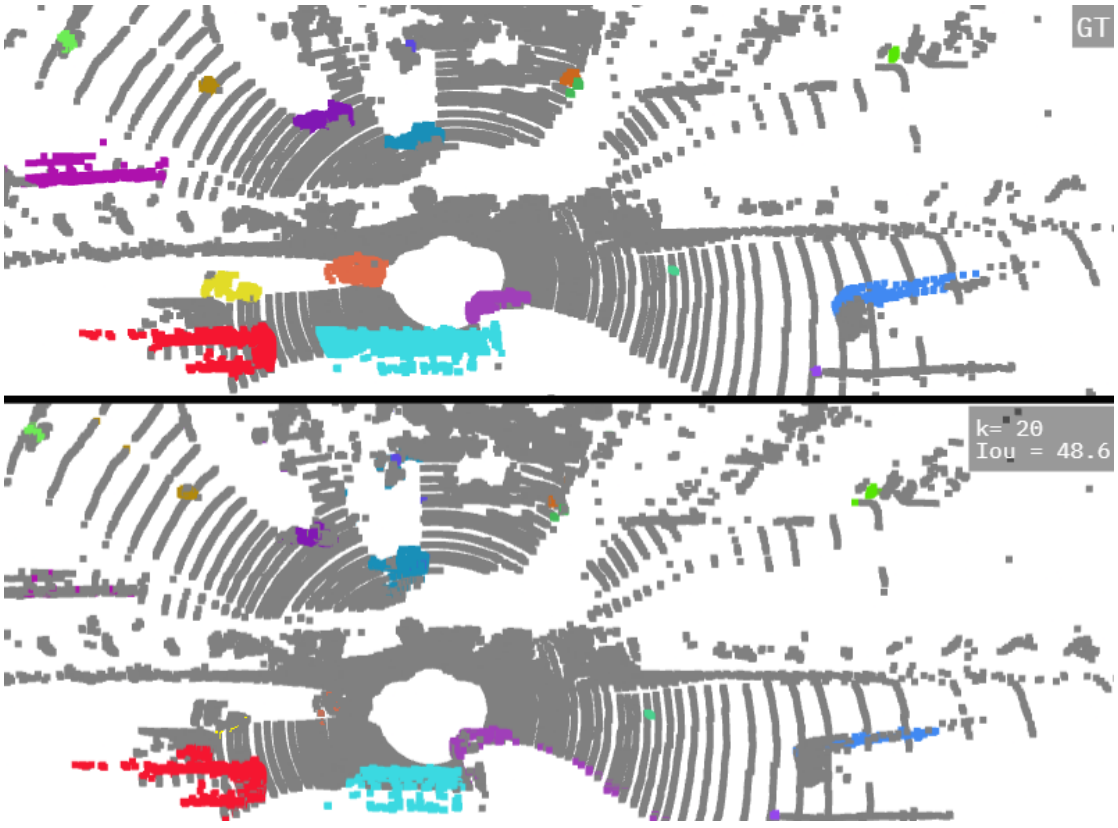

FIGURE 4.6: Prediction Visualization of a Complete ApolloScape Scan

different object characteristics than what the model has been trained on. Figure 4.6 illustrates the model's segmentation results on an ApolloScape scan.

The ApolloScape results reveal more pronounced challenges in segmentation accuracy compared to the KITTI360 scan. The average IoU across instances in this scan is 48.6, indicating a noticeable drop in performance. We observe a higher incidence of both false positives and false negatives in the predicted segmentations. Some objects in the ground truth have very few corresponding points in the prediction, suggesting difficulties in detecting certain objects entirely. Conversely, other predicted objects, while generally resembling their ground truth counterparts, exhibit scattered points extending beyond the true object boundaries.

The performance gap between KITTI360 and ApolloScape aligns with our cross-dataset evaluation findings, which could be attributed to several factors. One factor is the differing urban environments in ApolloScape compared to those in training, presenting new architectural styles, vehicle types, and road layouts. Additionally, variations in LiDAR sensor specifications or data collection methods may lead to differences in point cloud density or distribution. These nuances, though subtle, could significantly impact results. Despite these challenges, the model still demonstrates a degree of effectiveness in segmenting many objects within the ApolloScape scan, suggesting that while RangeSAM's

performance does degrade when applied to significantly different environments, it retains some ability to generalize its learned segmentation strategies.

Finally, our qualitative analysis revealed an important insight into the prompt sampling strategy employed in RangeSAM. Figure 4.7 illustrates a recurring issue we observed in the spatial distribution of prompts.

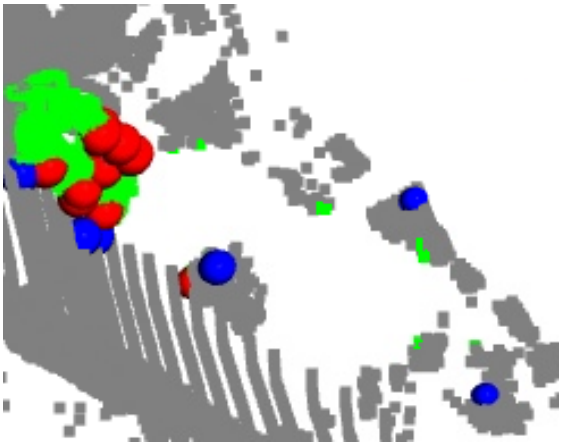

FIGURE 4.7: Prompt Distribution Containing Background and Foreground Artifacts

This visualization highlights that background prompts may possibly be located far from the object of interest, and in rare cases, even foreground prompts can be misplaced outside the object's true boundaries. This suboptimal prompt distribution can lead to several issues:

1. Reduced segmentation accuracy: Distant prompts prompts may cause the model to focus on irrelevant areas of the point cloud, potentially leading to over-segmentation or misidentification of object boundaries, reducing overall segementation quality.
2. Inconsistent instance grouping: Misplaced prompts prompts might result in the inclusion of foreground points from adjacent objects or background regions, contributing to the lower S_{assoc} scores observed in our quantitative analysis.
3. Inefficient use of prompt information: Prompts placed far from the object of interest provide less valuable information for refining the segmentation, potentially explaining why performance improvements diminish with higher prompt counts.

The root cause of this observation lies in our current prompt sampling strategy, which operates in the 2D space of the range image without considering 3D spatial relationships. As touched upon in our methodology, prompts are sampled from a 2D expanded bounding box on the object of interest. This approach simplifies implementation but fails to account for the complex 3D structure of the point cloud. While the range image preserves depth information, our sampling strategy treats the space as a flat 2D

projection when selecting prompts. Consequently, when these 2D-sampled prompts are projected back into 3D space, they can end up in unexpected locations relative to the object of interest, leading to suboptimal prompt placements.

Overall, our qualitative analysis offers deeper insights into RangeSAM's behavior, corroborating and elucidating our quantitative findings. We observe the model's ability to refine segmentations with increasing prompt counts, its effectiveness in handling complex scenes with multiple objects, and its challenges in generalizing to significantly different environments. The analysis also reveals opportunities for improvement, particularly in our prompt sampling strategy.

4.4 Additional Experiments

4.4.1 Dataset Influence on Generalization

To further investigate the impact of training data diversity on RangeSAM's generalization capabilities, we conducted an additional experiment comparing models trained on different datasets. This analysis aimed to determine whether the broader range of object types in KITTI360 enhances the model's ability to generalize to significantly different urban environments.

We trained two versions of RangeSAM: one on KITTI360 and another on SemanticKITTI, both for 50 epochs under similar training conditions trained over 15,000 scans each. We then evaluated both models on the ApolloScape dataset, which represents urban environments distinctly different from the training data. Table 4.3 presents the IoU@K values for these models.

Training Dataset	**IoU@5**	**IoU@10**	**IoU@15**	**IoU@20**
KITTI360	22.92	32.19	37.81	41.89
SemanticKITTI	21.07	29.47	33.57	36.35

TABLE 4.3: Training Source Impact on Model Generalization

The results show that the model trained on KITTI360 consistently outperforms the one trained on SemanticKITTI when evaluated on ApolloScape, with the performance gap widening to over 5% at $K = 20$. These findings suggest that training on a dataset with a broader range of object types and urban scenarios enhances the model's ability to generalize to different environments. KITTI360's diverse semi-urban scenes likely expose the model to a wider variety of object shapes, sizes, and spatial relationships, fostering more robust and adaptable feature representations for novel urban layouts

and object appearances in ApolloScape. Conversely, the more homogeneous suburban environments of SemanticKITTI may limit the model's exposure to the full spectrum of urban complexity, potentially constraining its generalization capabilities.

To gain a more granular understanding of the models' performance across different object types, we conducted a class-wise evaluation on the ApolloScape dataset. Table 4.4 presents the IoU@20 values for each object class for both the KITTI360-trained and SemanticKITTI-trained models.

Model	**Ped.**	**Traffic**	**Cyclist**	**S. Vehicle**	**B. Vehicle**	**Other**
KITTI360-trained	47.04	47.83	44.00	35.76	36.61	41.32
SemanticKITTI-trained	35.45	40.81	33.22	43.84	27.07	32.16

TABLE 4.4: Object Category Performance by Training Source on ApolloScape

The class-wise evaluation reveals interesting patterns in the models' generalization capabilities. The KITTI360-trained model significantly outperforms the SemanticKITTI-trained model in most classes, particularly for pedestrians, cyclists, and big vehicles. This suggests that KITTI360's more diverse urban scenes provide better training for these object types. The SemanticKITTI-trained model, however, shows a noticeable advantage on small vehicles, likely due to its focus on a narrower range of object types, allowing for more exposure to this specific object type during training.

Overall, these results underscore the significant impact of training data diversity on the model's ability to generalize across different object classes and urban environments. The experiment highlights the importance of exposing our model to a wide range of scenarios during training to enhance its adaptability to novel environments. While both datasets are valuable, the broader diversity of KITTI360 appears to confer advantages in cross-dataset generalization, particularly when dealing with significantly different urban layouts and object appearances.

4.4.2 Ablation Study of Novel Architecture Components

To assess the impact of RangeSAM's key innovations, we conducted an extensive ablation study. This analysis centers on three critical components we introduced in the architecture: the learnable scaling factor, the prompt loss function, and prompt label enforcement. These elements were designed to optimize the model's ability to effectively leverage prompts during segmentation.

As discussed in the methodology, the bi-directional transformer's learnable scaling factor enables dynamic adjustments between prompt and image embeddings, allowing the

model to learn a balance that improves context-aware segmentations. The prompt loss function integrates a binary cross-entropy signal within a kernel around each prompt, encouraging the model to focus on nearby regions for more responsive prompt-guided segmentation. Finally, prompt label enforcement aligns predictions with the provided prompts' labels, enhancing consistency and accuracy in segmenting ambiguous regions by reinforcing the importance of prompt locations.

To assess the individual and combined effects of these components, we trained five variants of RangeSAM. Each model was trained for 15 epochs on a subset of 3,000 training scans from the KITTI360 dataset and validated on an additional 1,000 scans. To facilitate a controlled comparison, we employed a fixed set of 20 prompts for the evaluation, while also ensuring that the prompt locations remain consistent across all model variants in each scan. This setup allows us to isolate the impact of each component while controlling for other variables.

The results of our ablation study are presented in Table 4.5, demonstrating that each of our novel components contributes positively to the model's performance, with the combination of all three yielding the best outcome.

Prompt Loss	**Label Enforcement**	**Learnable Scaling**	**IoU**
×	×	×	26.35
×	✓	✓	29.81
✓	×	✓	30.72
✓	✓	×	32.02
✓	✓	✓	34.32

Table 4.5: RangeSAM Component Ablation Results

To understand each component's contribution, we systematically removed one component while keeping the other two active. The prompt loss function proved to be the most crucial component, as its removal resulted in the largest performance drop, with IoU decreasing by 4.51% (from 34.32 to 29.81). Label enforcement showed the second-highest impact, with its removal leading to a decrease of 3.6% (to 30.72 IoU). Finally, while still beneficial, the learnable scaling factor had the smallest individual impact, with its removal resulting in a performance drop of 2.3% (to 32.02 IoU).

When all components work together, we observe a synergistic effect, with the full model achieving 34.32 IoU, which is 7.97% higher than the baseline in which all components are disabled. This demonstrates that while each component makes a valuable individual contribution, their combination leads to the most effective promptable segmentation

system. The prompt loss provides the foundation for effective prompt utilization, enhanced by label enforcement's consistency constraints and the learnable scaling factor's dynamic information balancing.

To further validate our findings, we assessed the cross-dataset generalization of the two extremes of the KITTI360-trained models: the fully equipped RangeSAM model with all three components, and the baseline version not containing any of these components. Both models were tested on the ApolloScape dataset using a consistent setup of 1000 test scans, each with 20 prompts. The results are detailed in Table 4.6.

RangeSAM Variant	**IoU**
All Components	18.95
No Components	16.71

TABLE 4.6: Cross-Dataset Performance of Ablation Variants

The results demonstrate that our architectural improvements enhance not only performance but also generalization capabilities. When evaluated on the previously unseen ApolloScape dataset, our full RangeSAM model maintains its advantage, achieving an IoU of 18.95 compared to 16.71 for the baseline model. While the overall performance is lower than on KITTI360 due to the domain shift, the relative improvement of 2.24% indicates that our proposed components contribute to better generalization across different LiDAR datasets. This suggests that the architectural benefits of our components are not dataset-specific but represent genuine improvements in the model's ability to handle point cloud segmentation tasks.

To conclude, the ablation study and the subsequent analysis reveal the individual and synergistic contributions of RangeSAM's novel components, underscoring the effectiveness of our multi-faceted approach to integrating prompt information, which enhances segmentation performance in sparse outdoor LiDAR point clouds. However, this study was limited to a subset of the full dataset and a restricted number of training epochs, indicating that further exploration with extended training and larger datasets may uncover deeper insights into the long-term effects and interactions of these components.

Chapter 5

Conclusion

5.1 Overview

This research set out to address a critical gap in 3D point cloud segmentation: the lack of promptable segmentation models capable of handling the varying point densities characteristic of LiDAR scans. Our central research question asked whether integrating range projection with attention mechanisms could mitigate the challenge of sparsity in outdoor LiDAR scans, enabling promptable segmentation in them. The findings suggest a qualified "yes" to this question, albeit with important nuances that merit discussion.

RangeSAM demonstrates robust performance across varying point densities, achieving competitive results against both promptable and class-agnostic segmentation models. The model's ability to maintain strong performance on sparse data, particularly evident in our experiments, indicates that our approach successfully adapts to sparse data. The rapid improvement in segmentation quality with increasing prompt counts, coupled with strong point-level accuracy metrics, suggests that RangeSAM leverages point prompts effectively to guide segmentation in challenging outdoor environments.

Furthermore, the cross-dataset evaluation results, particularly on ApolloScape, indicate that our approach generalizes reasonably well to unseen urban environments, albeit with some performance degradation. This underscores the ongoing challenge of domain adaptation in 3D point cloud segmentation. Additionally, the lower instance-level coherence scores compared to a state-of-the-art conventional class-agnostic model suggest that while RangeSAM excels at point-level classification, it sometimes struggles with grouping points into coherent object instances.

Our findings both align with and extend existing literature. While previous works have demonstrated the effectiveness of promptable segmentation in dense point clouds, our

research shows that these principles can be adapted to sparse LiDAR data through careful architectural design. However, the challenge of evaluating our work against existing methods highlighted a broader issue in the field: the lack of standardized benchmarks for promptable 3D segmentation in sparse environments. Despite our attempts to bridge this gap through careful experimental design, the difficulty in making direct comparisons underscores the nascent nature of this research direction.

5.2 Contribution

This research makes several significant contributions to the field of 3D computer vision and point cloud processing. First and foremost, it introduces a novel architecture that bridges the gap between promptable segmentation and sparse LiDAR processing, representing the first dedicated attempt to address the challenges of varying point densities in 3D promptable segmentation. This opens new possibilities for interactive scene understanding in outdoor environments.

The integration of range projection and attention mechanisms in RangeSAM introduces a novel paradigm for handling sparse 3D data. Our approach demonstrates that effective promptable segmentation is achievable even in sparse outdoor scenarios, paving the way for future advancements. Furthermore, the three architectural innovations we introduced—the learnable scaling factor, prompt loss function, and prompt label enforcement—significantly improved promptable segmentation performance as shown in our ablation study and could be valuable additions to other promptable segmentation approaches in both 2D and 3D domains.

From a practical perspective, our work has immediate implications for applications in autonomous driving and urban scene understanding. RangeSAM's ability to handle sparse LiDAR data while maintaining strong segmentation performance makes it particularly valuable for real-world scenarios where point density varies significantly with distance and environmental conditions. The model's prompt-driven nature offers flexibility in deployment. It supports user-guided refinement for efficient outdoor data annotation, and although not explored in this research, its architecture could accommodate automated prompt generation processes to enable scene-wide segmentation.

Furthermore, our research contributes to the ongoing discourse about the role of user interaction in 3D scene understanding. By demonstrating the effectiveness of point prompts in guiding segmentation, even in challenging sparse environments, we provide evidence for the viability of interactive approaches in practical 3D vision systems. This has implications beyond autonomous driving, potentially influencing applications in

robotics, augmented reality, and urban planning where user-guided scene understanding is valuable.

5.3 Future Work

While RangeSAM has demonstrated significant progress in promptable segmentation of sparse LiDAR point clouds, several areas offer opportunities for future improvement:

1. Prompt Sampling in 3D Space: The current prompt sampling strategy operates solely in the 2D space of range images, occasionally generating prompts that are spatially distant from target objects in 3D space. Incorporating depth information into the sampling process or implementing 3D sampling could enhance prompt placement and improve segmentation accuracy.

2. Automated Scene Segmentation: Unlike 2D promptable segmentation models such as SAM, which have the ability to automatically propose object regions via single-point foreground prompts without reference labels, RangeSAM currently does not support this functionality. Future work could focus on improving single-point prompt performance or incorporating multiple prompts to enable efficient automated scene segmentation in LiDAR data.

3. Iterative Refinement: Due to computational limitations, we were unable to implement iterative refinement schemes during training. Future work with greater resources could explore sequential prompt placements based on error regions, imitating human-like interaction more closely and potentially improving segmentation performance.

4. Evaluation Protocols: Although our experimental design aimed to contextualize RangeSAM's performance, direct comparisons with existing models on sparse point clouds were constrained by unavailable model weights and architectural incompatibilities. Developing standardized evaluation protocols for sparse 3D data in promptable settings would foster more comprehensive comparisons as the field matures.

5. Generalization to New Environments: The model currently benefits from training on specific urban environments, leaving room for improvement in handling diverse landscapes. Future research could involve training with a broader range of outdoor datasets, including synthetic data, and exploring dataset fusion techniques to enhance generalization across varied environments.

These challenges point toward exciting directions for future research. Addressing them would likely yield a more robust and generalizable framework for promptable segmentation of sparse point clouds, while reinforcing the progress already made by RangeSAM.

Bibliography

[1] Radu Bogdan Rusu and Steve Cousins. 3d is here: Point cloud library (pcl). In *2011 IEEE international conference on robotics and automation*, pages 1–4. IEEE, 2011.

[2] Ying Li, Lingfei Ma, Zilong Zhong, Fei Liu, Michael A Chapman, Dongpu Cao, and Jonathan Li. Deep learning for lidar point clouds in autonomous driving: A review. *IEEE Transactions on Neural Networks and Learning Systems*, 32(8):3412–3432, 2020.

[3] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012.

[4] Yang Zhang, Zixiang Zhou, Philip David, Xiangyu Yue, Zerong Xi, Boqing Gong, and Hassan Foroosh. Polarnet: An improved grid representation for online lidar point clouds semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9601–9610, 2020.

[5] Yuxing Xie, Jiaojiao Tian, and Xiao Xiang Zhu. Linking points with labels in 3d: A review of point cloud semantic segmentation. *IEEE Geoscience and remote sensing magazine*, 8(4):38–59, 2020.

[6] Lucas Nunes, Xieyuanli Chen, Rodrigo Marcuzzi, Aljosa Osep, Laura Leal-Taixé, Cyrill Stachniss, and Jens Behley. Unsupervised class-agnostic instance segmentation of 3d lidar data for autonomous vehicles. *IEEE Robotics and Automation Letters*, 7(4):8713–8720, 2022.

[7] Jun Cen, Peng Yun, Junhao Cai, Michael Yu Wang, and Ming Liu. Open-set 3d object detection. In *2021 International conference on 3D vision (3DV)*, pages 869–878. IEEE, 2021.

[8] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al.

Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

[9] Yuchen Zhou, Jiayuan Gu, Tung Yen Chiang, Fanbo Xiang, and Hao Su. Pointsam: Promptable 3d segmentation model for point clouds. *arXiv preprint arXiv:2406.17741*, 2024.

[10] Yuanwen Yue, Sabarinath Mahadevan, Jonas Schult, Francis Engelmann, Bastian Leibe, Konrad Schindler, and Theodora Kontogianni. Agile3d: Attention guided interactive multi-object 3d segmentation. *arXiv preprint arXiv:2306.00977*, 2023.

[11] 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.

[12] Lingdong Kong, Youquan Liu, Runnan Chen, Yuexin Ma, Xinge Zhu, Yikang Li, Yuenan Hou, Yu Qiao, and Ziwei Liu. Rethinking range view representation for lidar segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 228–240, 2023.

[13] Theodora Kontogianni, Ekin Celikkan, Siyu Tang, and Konrad Schindler. Interactive object segmentation in 3d point clouds. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2891–2897. IEEE, 2023.

[14] Martin Weinmann. *Reconstruction and analysis of 3D scenes: From irregularly distributed 3D points to object classes*. Springer, 2016.

[15] Jiaying Zhang, Xiaoli Zhao, Zheng Chen, and Zhejun Lu. A review of deep learning-based semantic segmentation for point cloud. *IEEE access*, 7:179118–179133, 2019.

[16] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep learning for 3d point clouds: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(12):4338–4364, 2020.

[17] Mingqiang Wei, Zeyong Wei, Haoran Zhou, Fei Hu, Huajian Si, Zhilei Chen, Zhe Zhu, Jingbo Qiu, Xuefeng Yan, Yanwen Guo, et al. Agconv: Adaptive graph convolution on 3d point clouds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(8):9374–9392, 2023.

[18] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9297–9307, 2019.

[19] Weiqi Zhou. An object-based approach for urban land cover classification: Integrating lidar height and intensity data. *IEEE Geoscience and Remote Sensing Letters*, 10(4):928–931, 2013.

[20] MathWorks. Introduction to lidar - principles of lidar. https://www.mathworks.com/help/lidar/ug/lidar-processing-overview.html, 2021. Principles of Lidar.

[21] Mathieu Dassot, Thiéry Constant, and Meriem Fournier. The use of terrestrial lidar technology in forest science: application fields, benefits and challenges. *Annals of forest science*, 68:959–974, 2011.

[22] Bernhard Höfle and Martin Rutzinger. Topographic airborne lidar in geomorphology: A technological perspective. *Zeitschrift fur Geomorphologie-Supplementband*, 55(2):1, 2011.

[23] Iván Puente, H González-Jorge, J Martínez-Sánchez, and Pedro Arias. Review of mobile mapping and surveying technologies. *Measurement*, 46(7):2127–2145, 2013.

[24] Clément Mallet and Frédéric Bretar. Full-waveform topographic lidar: State-of-the-art. *ISPRS Journal of photogrammetry and remote sensing*, 64(1):1–16, 2009.

[25] You Li and Javier Ibanez-Guzman. Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems. *IEEE Signal Processing Magazine*, 37(4):50–61, 2020.

[26] Thomas Luhmann, Stuart Robson, Stephen Kyle, and Jan Boehm. *Close-range photogrammetry and 3D imaging*. Walter de Gruyter GmbH & Co KG, 2023.

[27] Matthew J Westoby, James Brasington, Niel F Glasser, Michael J Hambrey, and Jennifer M Reynolds. 'structure-from-motion'photogrammetry: A low-cost, effective tool for geoscience applications. *Geomorphology*, 179:300–314, 2012.

[28] Jason Geng. Structured-light 3d surface imaging: a tutorial. *Advances in optics and photonics*, 3(2):128–160, 2011.

[29] Larry Li et al. Time-of-flight camera—an introduction. *Technical white paper*, 2014.

[30] Yuquan Xu, Vijay John, Seiichi Mita, Hossein Tehrani, Kazuhisa Ishimaru, and Sakiko Nishino. 3d point cloud map based vehicle localization using stereo camera. In *2017 IEEE intelligent vehicles symposium (IV)*, pages 487–492. IEEE, 2017.

[31] Yiyi Liao, Jun Xie, and Andreas Geiger. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3292–3310, 2022.

[32] Rui Zhang, Yichao Wu, Wei Jin, and Xiaoman Meng. Deep-learning-based point cloud semantic segmentation: A survey. *Electronics*, 12(17):3642, 2023.

[33] Tzu-Hsuan Chen and Tian Sheuan Chang. Rangeseg: Range-aware real time segmentation of 3d lidar point clouds. *IEEE Transactions on Intelligent Vehicles*, 7 (1):93–101, 2021.

[34] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 945–953, 2015.

[35] Charles R Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5648–5656, 2016.

[36] Alexandre Boulch, Joris Guerry, Bertrand Le Saux, and Nicolas Audebert. Snapnet: 3d point cloud semantic labeling with 2d deep segmentation networks. *Computers & Graphics*, 71:189–198, 2018.

[37] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 922–928. IEEE, 2015.

[38] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3577–3586, 2017.

[39] Lyne Tchapmi, Christopher Choy, Iro Armeni, JunYoung Gwak, and Silvio Savarese. Segcloud: Semantic segmentation of 3d point clouds. In *2017 international conference on 3D vision (3DV)*, pages 537–547. IEEE, 2017.

[40] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[41] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

[42] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11108–11117, 2020.

[43] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)*, 38(5):1–12, 2019.

[44] Li Jiang, Hengshuang Zhao, Shu Liu, Xiaoyong Shen, Chi-Wing Fu, and Jiaya Jia. Hierarchical point-edge interaction network for point cloud semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10433–10441, 2019.

[45] 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.

[46] 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: 187–199, 2021.

[47] Bichen Wu, Alvin Wan, Xiangyu Yue, and Kurt Keutzer. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 1887–1893. IEEE, 2018.

[48] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. Rangenet++: Fast and accurate lidar semantic segmentation. In *2019 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 4213–4220. IEEE, 2019.

[49] Angelika Ando, Spyros Gidaris, Andrei Bursuc, Gilles Puy, Alexandre Boulch, and Renaud Marlet. Rangevit: Towards vision transformers for 3d semantic segmentation in autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5240–5250, 2023.

[50] Weiyue Wang, Ronald Yu, Qiangui Huang, and Ulrich Neumann. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2569–2578, 2018.

[51] Jingdao Chen, Zsolt Kira, and Yong K Cho. Lrgnet: Learnable region growing for class-agnostic point cloud segmentation. *IEEE Robotics and Automation Letters*, 6 (2):2799–2806, 2021.

[52] Dipesh Gyawali, Jian Zhang, and Bijaya B Karki. Region-transformer: Self-attention region based class-agnostic point cloud segmentation. *arXiv preprint arXiv:2403.01407*, 2024.

[53] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[54] Yunhan Yang, Xiaoyang Wu, Tong He, Hengshuang Zhao, and Xihui Liu. Sam3d: Segment anything in 3d scenes. *arXiv preprint arXiv:2306.03908*, 2023.

[55] Mutian Xu, Xingyilang Yin, Lingteng Qiu, Yang Liu, Xin Tong, and Xiaoguang Han. Sampro3d: Locating sam prompts in 3d for zero-shot scene segmentation. *arXiv preprint arXiv:2311.17707*, 2023.

[56] Rui Huang, Songyou Peng, Ayca Takmaz, Federico Tombari, Marc Pollefeys, Shiji Song, Gao Huang, and Francis Engelmann. Segment3d: Learning fine-grained class-agnostic 3d segmentation without manual labels. *arXiv preprint arXiv:2312.17232*, 2023.

[57] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019.

[58] Junsheng Zhou, Jinsheng Wang, Baorui Ma, Yu-Shen Liu, Tiejun Huang, and Xinlong Wang. Uni3d: Exploring unified 3d representation at scale. *arXiv preprint arXiv:2310.06773*, 2023.

[59] Xinyu Huang, Xinjing Cheng, Qichuan Geng, Binbin Cao, Dingfu Zhou, Peng Wang, Yuanqing Lin, and Ruigang Yang. The apolloscape dataset for autonomous driving. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 954–960, 2018.

[60] Marcus Schilling. recover-kitti360-label. https://github.com/MarcusSchilling/recoverKITTI360label, 2023.

[61] Yoshua Bengio, Ian Goodfellow, and Aaron Courville. *Deep learning*, volume 1. MIT press Cambridge, MA, USA, 2017.

[62] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009.

[63] Tete Xiao, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollár, and Ross Girshick. Early convolutions help transformers see better. *Advances in neural information processing systems*, 34:30392–30400, 2021.

[64] Tiago Cortinhal, George Tzelepis, and Eren Erdal Aksoy. Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds. In *Advances in Visual Computing: 15th International Symposium, ISVC 2020, San Diego, CA, USA, October 5–7, 2020, Proceedings, Part II 15*, pages 207–222. Springer, 2020.

[65] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[66] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[67] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in neural information processing systems*, 33:7537–7547, 2020.

[68] Won-Dong Jang and Chang-Su Kim. Interactive image segmentation via backpropagating refinement scheme. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5297–5306, 2019.