Empower dynamic scene understanding through scene flow estimation and object segmentation

by

Zhiqi Li

National Centre for Computer Animation

Faculty of Media & Communication

Bournemouth University

A thesis submitted in partial fulfilment of the requirements of Bournemouth University for the degree of *Doctor of Philosophy*

April. 2025

Copyright Statement

Abstract

Understanding dynamic 3D scenes—critical for applications like autonomous navigation and mixed reality—requires parsing both motion (scene flow) and object interactions (segmentation). Scene flow captures 3D motion fields, while segmentation isolates objects, enabling systems to interpret evolving environments. Integrating these tasks offers a holistic view but faces computational challenges due to scene flow's high dimensionality.

This work proposes a lightweight deep learning architecture combining an enhanced Point Transformer for efficient feature extraction and a point-voxel correlation module for stable motion estimation.

To bypass labor-intensive object annotations, scene flow is leveraged as auxiliary supervision. Instead of predicting masks for all points, this thesis focuses on key points, reducing complexity while maintaining accuracy. The proposed clustering-free approach achieves state-of-the-art results on indoor datasets.

For temporal consistency, an unsupervised method integrates continuous point cloud sequences (encoding spatial embeddings) with time-independent queries (encoding object semantics). This enables gradual mask prediction across frames without direct labels, accommodating dynamic inputs. This framework advances dynamic scene understanding by harmonizing motion and segmentation, validated through competitive benchmarks and flexible input handling.

Acknowledgements

Throughout the writing of this thesis, I have received a great deal of support and assistance from my supervisors, classmates, friends, and family members. Foremost, I wish to express my deepest gratitude to my Ph.D. supervisor, Prof. Xiaosong Yang, for his patience, encouragement, and exceptional mentorship. His profound professionalism and enduring passion for scientific inquiry have shaped my academic journey. I am immensely grateful for Prof. Yang's openness in supporting my independent research initiatives and extracurricular engagements, as well as his empathetic guidance during challenging phases of my doctoral studies. His unique balance of offering sage advice while empowering me to pursue self-directed research topics and career aspirations has been instrumental in fostering both my intellectual growth and personal resilience.

I extend my heartfelt gratitude to Bournemouth University for their support in facilitating my participation in international conferences, deep learning summer schools, and the Turing Overseas Scheme.

I am deeply indebted to my colleagues, collaborators, and mentors for their intellectual generosity. The insightful discussions we shared have profoundly shaped my academic growth and refined the ideas presented in this work. Special thanks

to the staff of the NCCA department, the FMC research administration team, and the Doctoral College for their assistance in navigating administrative processes and supporting my research milestones.

I am also grateful to The Alan Turing Institute, The Hong Kong Polytechnic University (PolyU), and the Chinese Scholarship Council for providing me with valuable opportunities, including research attachments and funding my doctoral studies. My sincere appreciation goes to Dr. Bo Yang and Mr. Ziyang Song at PolyU, whose mentorship and spirit of discovery ignited my curiosity and deepened my technical expertise. Their dedication to innovation has left an indelible mark on my approach to research.

Finally, to my parents and my partner—your unconditional love, patience, and encouragement have been my steadfast anchor throughout this journey. Thank you for inspiring me to persevere with resilience and confidence.

Declaration

I, Zhiqi Li, declare that this thesis is submitted in fulfillment of the requirements for the Degree of Doctor of Philosophy, represents my own work except where due acknowledgement have been made. I further declared that it has not been previously included in a thesis, dissertation, or report submitted to this University or to any other institution for a degree, diploma or other qualifications.

Signed: Li zhi qi

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivations

Most organisms perceive their surrounding environment through their eyes, benefiting from biological visual sensors that allow them to interact with their surroundings and make safe decisions. However, for autonomous systems and intelligent robotics, understanding the observed scene and identifying potential dynamic objects is extremely challenging (Muhammad et al. 2022), due to the processing of a large amount of complex data resulting from the diversity and variability of visual information, as well as the need for low-latency perceptual reasoning (Falanga et al. 2019).

The stack of tasks in understanding dynamic scenes is inherently compositional. In the human brain, object- and scene-agnostic estimation of apparent motion occurs at an early stage in the visual cortex (Hubel and Wiesel 1968), characterized by very low latency. This initial processing stage provides crucial information for low-latency body control and reaction, enabling actions like catching an incoming ball with millisecond precision. Higher-level recognition of known objects takes place later in

the visual cortex utilizing motion estimates. Although this higher-level reasoning operates with a larger latency, it builds on the foundational motion estimation mechanism provided by the early stages of visual processing. Thus, the estimation of apparent motion serves as a vital input for both high-level visual reasoning and low-latency behavioral responses, demonstrating its importance across multiple levels of dynamic scene understanding.

Therefore, the aim of this thesis is to take advantage of the compositional structure of scenes to enhance the understanding of dynamic scenes. This complex vision can be further split into two sub-problems: **scene flow estimation** and **object segmentation**.

Dynamic scene understanding is important for effectively interacting with the environment. It is not only paramount in computer vision, but also in robotics and neuroscience. This insight has driven extensive research into estimating 3D motion in point cloud sequences (Liu et al. 2019b, Huang et al. 2022a) and segmenting distinct objects over time (Chen et al. 2021a, Sun et al. 2020). However, despite significant scientific advances of the past decades, these tasks are not solved to satisfaction yet.

Humans naturally perceive the surrounding environment in three dimensions. Scene flow is mathematically described as the 3D displacements between two consecutive frames (Vedula et al. 1999). However, scene flow estimation is a high-dimensional and computationally demanding task (Vedder et al. 2023). At its core, scene flow estimation is a pointwise matching problem in 3D vision, which requires one-to-one correspondence computation. Moreover, the unordered and uneven distribution of 3D point clouds presents a challenge in determining the appropriate target point for computing the flow vector from a given source point. To address these problems, many frameworks (Xu et al. 2022, Gu et al. 2019, Liu et al. 2019a, Kittenplon et al. 2021) propose to learn point

cloud features and compute feature similarity to obtain point correspondences in a trainable fashion. However, the dynamic nature of objects themselves can cause them to move unpredictably, making it difficult to establish stable correspondences across frames.

Thus, the first research question (RQ) studied in this thesis is: **RQ1:** ***how to ensure efficiency*** (fast feature extraction on point cloud) and ***efficacy*** (stable point correspondence learning) in scene flow estimation. Intuitively, this question requires solutions of feature learning for unordered point cloud data and the dense matching problem in a trainable fashion to fit available scene flow datasets.

Once the scene flow estimation task is addressed, the subsequent question pertains to **RQ2:** ***how to effectively segment 3D object with observed motion patterns***. The complexity of this problem arises from the inherent variability in the motion patterns of objects. Moreover, the diversity in object types and shapes adds another layer of complexity to the segmentation process. Segmenting object in given dynamic sequences is quite tricky without any annotations. Typical method of object segmentation leverage clustering-based algorithms (Ahmed and Chew 2020, Zhang et al. 2020). However, these methods could be significantly affected when the data contains noise or outliers, which leads to unstable clustering results or even incorrect segmentation. As presented in the Gestalt theory (Fussell 2023), humans subconsciously impose pattern and structure on visual representation. This indicates that the identification and segmentation of individual objects could be addressed with compositional structure of motion patterns rather than rely on classical clustering-based algorithms (Ester et al. 1996).

Unlike supervised segmentation, which relies on annotated data, unsupervised methods are more flexible to segment objects in dynamic scenes. Therefore, a natural question arises: Does unsupervised method still useful in segmenting objects where their positions and appearances

change across frames? If so, **RQ3:** ***how to obtain consistent object segmentation through unsupervised learning?*** Furthermore, this unsupervised method should be flexible to varying number of input frames and maintain segmentation consistency of multiple frames.

In conclusion, by combining unsupervised segmentation with scene flow estimation, it is possible to develop a framework that identifies, segments, and tracks objects without the need for expensive labeled data. This advancement not only propels progress in 3D scene flow estimation but also establishes robust computational frameworks for critical downstream applications including object recognition, motion analysis, and trajectory prediction. The framework's robustness is evident in its ability to handle partial occlusions, sensor noise, and complex motion patterns through an unsupervised learning mechanism.

1.2 Research Objectives

Following the research questions discussed above, this thesis aims to achieve three major objectives:

- **A novel scene flow estimation model:** To tackle the challenge of scene flow estimation mentioned before, this thesis aims to develop an architecture that can balance efficacy (stable flow estimation) and efficiency (fast feature aggregation). The first sub-objective is to construct a network architecture that can directly detect both local and global patterns, such as small object movements within localized regions and broader changes across the entire scene. The learned point features is then used in point correlation reasoning. Thus the second sub-objective is to develop an effective point correlation module that can generate scene flow between consecutive point clouds.

- **A clustering-free object segmentation framework:** The second aim of this thesis is to effectively segment 3D objects from

observed motion patterns. This includes exploring techniques that can segment point clouds into distinct objects without relying on any object-level labels and clustering algorithms. A sub-objective focuses on developing strategies to handle over-segmentation (a rigid object has multiple labels) and under-segmentation (two or more objects share the same label) issue simultaneously.

- **Unsupervised object segmentation from multi-frame point clouds:** The third aim of this thesis is to develop a multi-frame object segmentation network. The initial sub-objective is to establish an effective learning framework for segmenting multi-frame point clouds. The subsequent sub-objective focuses on enhancing the robustness of the segmentation model and ensuring temporal consistency on multi-frame input.

1.3 Contributions

This thesis contributes to the field of dynamic scene understanding, specifically within the context of scene flow estimation and object segmentation from point cloud sequences. The contributions of this thesis follow the above motivations.

C1: Literature Review This thesis provides a comprehensive comparison and in-depth analysis of recent deep learning methods for scene flow estimation from 2019 to 2023, covering supervised, weakly-supervised, and self-supervised approaches. In addition, an overview of current challenges in scene flow estimation, which are categorized into data-related challenges and deep learning-specific challenges is presented. Furthermore, a review of static object segmentation and dynamic object segmentation is outlined.

C2: A novel scene flow estimation model This thesis studies the problem of estimating scene flow from two consecutive point clouds. The proposed method integrates the local feature learning and global feature

learning via an improved point Transformers. The point Transformer allows the proposed model to capture fine-grained point features efficiently while maintains a global understanding of scene context. Furthermore, a global motion aggregation module boosts the efficacy of point-voxel correlation learning. The proposed method achieves competitive results compared to other learning frameworks.

C3: A clustering-free object segmentation framework A clustering-free, compact framework is studied to predict 3D object segmentation masks via key points. It is inspired by the physical moving patterns, which indicates that object points essentially move in groups or sets. Consequently, the per-point mask can be predicted by only optimizing on a smaller set of points, which reduces the computational burden. Unlike previous approaches that rely on clustering methods or object detectors, this study utilizes scene flow as auxiliary supervisory. The proposed method achieves state-of-the-art segmentation results on an indoor dataset, as well as its single-view counterpart. The Indoor Dynamic Room dataset is selected for its complex motion patterns and cluttered layouts, which inherently exacerbate segmentation challenges due to occlusions and partial object visibility. Moreover, this framework is enhanced by leveraging multi-frame sequence inputs, which ensures temporal consistency across frames.

C4: Unsupervised object segmentation on multi-frame point clouds Based on **C3**, an end-to-end trainable architecture is studied in this thesis. While short-term observations often fail to capture the complete shape of an object, combining multiple frames provides a more comprehensive view. This thesis verifies that continuous observations over multiple frames are more beneficial for segmenting moving objects. The experimental results show that this multi-frame approach yields more accurate and robust segmentation outcomes. By utilizing key points coupled with sequence of continuous point cloud frames (that provide point embeddings) and a time-independent query (that provided

object embedding), the proposed network is able to predict gradual segment mask without direct supervision. Moreover, it allows a more flexible input of dynamic sequence.

1.4 Organizations

This thesis is laid out as follows.

Chapter 2 introduces synthetic and real scene flow datasets, followed by an overview of previous works in scene flow esitmation and point cloud segmentation.

Chapter 3 presents a Transformer-based paradigm for scene flow estimation, alongside experimental results on the FlyingThings3D and KITTI benchmarks.

Chapter 4 introduces a clustering-free object segmentation algorithm that leverages key points and kernel function to improve object segmentation accuracy without any direct object-level labels.

Chapter 5 outlines an unsupervised learning method for object segmentation in dynamic sequences, describing the multi-frame segmentation pipeline for moving object segmentation in detail. Finally a discussion is presented on how the utilization of object masks can enhance scene flow estimation.

Chapter 6 concludes the presented work and gives an outlook for future research.

2 Literature Review

In this thesis, the research of point cloud scene flow estimation is conducted using a deep-learning method while the task of object segmentation is addressed in both non-learning and deep learning manner. To begin with, this chapter demonstrates related datasets for scene flow estimation (Chapter 2.1). Then this chapter gives a comprehensive review on existing deep-learning methods for scene flow estimation (Chapter 2.2). Finally, to follow the course of this research, which is to segment individual object in given point cloud sequence, recent advancement of static segmentation and dynamic segmentation approaches are introduced. (Chapter 2.3).

2.1 Scene Flow Datasets

As an analog of optical flow (Xu et al. 2022), 3D scene flow has attracted increasing research attention in recent years. Scene flow (shown in Fig 2.1) is defined as the 3D motion field that describes the movement of each point in a scene over time (Li et al. 2022d), providing essential information for understanding dynamic environments. Scene flow esti-

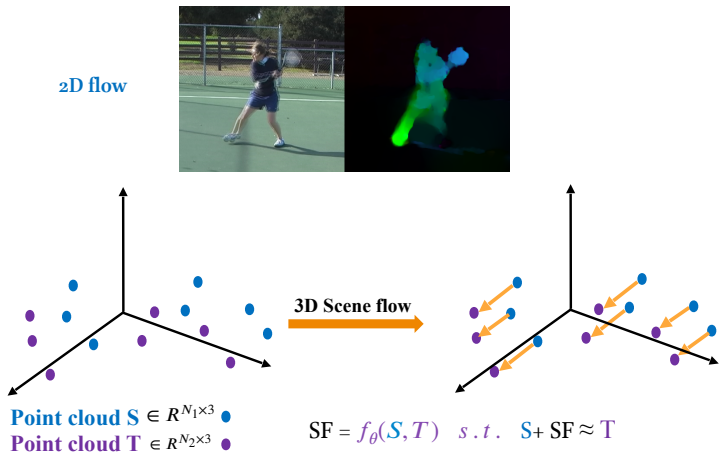


Figure 2.1: While 2D optical flow (upper figure from (Brox and Malik 2010)) quantifies the apparent motion of pixels between consecutive 2D image frames, 3D scene flow (lower figure) constitutes its extension into three-dimensional space. By employing depth sensors such as LiDAR, two temporally consecutive point clouds, denoted as S (source) and T (target), are acquired. The 3D scene flow is defined as the 3D vector field that associates each point with a displacement vector SF, which maps S to its corresponding location in the subsequent point cloud T.

mation datasets are designed to capture this motion across consecutive frames, offering valuable benchmarks for developing and evaluating algorithms in this area. These datasets can vary significantly in their data sources, each bringing unique challenges and advantages.

In the context of dynamic scenes represented by point clouds, synthetic data refers to data created through software with manually designed attributes. These datasets provide the necessary ground truth information for models to learn from, enabling them to gradually recognize specific entities of the real world. Real data, on the other hand, typically refers to data collected by laser scanners, Kinect sensors, and LiDAR sensors in actual driving scenarios. This section introduces and compares point cloud datasets for scene flow estimation. A taxonomic study is presented in terms of the source of the data, as elaborated in Fig. 2.2.

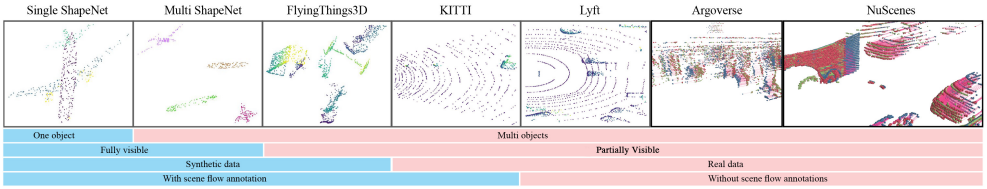


Figure 2.2: Illustration and summarization of the differences between datasets: Single ShapeNet, Multi ShapeNet (Chang et al. 2015), FlyingThings3D (Mayer et al. 2016), KITTI Object (Menze and Geiger 2015), Lyft (Houston et al. 2020), Argoverse (Chang et al. 2019), and NuScenes (Caesar et al. 2020). For clarify here, two datasets (NuScenes (Caesar et al. 2020) and Argoverse (Chang et al. 2019)) are added based on the version of (Zuanazzi et al. 2020).

2.1.1 Synthetic datasets

- **Single ShapeNet** is made of one moving object in a single scene and is fully visible. The geometry information of the object does not change between frames. **Multi ShapeNet** extends the complexity of the whole scene by introducing additional objects. Although the geometry of individual object is always kept consistent, the geometry of the scene may unsteadily change. The two datasets are generated from ShapeNet (Chang et al. 2015) where the objects are represented by point cloud. Each 3D object in the second frame is yielded through a transformation matrix.

- **Flyingthings3D** is a synthetic dataset tailored for tasks such as optical flow, disparity, and scene flow estimation. It introduces multiple partial visible objects, which means different objects may occlude each other, and there are some objects excluded in the scene. It contains over 35,000 stereo image pairs with ground truth disparity, optical flow, and scene flow. The training set consists of 19,640 examples and the test set has 3,824 examples. Existing literature proposed two versions of this dataset for scene flow estimation task. The first strategy of data preprocessing proposed by HPLFlowNet (Gu et al. 2019) directly removed occlusion from the raw data, which ensures hard correspondences between two frames.

Name	Avg points per frame	Train	Test	Scenes	Resolution	D&N	Traf.	Annotation
KITTI2015 (Menze et al. 2015)	N/A	150	50	22	(375,1242)	✗	urban, rural	150 frames
LiDAR KITTI (Geiger et al. 2012a)	120K	N/A	N/A	N/A	N/A	✗	urban	Occlusion labels, 3D labels
NuScenes (Caesar et al. 2020)	34K	1,513	310	1,000	N/A	✓	urban	40K frames
Waymo (Sun et al. 2020)	117K	N/A	N/A	1,150	(1920, 1280/1040)	✓	urban	230K frames
Argoverse (Chang et al. 2019)	107K	2,691	212	113	(2056,2464)	✗	urban	22K frames
Lyft (Kesten et al. 2019)	N/A	18,900	3,780	22,680	N/A	✗	urban	46K frames

Table 2.1: Open real-world datasets. "Avg points per frame" refers to LiDAR returns. "N/A" indicates unavailable data. "Train" refers to the number of training samples, "Test" refers to the number of testing samples. "D&N" refers to day & night. "Traf." refers to traffic conditions.

The second strategy is proposed by FlowNet3D (Liu et al. 2019a). FlowNet3D preserves occluded point clouds as well as masks that indicate the invalid points without corresponding ones in the subsequent frame.

- **GTA-SF** is proposed by DCA-SRSFE (Jin et al. 2022) for synthesizing real-world scenarios. GTA-SF has 54,287 pairs of consecutive point clouds with dense annotations. It collects larger-scale and more realistic point clouds than existing synthetic datasets. Another advantage of GTA-SF is the rich variety of scenarios. The data was collected from downtown areas, highways, streets and other driving areas along six different routes at outdoor areas.

2.1.2 Real datasets

As shown in Table. 2.1, this section summarize the key properties (e.g., the scale of point clouds, resolution, annotations, etc.) of real scene

datasets used by current scene flow estimation approaches.

- **LiDAR KITTI** (Geiger et al. 2012a) was originally proposed in 2012 for stereo matching and optical flow estimation. It also provides 3D object benchmarks and 3D visual odometry dataset. The dataset captures real-world driving scenarios with challenges such as occlusions, partial visibility (e.g., distant or truncated objects), and dynamic interactions between multiple agents. However, the scene flow annotations are limited to *sparse subsets* of LiDAR points due to the labor-intensive nature of manual labeling in non-rigid environments. This sparsity necessitates robust algorithms capable of inferring dense motion fields from incomplete supervision. A subset of 150 driving scenes with sparse but precise scene flow annotations is widely used in scene flow estimation.

- **KITTI Object** (Menze and Geiger 2015) is a subsequent extension of the original LiDAR KITTI benchmark. It provides 200 densely annotated driving scenes captured using a Velodyne HDL-64E LiDAR sensor. The annotations prioritize *road-relevant objects*, with an emphasis on vehicles and vulnerable road users, reflecting real-world safety-critical applications. However, the dataset's focus on sparse, class-specific annotations (versus dense scene flow) limits its utility for tasks requiring full-scene motion fields.

- **StereoKITTI** (Menze et al. 2015 2018) removes 58 scenes from original data (200 training samples and 200 testing samples). It contains 142 point cloud pairs for testing. The ground-truth scene flow is generated via lifting the disparity maps and optical flow to 3D space (Gu et al. 2019).

- **SemanticKITTI** (Behley et al. 2019) is based on the odometry dataset of the KITTI Vision Benchmark (Geiger et al. 2012a) collected in both urban and rural areas. It includes 21 LiDAR se-

quences which are split into eleven (00-10) LiDAR sequences for training and eleven (11-21) for testing. SemanticKITTI encompasses a diverse range of urban scenes captured by Velodyne LiDAR sensors, providing high-resolution point cloud data for 22 sequences. Each sequence offers complex scenarios, including various traffic scenarios, pedestrians, and diverse objects encountered in urban landscapes. The dataset includes semantic labels for 20 object classes, such as cars, pedestrians, cyclists, and vegetation, enabling the training and evaluation of models for semantic segmentation.

- **Lyft** (Kesten et al. 2019) contains 22,680 real-scanned scenes with multi-objects. However, it does not provide any point correspondence and is a partially visible dataset. The term partially visible refers to the lack of complete object geometries in individual scans due to occlusions, single-view LiDAR captures, or missing point correspondences across frames. Specifically, objects are often truncated or observed from limited viewpoints, and the dataset provides no explicit point-level tracking or dense annotations. These limitations restrict its utility to weakly-supervised training.

- **Argoverse** (Chang et al. 2019) is a dataset primarily for autonomous vehicle perception tasks including 3D tracking and motion forecasting. In the spirit of KITTI, a novel format of this dataset, "*Argoverse Scene Flow*" has been created by Pontes et al. (Pontes et al. 2020). The point clouds are collected from two Velodyne VLP-32 sensors. It is noteworthy that the vehicle poses and the 3D object trackings in the original Argoverse 3D Tracking set are utilized to generate pseudo scene flow annotations (Pontes et al. 2020). The whole dataset contains 2,691 training samples and 212 test samples.

- **NuScenes** (Caesar et al. 2020) is a dataset that has recorded diverse data from Boston and Singapore, which consists of tracking

information, map information, and LiDAR point clouds sensed by a Velodyne VLP-32 sensor. It is different from the KITTI dataset collected by the 64-beam Velodyne rotating at 10 Hz. The dataset consists of 20-second clips captured at a frequency of 20 Hz, meticulously selected to showcase a diverse range of driving maneuvers, traffic situations, and unexpected behaviors. Each scene is annotated at 2Hz, which means the bounding boxes are only annotated every 10 frames. nuScenes is the first large-scale dataset to provide data from the entire sensor suite of an autonomous vehicle (6 cameras, 1 LIDAR, 5 RADAR, GPS, IMU). Compared to KITTI (Geiger et al. 2012b), nuScenes includes 7x more object annotations. This difference leads to a discrepancy in data sparsity that yields a distribution shift between KITTI and NuScenes. However, NuScenes does not provide scene flow annotations, which poses a great challenge in deep learning based methods to predict accurate scene flow.

- **Waymo.** The Waymo dataset (Sun et al. 2020) includes a large number of 3D ground truth bounding boxes for LiDAR data and 2D tightly fitting bounding boxes for camera images, all of which are high quality and have been manually annotated. Each scene is a 20-second clip recorded by a 64-beam LiDAR sensor operating at a frequency of 10 Hz. It contains 158,081 training and 39,987 validation frames of point clouds with LiDAR labels (Jin et al. 2022), such as vehicles, pedestrians, signs and cyclists. However, scene flow labels are not included.

2.2 Scene flow estimation methods

As shown in Fig. 2.1, scene flow represents the pointwise motion field of a 3D scene (Vedula et al. 1999). Scene can be represented by depth images and point clouds. Methods based on images extract depth, disparity,

and optical information separately to learn the flow vector. However, image-based methods usually rely on standard variational formulations and energy minimization (Hur and Roth 2020), which yield limited accuracy and suffers from long runtime. The advent of affordable 3D sensors, e.g., LiDARs and RGB-D cameras, simplifies the process of acquiring large-scale 3D point clouds. With the flourishing demand from industry, leveraging point clouds as scene representations is becoming a hotspot in recent years.

2.2.1 The relation between point cloud odometry and scene flow estimation

Point cloud odometry and scene flow estimation are both essential for understanding motion within a 3D environment, but they serve distinct and complementary roles. Point cloud odometry focuses on estimating the overall displacement or movement of the sensor (LiDAR or RGB-D camera) itself, allowing systems to track their own position and orientation over time. This is crucial for tasks such as localization and navigation, where knowing the precise location and trajectory of the sensor (or vehicle) is essential (Chen et al. 2022). One of the primary challenges in LiDAR point cloud odometry is achieving precise scan-to-scan alignment, which necessitates the registration of corresponding points between consecutive point clouds. This registration often relies on nearest-neighbor searches, a computationally intensive task that can become increasingly demanding as the number of points per scan rises.

In contrast, scene flow estimation extends the concept of optical flow from 2D images into 3D space by capturing the motion of each individual point within the scene. Rather than tracking only the sensor's movement, scene flow provides detailed motion information for every visible point, describing how objects and surfaces within the environment move relative to the sensor. This enables a richer analysis of the dynamics within a scene, as scene flow captures pointwise velocity and direction.

The distinction highlights their complementary roles: odometry provides a global view of sensor movement, essential for navigation and localization, while scene flow delivers granular, per-point motion information that supports finer analysis of object interactions and scene dynamics.

2.2.2 Challenges

Thanks to the introduction of large-scale synthetic dataset FlyingThings3D (Mayer et al. 2016) with ground-truth flow annotations, many supervised methods are allowed to learn deep hierarchical features of point clouds and fuse these features to estimate scene flow. This supervised training strategy outperforms traditional registration algorithms, e.g., ICP (Besl and McKay 1992) and shows great potential to be applied in real scenarios. To this end, datasets such as KITTI (Menze and Geiger 2015), NuScenes (Caesar et al. 2020), and Argoverse (Chang et al. 2019) are created, which contain various real scenes.

However, datasets collected by LiDAR do not provide reliable correspondences between consecutive scans. Therefore, a lot of deep learning (deep learning) models have performance gap between synthetic dataset and real dataset. In addition, there are many unexpected occlusions in real scenarios which will affect the overall accuracy. In spite of recent attempts that exploit the advantages of deep learning models, unleashing the full power of deep neural networks on 3D point cloud understanding is still in its infancy. This section categorizes challenges in scene flow estimation into data challenges and deep learning models challenges, which are introduced in the following.

Data challenges

- **Noise.** Point cloud, as one of the most popular format of three dimensional data, is unstructured and noisy. Noise is inevitable from the scanning and reconstruction process. It will hinder the feature extraction and misguide the searching of correspondent points in the neighborhood.

- **Difference in point density.** A LiDAR system identifies the position of the light energy returns from a target to the LiDAR sensor. This inherent attribute of the LiDAR sensor leads to unevenly distributed points underlying a surface. The density decreases dramatically as distance from sensors increases. How to address the diversified point density is still an open problem.

- **Big data challenge.** Scene represented by point clouds contains millions of points. For example, in the Argoverse dataset, each point cloud produced by LiDAR sensor has 107k points at 10 Hz. Such amount of data increase the burden in processing.

- **Diversified motion fields.** Background motion and foreground motion co-exist in a scene. Likewise, large and small motion, close and far objects, rigid and non-rigid objects co-exist in dynamic scenes. The diversity of motion scales poses a great challenge on discriminating different motion fields.

- **Occlusions.** Scene points taken at time t, may be occluded in subsequent time steps. Consequently, a few objects will disappear due to occlusions. The presence of occlusions will significantly influence the flow estimation accuracy.

Challenges from Deep Learning models

- **Generalization ability.** Existing wisdom aims to improve the performance on a specific dataset but fails to generalize to other datasets, especially on the generalization from the simulated to real scenes.

- **Accuracy challenge.** It is impossible to obtain 100% accurate ground-truth scene flow from real scenarios. Due to limited annotations for real scenes, it is challenging to achieve satisfactory accuracy in deep learning algorithms.

- **Efficiency challenge.** Real-time processing ability is imperative for autonomous driving entities. However, the computing power and memory space allocated for processing massive 3D data constructed on vehicles are limited. Currently, efficient deep learning model that can produce real-time large scene perception is still under-explored.

As mentioned before, this thesis aims to balance efficiency and efficacy for scene flow estimation. Therefore, the challenge of generalization ability and accuracy challenge are closely related to this thesis. In the following sections, a comprehensive review on up-to-date compelling deep learning models applied in point cloud-based scene flow estimation approaches is presented from the perspective of supervision. A particular focus is set on analyzing how the state-of-the-art methods deal with challenges in scene flow estimation. These methods are roughly categorized into the following types: supervised, weakly supervised, and self-supervised methods.

2.2.3 Supervised methods

Early methods (Baur et al. 2019, Zou et al. 2019) project the point clouds onto 2D cylindrical maps and apply traditional CNNs to train their flow estimation model. Starting from methods that tackle a large amount of data, a core set of the most innovative work on supervised learning approaches have been identified for scene flow estimation. Many supervised learning approaches rely on ground-truth labels of scene flow. The deep networks are initially trained on synthetic datasets and then fine-tuned on real data.

FlowNet3D. FlowNet3D (Liu et al. 2019a) is the first work that extracts point features from point clouds directly to estimate scene flow. It has three main layers for point cloud processing and uses PointNet++

(Qi et al. 2017b) as its backbone for feature learning. The flow embedding layer aims to aggregate point similarities for scene flow encoding. FlowNet3D (Liu et al. 2019a) finds soft correspondences between point clouds in two consecutive frames. The set up convolutional layer is used for flow refinement. The model has shown good results on synthetic datasets, but has not achieved equivalent performance in real-world settings due to the difficulty of obtaining point-level supervision from real-world data.

HALFNet. Wang *et al.* (Wang et al. 2021b) proposed a hierarchical attention learning network with two different attentions in each flow embedding. Especially, a hierarchical attentive flow refinement module is designed to propagate and refine scene flow estimations layer by layer. HALFNet (Wang et al. 2021b) adopts a more-for-less strategy, which means the number of input points is greater than the number of output points in scene flow estimation. HALFNet has approved its effectiveness in gaining precise structure information of the scene and reducing the consumption of GPU memory. It is also noteworthy that HALFNet uses multiple Euclidean information, which allows the attentive flow embedded in a patch-to-patch manner. Generally, HALFNet demonstrates a better generalization ability of the 3D method than FlowNet3 (Ilg et al. 2018) in 2D metric (e.g., optical flow) and achieves reasonable accuracy compared with existing supervised methods. However, HALFNet does not train on a large real-world dataset, which limits its generalization ability.

FESTA. Previous methods, e.g., FlowNet3D (Liu et al. 2019a) and MeteorNet (Liu et al. 2019b) apply Farthest Point Sampling (FPS) to extract point features. However, FPS usually leads to different downsampled results from two point clouds that represent the same manifold (Wang et al. 2021c). Hence it is intractable to estimate accurate scene flow with the unstable features extracted by FPS. FESTA (Wang et al. 2021c) address this issue via the spatial abstraction with attention (SA^2)

Methods		Highlights	Datasets used
Feature embedding based Methods	Flownet3D⋆ (Liu et al. 2019a)	**Pros:** Pioneer work in using flow embedding layer. **Cons:** Suffer from occlusion and non-uniform data; Unable to maintain local geometric smoothness.	KITTI2015, FLY3D
	Flownet3D++ (Wang et al. 2020)	**Pros:** RGB-D data as input; Capable for non-static scenes; Point-to-plane loss; Geometry-aware; Effective for dynamic reconstruction. **Cons:** Error accumulated when iterating.	KITTI2015, FLY3D
	FESTA⋆ (Wang et al. 2021c)	**Pros:** Point clouds with RGB information as input; Temporal-Spatial attention mechanism; Occlusion aware. **Cons:** Poor generalization ability.	LiDAR KITTI, FLY3D
	HALFlow (Wang et al. 2021b)	**Pros:** More-for-less hierarchical architecture; Double attentive flow embedding; Good practical application ability on real LiDAR odometry task. **Cons:** Complex network structure; Poor efficiency.	StereoKITTI, FLY3D
	HCRF-Flow (Li et al. 2021a)	**Pros:** Point-level and region-level constraints; Good generalization ability. **Cons:** Time-consuming.	StereoKITTI, FLY3D
	Bi-PointFlowNet⋆ (Cheng and Ko 2022)	**Pros:** High accuracy on both occluded version and non-occluded version of FLY3D and KITTI. **Cons**: Struggles with extreme sparsity	KITTI2015, StereoKITTI, FLY3D
	RMS-Flownet(Battrawy et al. 2022)	**Pros:** Hierarchical learning method; Efficient. **Cons:** Limited generalization across sensor types.	StereoKITTI, FLY3D
	WhatMatters⋆ (Wang et al. 2022b)	**Pros:** All-to-all flow embedding layer; Achieved SOTA performance on both synthetic dataset and real dataset. **Cons:** Limitations on occluded scenarios.	StereoKITTI, FLY3D
	FH-Net⋆(Ding et al. 2022)	**Pros:** New data-augmentation strategy; Cross-frame feature enhancement; High inference speed.	KITTI2015, FLY3D, Waymo
Correspondences based Methods	FLOT⋆ (Puy et al. 2020)	**Pros:** Simple and efficient; Addressed transformation challenge. **Cons:** Annotation-hungry; Poor performance on occluded points.	StereoKITTI, FLY3D
	SCTN⋆ (Li et al. 2022a)	**Pros:** Pioneer in using a sparse convolution and transformer to exploit the coherent motions and model point correlations; Spatial feature-aware. **Cons:** Annotation-hungry.	KITTI2018, FLY3D
	PV-RAFT⋆ (Wei et al. 2021)	**Pros:** Pioneer in integrating point and voxel correlations in recurrent all-pairs field to estimate scene flow; GRU-based iterative method. **Cons:** Structure distortion; High time consumption.	KITTI2015, FLY3D
	SAFIT⋆ (Shi and Ma 2022)	**Pros:** Supervised and self-supervised training fashion; Small model size. **Cons:** Annotation-hungry.	KITTI2015, FLY3D, StereoKITTI
Cost volume based Methods	PointPWC-Net⋆ (Wu et al. 2020)	**Pros:** Coarse-to-fine strategy; Supervised and self-supervised training fashion. **Cons:** Some objects are out of view; Error accumulation in the early step.	StereoKITTI, FLY3D
	Res3DSF (Wang et al. 2021a)	**Pros:** Context-aware feature encoding layer and residual flow learning block; Good at learning long-distance motion and discriminating objects with similar pattern. **Cons:** Computation expensive.	KITTI2018, FLY3D
	PointConvFormer⋆ (Wu et al. 2022a)	**Pros:** Feature-based attention module; Improved reweighting mechanism in calculating convolutional weights. **Cons:** Poor performance on occlusions.	StereoKITTI, FLY3D
	Est&Pro⋆ (Wang and Shen 2022)	**Pros:** Occlusion-aware; Uncertainty guided network. **Cons:** The overall performance relies on ground-truth occlusion masks.	KITTI2015, FLY3D
Other Methods	HPLFlowNet⋆ (Gu et al. 2019)	**Pros:** Efficient; Addressed the difference in density challenge and big data challenge. **Cons:** Lack of evaluation on large-scale real dataset: NuScenes.	StereoKITTI, FLY3D
	MoNet (Lu et al. 2022)	**Pros:** Variations of motion across frames are captured; Point cloud prediction with content features; Recurrent neural network; Attention-based motion alignment module. **Cons:** Suffer from accuracy challenge.	Argoverse, LiDAR KITTI

Table 2.2: Summarization of fully supervised deep learning architectures for scene flow estimation. FLY3D is the abbreviation of FlyingThings3D. ⋆ denotes methods with open-sourced code.

layer and the temporal abstraction with attention layer. In the SA^2 layer, FESTA utilizes a trainable Aggregate Pooling module which is based on the shifted position of points by defining the attended regions.

PointPWC-Net. PointPWC-Net (Wu et al. 2020) predicts scene flow via constructing the cost volume at each feature pyramid level. To capture large motions, PointPWC-Net utilizes a coarse-to-fine strategy that concatenates the feature at level L with upsampled feature from level $L + 1$. The scene flows are refined by features generated from the cost volume, the upsampled flow, and the source point clouds. However, PointPWC-Net has some limitations on the KITTI dataset (Menze and Geiger 2015). It is hard to obtain effective correspondences from two consecutive frames due to the strong deformation of local shapes. At last, PointPWC-Net retains the ground points, which may affect the overall performance. PointConvFormer (Wu et al. 2022a) modifies the feature learning mechanism via transformers. It explores the computation of convolutional weights, leveraging the difference in features between points to recalculate the convolutional weights. Additionally, PointConvFormer uses a sigmoid activation for the attention weights that outperformed the use of softmax. These insights resulted in improved performance in experiments compared to traditional Transformer models. PointConvFormer has a 10% improvement of EPE3D on FlyingThings3D dataset than PointPWC-Net.

Res3DSF. Based on the observation that humans are good at perceiving the surrounding dynamic movement, Res3DSF (Wang et al. 2021a) includes a context-aware point feature pyramid module together with a residual flow refinement layer for scene flow estimation. Many previous methods ignored the discrimination of repetitive patterns in dynamic scenes. Res3DSF incorporates the contextual structure learning into their 3D spatial feature extraction layer and learn soft aggregation weights. Res3DSF adopts attentive cost volume to learn flow embeddings from the context-aware feature pyramid module. These flow embeddings

are then refined by the Three-NN interpolation and multiple MLP layers to acquire the final complete scene flow. The evaluation results illustrated in Table 3.1 indicate the effectiveness of the framework proposed by Res3DSF (Wang et al. 2021a). Res3DSF well addresses the diversity of motion fields, so that it can estimate long-distance motion.

FLOT. Several studies in graph matching, such as (Maretic et al. 2019, Nikolentzos et al. 2017), utilize optimal transport to find correspondences between two different graphs. Inspired by these works, FLOT (Puy et al. 2020) casts the task of scene flow estimation as finding soft correspondences on a pair of point clouds via solving an optimal transport problem. FLOT extracts point features through several convolution layers. The transport cost is then measured by cosine similarity of these point features. To circumvent the absence of correspondence on some points, FLOT (Puy et al. 2020) proposes a mass regularisation to ensure that mass is uniformly distributed over all points. A residual network is proposed to improve flow estimation through linear interpolation. FLOT demonstrates the superiority of the algorithm unrolling technique in scene flow estimation. Sinkhorn algorithm (Altschuler et al. 2017) is iteratively applied to update the cost matrix, further enhancing scene flow qualities.

SCTN. Different from FLOT (Puy et al. 2020) which only focuses on sparse 3D coordinates and applies point-based convolutions (Qi et al. 2017b) to learn features, SCTN (Li et al. 2022a) introduces a voxel-based convolution to produce consistent flows in 3D space. SCTN uses a combination of sparse convolution for feature extraction and a transformer module for accurate scene flow prediction. It is the first work to incorporate the transformer with sparse convolution, which allows it to learn relation-based contextual information on point clouds. SCTN uses a correlation matrix to estimate soft correspondences by combining features from both the sparse convolution and the transformer module. Additionally, SCTN proposes a feature-aware spatial consistency loss to improve its ability to distinguish different motion fields.

HCRF-Flow. Rigid and non-rigid motion co-exist in dynamic scenes, which hinders the estimation of accurate scene flow. In this setting, methods that only consider pointwise motion tend to neglect rigid motion in local regions. Therefore, it is indispensable to add constraints on the rigidity of the local transformation in local regions. To this end, HCRF-Flow (Li et al. 2021a) leverages a traditional graphical model: high-order conditional random fields (CRFs) where DNNs and CRFs work collaboratively to achieve pointwise motion regression. In particular, HCRF-Flow proposes a novel position-aware flow estimation module (PAFE) to get the matching cost. PAFE follows the same architecture of FlowNet3D (Liu et al. 2019a), leveraging its core components—set convolutional layers for local feature extraction, flow embedding layers for motion aggregation, and set upconvolutional layers for multi-scale refinement. To enhance spatial reasoning, PAFE integrates positional encoding mechanisms that explicitly encode 3D coordinates into the feature learning process, enabling the model to better capture geometric relationships for robust correspondence estimation. Furthermore, the continuous CRFs ensures the spatial smoothness and the local rigidity of the scene flow predictions. Therefore, rigid motion is well-considered in HCRF-Flow under the constraints of both point-level and region-level consistency.

PV-RAFT. As mentioned before, PointPWC-Net (Wu et al. 2020) utilizes a coarse-to-fine strategy to find point correspondences. However, it suffers from the error accumulation (Wei et al. 2021). PV-RAFT (Wei et al. 2021) is an innovative approach that builds correlation volumes to address limitations of previous cost-volume based methods. It is inspired by the recurrent all-pairs field used in 2D optical flow (Teed and Deng 2020). With voxel correlation features that encodes long-range point clouds, and point-based features that aggregates fine-grained local details, PV-RAFT efficiently captures both short-range and long-range correlations in consecutive point clouds. PV-RAFT utilizes a Gated Recurrent Unit (GRU) to iteratively update the predicted scene flow with

context features as auxiliary information. Besides, PV-RAFT also develops a truncation operation and a refinement module to further increase the accuracy.

HPLFlowNet. HPLFlowNet (Gu et al. 2019) operates on permutohedral lattice points and processes the lattice points by a few Bilateral Convolutional layers (BCL). This strategy improves feature extraction globally and shows better performance. HPLFlowNet directly removes all the occluded points to reduce computational cost. There are three BCL layers in HPLFlowNet, including DownBCL, UpBCL, and CorrBCL. HPLFlowNet also shows great generalization ability to different point densities. It evaluates on 16,384, 32,768, 65,536 points and the network is able to process up to 86K points in one pass.

WhatMatters. WhatMatters (Wang et al. 2022b) follows common practices to compute point features through the set convolutional layer. To capture reliable match candidates from point clouds even in a long distance, WhatMatters proposes a novel pointwise mixture module with backward reliability validation. A comprehensively analysis on point similarity calculation, designs of scene flow predictor, input elements of scene flow predictor, and flow refinement level design showcase what matters in 3D scene flow network.

FH-Net. FH-Net (Ding et al. 2022) deals with multi-scale flows from different layers with a much faster speed. To this end, FH-Net extracts keypoint features via hierarchical Trans-flow layer. The computed sparse flow is then used to obtain hierarchical flows at different resolutions through an inverse Trans-up layer. FH-Net also introduces a new data augmentation strategy to enhance the accuracy of predicted flow, particularly on complex dynamic objects. This work sets new standards for performance on the KITTI and Waymo datasets.

SAFIT. SAFIT (Shi and Ma 2022) introduces the concept of relation reasoning between object-level and point-level relations. The relation module captures relational features between objects, which diversifies the

feature palette of 3D point cloud and can be combined with other features to boost the performance of scene flow. This is different from other methods that only extract geometry or location features for individual objects. As presented in SAFIT, the supervised training scheme outperforms FLOT by 3.8%, 22.58% on preprocessed FlyingThings3D and KITTI dataset (Gu et al. 2019). Besides, SAFIT has 10.90% and 21.82% accuracy improvement over FLOT on FlyingThings3D and KITTI where occluded points are not removed (Liu et al. 2019a).

Dynamic3DSA. To facilatate the analysis of point cloud sequences, four different tasks are integrated into a complete multi-frame 4D scene analysis approach. Huang *et al.*(Huang et al. 2022b) comprehensively study point cloud registration, motion segmentation, instance segmentation, and piece-wise rigid scene flow estimation. To this end, it is necessary to separate individual moving objects from the static background and infer their temporal and spatial properties. Dynamic3DSA accumulates 3D points across multiple frames while representing the scene as a collection of rigid moving agents, followed by the reasoning of motion by agents.

Bi-PointFlowNet. Built upon successful bidirectional learning in time series-based tasks and 2D optical flow estimation, Bi-PointFlowNet (Cheng and Ko 2022) develops the first bidirectional model for 3D scene flow estimation. Bi-PointFlowNet targets at estimating the optimal non-rigid transformation that represents the best alignment from the source to the target frame. Previous standard procedure (i.e., grouping and pooling) usually leads to redundant computations. To address this issue, Bi-PointFlowNet decomposes the MLP weights in bidirectional flow embedding layer into three sub-weights. In this way, the local coordinates, the propogated feature, and the replicated feature of two point clouds can be transformed to produce a new fused feature vector. The following upsampling and warping layer are the same as PointPWC-Net. Compared to PointPWC-Net (Wu et al. 2020), Bi-PointFlowNet reduces the total operation by 44% and accelerates the inference by 33%.

Est&Pro. Est&Pro (Wang and Shen 2022) employs a subnet to predict the occlusion mask, which guides the flow predictor to focus on estimating the motion flows of non-occluded points. In this way, more valid matching costs can be calculated. Est&Pro designs a local-adaptive cost volume, which addresses the dissimilarity in local structure caused by sparse depth sensor (LiDAR) sampling. For occluded points, Est& Pro proposes an uncertainty-truncated propagation network to propagate the flows from nonoccluded points to those occluded points. Intuitively, the flow estimator is responsible forr the non-occluded points, while the flow propagation network focuses on motion flows of the occluded points.

RMS-FlowNet. RMS-FlowNet (Battrawy et al. 2022) employs feature extraction module consists of top-down pathway and bottom-up pathway. From the beginning level, they apply local-feature-aggregation and down-sampling to proceed features at each level. Then utilize up-sampling and transposed convolution to propogate point features. Unlike previous hierarchical structure (Wang et al. 2021b), RMS-FlowNet proposes a patch-to-dilated-patch flow embedding approach, which recomputes features generated from previous steps. This desgin could speed up the model without sacrificing the accuracy. RMS-FlowNet usess a fully supervised loss function similar to PointPWC-Net. This work achieves significant advancements in accelerating the prediction of large-scale, consecutive point clouds (e.g., >250K points), addressing critical efficiency challenges in prior methods. However, despite these advancements, a key limitation arises from its reliance on predefined scale hierarchies. Such fixed scales often inadequately adapt to scenes with variable object sizes (e.g., pedestrians versus trucks) or mixed resolutions, resulting in suboptimal feature aggregation for dynamically varying spatial structures.

2.2.4 Weakly/Self-supervised methods

While promising results have been shown, fully supervised methods rely on the absolute ground truth flow as supervision. Whereas the astro-

Methods		Highlights	Datasets used
Flow embedding based	Just-Go⋆ (Mittal et al. 2020)	**Pros:** Proposed a nearest neighbour loss and a cycle consistency loss; Addressed annotation challenge. **Cons:** Violated the real data distribution; Suffer from accuracy challenge.	FLY3D, NuScenes, LK, KITTI2018
	SFPC (Pontes et al. 2020)	**Pros:** Self supervised learning and non-learning scheme; Applied to point cloud densification and motion segmentation application. **Cons:** Suffer from occlusion challenge and efficiency challenge.	KITTI2015, FLY3D, Argoverse, NuScenes
	Adversarial-SFE (Zuanazzi et al. 2020)	**Pros:** Addressed deep model generalization challenge; Local structures aware. **Cons:** Suffer from occlusions.	KITTI Object, FLY3D, Lyft
	SFGAN (Wang et al. 2022c)	**Pros:** Adversarial learning between the scene flow generator and the point cloud discriminator. **Cons:** Suffer from occlusion challenge.	FLY3D, LK
	OGC⋆ (Song and Yang 2022)	**Pros:** Simultaneous 3D objeccts segmentation and scene flow estimation.	FLY3D, KITTI2015
Correspondences based	Self-Point-Flow⋆ (Li et al. 2021b)	**Pros:** Combined multiple clues (i.e., colors, surface normal); Addressed annotation challenge; Good generalization ability. **Cons:** Suffer from occlusion challenge.	KITTI2015, FLY3D
	Noisy-Pseudo (Li et al. 2022b)	**Pros:** Monocular RGB images and point clouds as data source; Addressed annotation challenge and generalization challenge. **Cons:** Suffer from efficiency challenge.	FLY3D, SK, LK
	Pseudo-LiDAR⋆ (Jiang et al. 2022)	**Pros:** Adapted 2D stereo images to 3D scene flow estimation. **Cons:** Suffer from data noise and accuracy challenge.	FLY3D, SK, NuScenes, Argoverse
	SCOOP⋆ (Lang et al. 2022)	**Pros:** A good balance between error reduction and inference time. **Cons:** Suffer from occlusion challenge; Computationally expensive due to multiple optimization objectives.	FLY3D, KITTI2015
	RC-SFE (Dong et al. 2022)	**Pros:** State-of-the-art weakly supervised; Good generalization ability; Addressed the transformation challenge. **Cons:** Sensitive to the accuracy of background masks; Rely on rigidity assumption; Suffer from occlusions.	SeK, SK, Waymo
	RigidFlow (Li et al. 2022d)	**Pros:** Enhanced local rigidity in scene flow estimation; Good generalization ability. **Cons:** Failed on non-rigid motion; Suffer from occlusions.	SK, FLY3D
	FlowStep3D⋆ (Kittenplon et al. 2021)	**Pros:** Recurrent architecture for non-rigid scene flow; All-to-all correlation learning; Addressed big data challenge and annotation challenge. **Cons:** Manually set iteration parameters; Suffer from occlusion challenge.	SK, FLY3D
	RCP (Gu et al. 2022)	**Pros:** Addressed the difference in sampling data challenge; Simultaneous scene flow estimation and point registration. **Cons:** Suffer from efficiency challenge and occlusion challenge.	FLY3D, SK, ModelNet40 ?
	Rigid3DSF⋆ (Gojcic et al. 2021)	**Pros:** Weakly supervised; Robust to different motion fields and occluded points. **Cons:** Relies on soft correspondence; Lack of the similarity measurement of point spatial features.	SK, SeK, FLY3D
	DCA-SRSFE⋆ (Jin et al. 2022)	**Pros:** Reduced the domain gap between the synthetic dataset and the real dataset; Avoided shape deformations; Addressed the transformation challenge. **Cons:** The predictions on non-rigid objects are not accurate.	GTA-SF, FLY3D, Waymo, Lyft, SK
Correspondences free	SLIM ⋆(Baur et al. 2021a)	**Pros:** Motion-aware; Good generalization to unseen data. **Cons:** The aggregated transform matrix is only suitable for stationary points	FLY3D, NuScenes, CARLA, KITTI2018
	Occlusion-G⋆ (Ouyang and Raviv 2021b)	**Pros:** Occlusion-weighted cost volume structure; Detection on large motion and occlusions. **Cons:** Poor generalization ability.	KITTI2015, FLY3D
	PillarML⋆ (Luo et al. 2021)	**Pros:** Multi-modal data as input; Accurate motion learning; Good generalization ability; Efficient. **Cons:** Multi-resolution features are not aggregated in the pillar motion.	NuScenes

Table 2.3: Summarization of self-supervised/weakly supervised deep learning scene flow estimation methods. ⋆ denotes methods with open-sourced code.

nomical cost of time and money on annotating scene flow for real dataset is expensive. To solve problems from limited annotations, some efforts have been made to relax the expensive labeling burden through exploring weakly-supervised and self-supervised learning strategy. Additionally, several works aim to address the performance gap across different datasets by developing self-supervised architectures. Based on the backbone used by these self-supervised methods, they can be categorized into flow embedding-based, correspondence-based, and correspondence-free approaches. A summary of self-supervised and weakly-supervised deep learning architectures for point cloud-based scene flow estimation is provided in Table. 2.3, where FLY3D refers to FlyingThings3D, LK to LiDAR KITTI, SeK to SemanticKITTI, and SK to StereoKITTI.

Just-Go. Mittal *et al.* (Mittal et al. 2020) utilize nearest neighbor loss and cycle consistency loss based on the framework of FlowNet3D (Liu et al. 2019a). Nearest neighbor loss is formulated as the average Euclidean distance of the transformed point to its nearest neighbor in the second point cloud. So it regularizes the initial flow to be as close as possible to the correct scene flow. Cycle consistency loss is calculated through the absolute Euclidean distance between the transformed point from reverse flow and the original point. The combination of the above two self-supervised losses enables training on large unlabeled autonomous driving datasets that contain sequential point cloud data. However, it ignores the local geometrical properties of point clouds.

Adversarial-SFE. Victor *et al.* (Zuanazzi et al. 2020) proposed a metric learning approach for self-supervised scene flow estimation. Unlike previous self-supervised methods which rely on fine-tuning and finding correspondence in the input data to search for nearest neighbors, Adversarial-SFE. (Zuanazzi et al. 2020) utilizes an adversarially learning loss. Hence Adversarial-SFE does not suffer from the domain shift between synthetic data and real data. Moreover, Adversarial-SFE takes advantage of the permutation invariant nature of the point cloud. It proposes triplet loss

by sampling points together with cycle consistency loss. Adversarial-SFE computes the distance between a pair of point clouds on a latent space. The proposed adversarial metric learning consists of four components:(1) a triplet loss with anchor and positive sampling, (2) a cycle consistency loss, (3) multi-scale triplets for global and local consistency, and (4) adversarial optimization.

SFGAN. 3D point clouds represent the continuous motion of objects in real scenarios. Based on this insight, Wang *et al.* (Wang et al. 2022c) utilize generative adversarial networks (GANs) to learn scene flow. SFGAN (Wang et al. 2022c) presents a novel strategy via discriminating between the generated point clouds and the real point clouds. The predicted scene flow and the source point cloud are incorporated to generate the fake point cloud identical to the target point cloud. Then the discriminator discerns the consistency between the real scene and the synthesized 3D scene (fake point cloud) to enhance the performance of the scene flow generator. SFGAN's adversarial training ensures temporal scene consistency.

Self-Point-Flow. Note that each point not only possesses a spatial position (x, y, z) but also potentially has vectors of attributes, such as normal, color, or material reflection. Self-Point-Flow (Li et al. 2021b) uses global mass constraints with multiple descriptors to formulate one-to-one matching with 3D point coordinate, color, and surface normal as measures. In the optimal transport module, the sum of these three individual costs represents the final transport cost in the entropic regularization term that is solved by the Sinkhorn algorithm. This enables the generation of pseudo labels for real data, which is generated from the assignment matrix. However, conflicting results that exist on local regions will lead to incomplete pseudo-label generation. To address this issue, Self-point-Flow builds a graph through random walk theory that integrates local consistency to refine the pseudo labels. This algorithm is

executed on a fully-connected undirected subgraph and refined with several random walk steps. Then, it propagates to directed subgraph without initial pseudo labels and infers new pseudo labels based on the affinity matrix that describe the nearness between each point in the undirected subgraph (labeled node set) and directed subgraph (unlabeled node set).

FlowStep3D. Inspired by RAFT (Teed and Deng 2020), FlowStep3D (Kittenplon et al. 2021) introduces a recurrent structure to unroll scene flow estimation model with refinement operation. In FlowStep3D, the initial flow vector is estimated by a global correlation matrix, then the rest of the flow sequences are updated based on local correlations in the GRU. FlowStep3D adopts several basic layers, e.g., set convolutional layer, flow embedding layer in Flownet3D (Liu et al. 2019a). Two regularization loss weights are proposed to adjust the regularization. It contributes to the updating of scene flow during iterations.

SFPC. SFPC (Pontes et al. 2020) defines a geometrically interpretable objective function to optimize the scene flow and provides an alternative strategy with learning as self-supervisory signal. Basically, the objective function consists of two different terms. The first term minimizes the 3D distance while the second term is a graph laplacian constraint for keeping the nearby points from shifting too much. To explore the underlying topology connection and context information, SFPC builds an explicit graph on the source point cloud. Compared with recent methods (Wu et al. 2020, Mittal et al. 2020) that group point features in multi-scales, SFPC presents a new clue for estimating scene flow without relying on recursive point features by using an interpretable objective function. SFPC performs well on both synthetic data and real data where the learning strategy shows optimal speed while the non-learning strategy gains better robustness. However, SFPC requires more computation when dealing with larger scale point clouds because a denser point cloud yields more complicated graph connectivity and searching space.

PillarML. Stemmed from the merits of motion representation in bird's eye view (BEV), PillarML (Luo et al. 2021) organizes points into different pillars in vertical order and estimate pillar motion by the velocity residing on each pillar. PillarML (Luo et al. 2021) consists of LiDAR-based structural consistency, probabilistic motion masking, and a cross-sensor motion regularization module. The pillar motion is estimated from unlabeled point clouds paired with 2D images. Statistical observation shows that a self-driving vehicle generates abundant data but only 5% of the data is usable. Therefore, PillarML utilizes multi-sensor as sources of data and exploit free signals from them.

SLIM. SLIM (Baur et al. 2021a) removes the annotation requirement constraint on realistic data by integrating the self-supervised scene flow estimation and the motion segmentation framework. SLIM presents that the motion segmentation signal can be generated by detecting the discrepancy between raw flow predictions and rigid ego-motion. Compared to existing methods (Mittal et al. 2020, Wu et al. 2020), SLIM leverages arbitrary point densities and does not rely on one-to-one correspondences. SLIM is upgraded based on RAFT (Teed and Deng 2020) and evaluated on several real datasets: KITTI2018 (Menze et al. 2018), Nuscenes (Caesar et al. 2020), CARLA (Dosovitskiy et al. 2017), and KITTI-RL (Geiger et al. 2013).

Occlusion-G. A dynamic scene contains multiple different objects that hold their own moving patterns and different 3D object possess specific complicated geometry, hence making it inefficient for scene flow estimation by simply removing occluded regions. The main difficulty of scene flow estimation under occlusion is related to acquiring the exact magnitude of the occlusion. Occlusion-G (Ouyang and Raviv 2021b) aims to estimate 3D scene flow with occlusions in a self-supervised way. It uses a cost volume structure same as PointPWC-Net (Wu et al. 2020), but with added occlusion masking operation where the cost volume of the occluded point is assigned with zero. Besides, Occlusion-G is an

occlusion-weighted mechanism that treats occluded and non-occluded regions separately. Occlusion-G varies from the previous version (Ouyang and Raviv 2021a) in the training stage, where Occlusion-G is free from ground-truth occlusion labels. The idea stemmed from using a synthetic target point cloud to predict occlusion.

Noisy-Pseudo. Noisy-Pseudo (Li et al. 2022b) is a novel multi-modality framework that utilizes both RGB images and point clouds to generate pseudo labels for training scene flow networks. The selection of pseudo labels depends on the geometric information of point clouds. The distance between pseudo labels and their nearest point in the second point cloud tells the reliability of the pseudo label. So that these inaccurate noisy labels are assigned low confidence to reduce the negative effect on network training. To refine the confidence scores of pseudo labels, Noisy-Pseudo updates the confidence score via a local geometry-aware weighted confidence of all the neighboring pseudo labels. Additionally, the combination of both 2D information and 3D information contributes to the self-supervised learning and leads to good performance on both synthetic data and real-world LiDAR data. This method highlights the effectiveness of using multi-sensor data in scene flow estimation.

DCA-SRSFE. Jin *et al.* (Jin et al. 2022) proposed a mean-teacher framework for unsupervised domain adaptation from synthetic data to real data. DCA-SRSFE (Jin et al. 2022) consists of a student model that uses ground-truth scene flow labels for supervision and a teacher model updated as the Exponential Moving Average (EMA) of the student model weights. A deformation regularization module and a correspondence refinement module are introduced to produce high-quality pseudo labels. In the deformation regularization module, a rigid motion between the first point cloud and the warped point cloud is predicted via Kabsch algorithm (Kabsch 1976). This module encourages shape distortion awareness in the student model and promotes adaptive deformations for the

target domain. The flow vector is later improved with surface correspondence by refining local geometry. DCA-SRSFE is supervised by ground truth flow labels in the source domain and trained with a consistency loss over the target domain. The proposed synthetic dataset GTA-SF is a large-scale dataset with real-world labels. According to the experiments, DCA-SRSFE has narrowed down the performance gap between synthetic datasets and real-world scenarios.

RCP. RCP (Gu et al. 2022) decomposes scene flow estimation into two interlaced steps. The first step optimizes 3D flow pointwisely, followed by a recurrent network to optimize 3D flow globally. In the pointwise optimization module, an auxiliary flow vector is calculated by concatenating the point feature and positional encoding. In the second optimization step, RCP leverages GRU to update the hidden state for the estimation of residual flow vectors. RCP is trained in both the fully-supervised manner and the self-supervised manner. RCP also conducts experiments on point cloud registration, where 6-DoF poses are generated by point-to-point costs. The results on scene flow estimation and point cloud registration have achieved on-par performances with state-of-the-art methods.

Ego-motion. Inspired by HPLFlowNet (Gu et al. 2019), Ego-motion (Tishchenko et al. 2020) uses DownBCL and CorrBCL as building blocks to regress relative poses from a pair of point clouds. It estimates non-rigid flow and ego-motion jointly with iterative update module to refine the rigid transformation. Ego-motion also compares performance between fully-supervised, hybrid, and self-supervised training strategy, which shows that hybrid training scheme performs better on FlyingThings3D (Mayer et al. 2016) and KITTI2015 (Menze et al. 2015, Menze and Geiger 2015).

RigidFlow. RigidFlow (Li et al. 2022d) introduces local rigidity prior in self-supervised scene flow learning. Based on the assumption that a scene is composed of several rigid moving parts, RigidFlow decomposes

the source point cloud into a collection of local rigid regions. Different from recent self-supervised works (Baur et al. 2021a, Pontes et al. 2020) that utilize local rigidity as regularization terms, RigidFlow enhances the pseudo label generation module via integrating local rigidity in region-wise scene flow estimation. With a pre-trained predicted flow (Li et al. 2021b), the initial point mapping and rigid transformation are calculated. Then the rigid transformation and pseudo labels for each supervoxel is updated accordingly by solving a least-square problem. This least-square problem aims at calculating rotation matrix and translation vector that aligns independent rigid body from source to target. After several iterations, all of the optimal pseudo rigid scene flow from every supervoxel are combined to form the complete pseudo scene flow.

Pseudo-LiDAR (Jiang et al. 2022). This work can accurately perceives 3D dynamics in 2D images by utilizing a pseudo-LiDAR point cloud as a bridge to compensate for the limmitations of estimating 3D scene flow from LiDAR point clouds. Points that do not contribute to the scene flow preditons are filtered out. In addition, a disparity consistency loss is proposed to boost the self-supervised training.

OGC. OGC (Song and Yang 2022) focuses on making use of inherent object dynamics to assist object segmentation. To extract per-point features and generate object masks, an object segmentation network is first applied to a single point cloud. Then, a self-supervised network is utilized to estimate per-point motions from a pair of point clouds. Due to the challenging moving patterns of different objects, how to fully utilize object dynamics to assist object segmentation becomes more tricky. To tackle this problem, OGC introduces three loss terms to yield effective segmentation supervision. The geometry consistency over dynamic object transformations allows for high-quality masks learning for given flows. Regularization of geometry smoothness ensures that flow vectors in a local area remain consistent with the central point. The geometry

invariance loss drives the estimated object masks to be invariant across different views of point clouds.

MBSE3. MBSE3 (Zhong et al. 2023) leverages SE(3)-equivariant networks to inherently respect 3D rigid transformations, ensuring consistent predictions under arbitrary viewpoint changes. This eliminates the need for explicit data augmentation to handle geometric transformations. It achieves joint rigid object segmentation and motion estimation without labeled data, reducing reliance on costly annotations. However, MBSE3 relies on distinguishable motion patterns between objects and struggles with static scenes or objects with identical motions.

SCOOP. SCOOP (Lang et al. 2022) consists of a self-supervised neural network and an optimization module that work hybrideep learningy to estimate scene flow. In the initialization step of scene flow estimation, SCOOP focuses on extracting point features to obtain soft correspondences, in which cosine similarity is applied to compute matching cost. In the flow refinement step, two optimization functions, basically deployed for reducing the error and increasing the consistency of scene flow field. According to the results, SCOOP reduces errors by over 50% compared to feed-forward models and provides 10 times faster inference time than the Neural Prior work (Li et al. 2021c) relying solely on optimization. Additionally, SCOOP allows for a unique trade-off between time and performance.

Rigid3DSF. To ease the high demand of supervision in scene flow estimation problem, Gojcic *et al.* (Gojcic et al. 2021) proposed a data-driven method that integrates flow into a higher-level scene abstraction represented by multi rigid-body motion. Rigid3DSF (Gojcic et al. 2021) connects pointwise flow with other higher level scene understanding tasks through an object-level deep network. In detail, Rigid3DSF divides the scene into foreground, background, and abstract rigid objects as scene components. As such, scene flow in the background is assigned as ego-motion of sensors and motion prediction in the foreground can be rea-

soned on the level of individual object. To exploit the geometry of the rigid entities, Rigid3DSF introduces an inductive bias. Rigid3DSF also proposes a new test-time optimization to refine the flow predictions. For the training on real dataset under weak supervision, Rigid3DSF uses SemanticKITTI (Behley et al. 2019) without dense scene flow annotations.

RC-SFE. RC-SFE (Dong et al. 2022) is a weakly-supervised scene flow learning framework based on GRU recurrent network. Apart from the source point cloud and the target point cloud, RC-SFE also takes a set of abstraction masks of the source point cloud generated by a pre-trained segmentation network (Gojcic et al. 2021) as input. To convert the initial point correspondences status and pre-warped scene flow, RC-SFE applies Kabsch algorithm (Kabsch 1976) to obtain transformations for each segmented abstractions. So the rigid flow is calculated by the abstraction transformations and abstraction masks. During the updating stage, an GRU-based error awarded optimization is utilized to refine the prediction. Compared to previous work that use indirect constraints into iterative optimization, RC-SFE introduces direct multi-body rigidity constraints to alleviate structure distortion. After several recurrent updates, an optimal mix of scene flow and rigid flow are calculated to form the final hybrid scene flow. However, RC-SFE cannot address the estimation of scene with many non-rigid parts. Same as Rigid3DSF (Gojcic et al. 2021), RC-SFE relies on the segmentation of background to generate accurate estimation. Dealing with non-rigid motions and occlusions is worthy of further exploration in the future.

The contributions of current notable works, highlight the potential for improved accuracy, robustness, and generalization in scene flow estimation. In the following section, this thesis presents related works for another sub-task of dynamic scene understanding: object segmentation.

2.2.5 Hybrid methods

Traditional approaches to scene flow estimation predominantly rely on single-modality data, which introduces inherent limitations depending on the sensor type. These constraints stem from the inability of single-modality systems to capture complementary scene properties (e.g., geometry and appearance and motion). Consequently, recent works like CM-Flow (Ding et al. 2023) advocate for cross-modal fusion, integrating LiDAR, radar, and camera inputs to overcome the trade-offs of unimodal frameworks. Hybrid methods (Teed and Deng 2021, Liu et al. 2022) fuse 3D point clouds (geometry) with 2D images/videos (appearance) to harness complementary information, demonstrating superior robustness, occlusion handeep learninging, and accuracy in motion estimation. This multimodal paradigm has emerged as a key research focus in recent years.
CM-Flow. CM-Flow (Ding et al. 2023) introduces a cross-modal learning framework for 4D radar-based scene flow estimation by leveraging co-located sensors on autonomous vehicles. The core innovation lies in its use of odometer, LiDAR, and camera to generate pseudo-supervision signals, enabling training without human-annotated labels. This approach frames scene flow estimation as a multi-task learning problem, where modality-specific tasks (e.g., optical flow estimation) provide auxiliary supervision. However, two critical limitations arise: Noisy Supervision: Signals from individual modalities (e.g., optical flow) inherently contain higher noise compared to human annotations. Sparse Constraints: Pseudo-labels for scene/optical flow only apply to detected moving points, which are vastly outnumbered by static points, thereby limiting their impact on overall model performance.

To mitigate these issues, CM-Flow opportunistically fuses cross-modal supervision from three sensors commonly co-deployed with 4D radar: Odometer (GPS/INS) for ego-motion compensation, LiDAR for sparse 3D geometric priors, RGB Camera for dense texture-based correspon-

dence.

RAFT-3D (Teed and Deng 2021) extends the RAFT optical flow framework by integrating monocular depth estimation with point cloud motion through a 2D-3D correlation volume, achieving state-of-the-art (SOTA) performance on the KITTI dataset.

2.3 Point cloud segmentation

Point cloud object segmentation is the task of identifying and isolating individual objects within 3D point cloud data. By separating distinct objects, such as cars, pedestrians, or other dynamic elements, segmentation lays the groundwork for understanding and analyzing object motion in real-world environments. In this section, a brief review of point cloud object segmentation techniques is introduced, covering approaches from static inputs to dynamic sequences.

2.3.1 Segmentation on Static point clouds

Before the advent of deep learning methods, point cloud segmentation relied heavily on optimization techniques and hand-crafted features. Methods often involved fitting geometric models to the point clouds or using graph-based approaches with optimization algorithms such as the Expectation-Maximization (EM) algorithm, RANSAC (Random Sample Consensus), and Markov Random Fields (MRFs) to segment and classify regions. These techniques aimed to optimize a cost function that encapsulated prior knowledge about the structure of the scene or the properties of the objects to be segmented. The introduction of deep learning, particularly PointNet and its variants, has since revolutionized the field by directly learning feature representations from raw point clouds, reducing the need for explicit optimization of such models as the primary strategy.

Recent research has focused on enhancing the interpretability and generalization of point cloud segmentation models. Techniques such as

attention mechanisms (Zhao et al. 2021, Mazur and Lempitsky 2021) and graph neural networks (Wang et al. 2019) are being integrated to better capture local structures and relationships among points, which can be crucial for understanding complex scenes in semantic level and instance level. Point Transformer (Zhao et al. 2021) employs self-attention to capture long-range dependencies among points, enhancing the model's ability to focus on relevant features while ignoring noise. This has proven particularly beneficial in complex environments where point clouds may contain significant variations. Additionally, emerging methods are combining multiple modalities, such as RGB images and point cloud data (Krispel et al. 2020, Lu et al. 2023). This integration helps create richer representations, enhancing the overall quality of segmentation. By leveraging information from different sources, these approaches improve accuracy and robustness in complex environments.

2.3.2 Segmentation on Dynamic point clouds

Dynamic point cloud segmentation is a challenging task due to the unique characteristics of point cloud data, which lacks a consistent structure and exhibits high variability in object appearances across frames. A notable approach, P4Transformer (Fan et al. 2021), introduces a novel deep learning model specifically designed to handeep learninge raw point cloud videos without relying on point tracking. This is achieved through a 4D convolution layer that captures local spatio-temporal patterns, alongside a self-attention transformer module that models long-term dependencies across frames. By focusing on self-attention mechanisms rather than explicit tracking, P4Transformer effectively captures appearance and motion information in the video, demonstrating success in 3D action recognition and 4D semantic segmentation tasks.

In contrast, (Lin et al. 2018) proposes a hierarchical segmentation method for RGBD data, which leverages the inherent 3D geometry captured by depth sensors. This method operates on low-level geometric

features like connectivity and compactness to form a hierarchical representation of objects that is propagated through time, maintaining temporal coherence by managing object connectivity, splits, and merges. This geometry-based method bypasses the need for large annotated datasets, making it suitable for generic scenes and settings with limited data. While P4Transformer excels in high-accuracy scenarios given ample data and computational resources, the RGBD-based method (Lin et al. 2018) offers a more efficient alternative for applications where depth information is available but annotated data is scarce. Together, these two approaches illustrate complementary strategies in dynamic point cloud segmentation: a data-driven, transformer-based model that excels with extensive data and compute resources, and a geometry-based, data-efficient model suitable for real-time applications and resource-limited environments.

In recent years, several works have explored segmentation of moving objects directly from point cloud sequences, independent of video data. These methods leverage the spatial and temporal continuity of point clouds captured over time, such as those generated by LiDAR sensors, to segment dynamic elements in scenes. A common approach involves tracking point clusters frame by frame, using point-based motion information to identify and isolate moving objects or moving rigid parts. For instance, some methods compute pointwise or cluster-wise trajectories across frames (Shi et al. 2021), detecting anomalies or deviations from static backgrounds to segment dynamic objects. Other approaches (Fan et al. 2022, Yin et al. 2021) employ spatio-temporal graph networks, where point clouds are represented as graphs and temporal edges capture the movement of points. These graph-based methods can capture complex motion patterns and relationships among points across frames, providing a robust basis for segmenting moving objects even in cluttered environments. Another set of works focuses on deep learning models,

utilizing recurrent neural networks (RNNs) (Shi et al. 2020) or 3D convolutional neural networks (3D CNNs) (Li et al. 2023) to process sequential point clouds and learn temporal features that help distinguish between static and moving objects. These methods demonstrate the effectiveness of using point cloud sequences alone, leveraging the inherent 3D information and temporal coherence of point clouds to achieve reliable segmentation of dynamic objects without relying on RGB video data.

Despite their impressive network architectures, current static and dynamic point cloud segmentation models typically adhere to a scene-wise training protocol. These approach treats each point cloud as an individual training sample, aggregating all classification errors within each scene for the optimization of network parameters. Consequently, these models overlook the rich relationships between points across different scenes, failing to regularize the feature embedding space from a holistic perspective. In this thesis, this limitation is addressed by integrating scene flow with object segmentation. In detail, this thesis aims to propose an unsupervised approach for segmentation on dynamic point clouds. Moreover, a multi-frame unsupervised learning framework is studied to produce accurate and robust object segmentations. The combination of scene flow and object segments enables systems not only to perceive changes across the entire scene but also to capture movement patterns of individual objects.

2.3.3 Evaluation Metrics for Object Segmentation

This section illustrates eight metrics in evaluating object segmentation performance.

2.3.3.1 Panoptic Quality

The Panoptic Quality (PQ) metric is designed to provide a comprehensive evaluation of panoptic segmentation results. It combines aspects of both semantic and instance segmentation evaluation into a single metric. PQ

is defined as follows:

$$\mathrm{PQ} = \frac{\sum_{(p,g)\in \mathrm{TP}} \mathrm{IoU}(p,g)}{|\mathrm{TP}| + \frac{1}{2}|\mathrm{FP}| + \frac{1}{2}|\mathrm{FN}|} \tag{2.1}$$

where: TP (True Positives) is the set of correctly matched pairs of predicted segments and ground truth segments and FP (False Positives) is the set of predicted segments that do not match any ground truth segment. FN (False Negatives) represents the set of ground truth segments that do not match any predicted segment. IoU(p,g) is the Intersection over Union between a predicted segment p and a ground truth segment g. The total number of segments involved in evaluation. This includes the count of true positives |TP| and false positives |FP|, and false negatives |FN|coefficients for FP and FN penalize the metric for each error, balancing the trade-off between precision and recall. High PQ indicates that the algorithm performs well in both correctly segmenting and accurately classifying instances while a low PQ suggests that the algorithm either misses many segments, produces many false segments, or the predicted segments do not match well with the ground truth segments.

2.3.3.2 UQ

Unknown quality, namely UQ is a recall-based metric that measures the performance on annotated instances only. It is defined as:

$$\mathrm{UQ} = \underbrace{\frac{\sum_{(p,g)\in \mathrm{TP}} \mathrm{IoU}(p,g)}{|\mathrm{TP}|}}_{\text{segmentation quality (SQ)}} \times \underbrace{\frac{|\mathrm{TP}|}{|\mathrm{TP}| + |\mathrm{FN}|}}_{\text{recall quality (RQ)}} \tag{2.2}$$

2.3.3.3 Precision

A predicted segment is considered as a match (True Positive) if its IoU with a ground truth segment exceeds 0.5. The ratio of true positive segments to the sum of true positive and false positive segments is represented by Pre:

$$\mathrm{Pre} = \frac{TP}{TP + FP} \tag{2.3}$$

2.3.3.4 Average Precision (AP)

AP score is a widely used evaluation metric in object segmentation task (Lin et al. 2014, Song and Yang 2022). AP is the average of the interpolated precision values at the 101 recall points. In object segmentation tasks, AP is a crucial metric because it balances both precision and recall, providing a single measure of the model's performance. High AP indicates that the model performs well in correctly identifying and accurately segmenting objects while minimizing false positives and false negatives.

2.3.3.5 Recall

Recall is the ratio of true positive segments to the sum of true positive and false negative segments (Powers 2020).

$$Recall = \frac{TP}{TP + FN} \tag{2.4}$$

2.3.3.6 Mean Intersection over Union

For each correctly matched pair of predicted and ground truth segments, the IoU is computed as the area of their intersection divided by the area of their union. The IoU is a measure of the overlap between two regions. A predicted segment and a ground truth segment are considered a match if their IoU exceeds a threshold, typically set to 0.5 (Everingham et al. 2010).

2.3.3.7 F1-score

F1 is the harmonic mean of Precision and Recall:

$$F1 = 2 \cdot \frac{P \cdot R}{P + R} \tag{2.5}$$

2.3.3.8 Rand Index

Rand index, also known as rand statistic, ranging from 0 to 1, it counts the number of agreements and disagreements between two clusterings. Given a set of n elements, let: $\mathcal{C}$ be the set of clusters in the ground

truth, $\mathcal{C}'$ be the set of clusters produced by the clustering algorithm. The Rand Index is computed as follows:

- **A**: The number of pairs of elements that are in the same cluster in both $\mathcal{C}$ and $\mathcal{C}'$.
- **B**: The number of pairs of elements that are in different clusters in both $\mathcal{C}$ and $\mathcal{C}'$.
- **C**: The number of pairs of elements that are in the same cluster in $\mathcal{C}'$ but in different clusters in $\mathcal{C}$.
- **D**: The number of pairs of elements that are in different clusters in $\mathcal{C}'$ but in the same cluster in $\mathcal{C}$.

The total count of A+B+C+D is denoted as N, and the Rand Index is calculated as (A+B)/N.

This chapter provides a comprehensive review of methodologies for advancing scene flow estimation and 3D object segmentation in point cloud data. It systematically contrasts current approaches for estimating scene flow based on learning strategies. Additionally, the chapter explores methods for 3D object segmentation, including strategies that work without extensive labeled datasets or strong supervision. In conclusion, the landscape of dynamic scene understanding is rapideep learningy advancing, driven by innovative methodologies and a growing understanding of the complexities involved in 3D dynamic analysis. By highlighting recent innovations in both scene flow estimation and unsupervised segmentation, this thesis aims to enhance dynamic scene understanding via deep learning method.

3

Estimating 3D Scene Flow via Grouped Attention and Global Motion Aggregation

This chapter introduces a novel scene flow estimation method: **GAMAFlow**. To ensure efficiency and efficacy, a point Transformer model with grouped attention mechanism is proposed, which can be effectively used for point feature extraction. Moreover, the necessary building blocks and network layers are explained. Next, a new operation: global motion aggregation is introduced to enhance the local motion feature with global-aware context feature. In summary, this chapter has been previously published (with some amendments) as, Zhiqi Li, et al. GAMAFLOW: Estimating 3D Scene Flow via Grouped Attention and Global Motion Aggregation.

3.1 Motivation

Scene flow estimation was proposed by Vedula et al. (Vedula et al. 1999), where scene flow is defined as 3D motion vector and predicted through optimization process. It is a crucial primitive to various visual perception and understanding tasks like motion segmentation (Baur et al. 2021b, Ay-

gun et al. 2021), object tracking (Zhai et al. 2020), and motion prediction (Wang et al. 2022a, Chen et al. 2021b). Many scene flow estimation methods (Shi and Ma 2022, Liu et al. 2019a, Kittenplon et al. 2021, Wei et al. 2021, Li et al. 2022c) have been proposed to predict 3D scene flow between two consecutive point cloud frames directly through deep neural networks. The first pioneer work is FlowNet3D (Liu et al. 2019a), which is built based on PointNet++ (Qi et al. 2017b) layers. FLOT (Puy et al. 2020) utilizes optimal transport to find soft correspondences between a pair of point clouds.

Many recent methods have emphasized feature embedding to aggregate local features across multiple scales (Wu et al. 2020, Wang et al. 2021b a, Ding et al. 2022). Techniques such as coarse-to-fine architectures and hierarchical feature learning have demonstrated the strength of deep neural networks for scene flow estimation by performing multi-stage refinements at various scales (Kittenplon et al. 2021) or single resolutions (Wang et al. 2022d). These pipelines generally predict flow at each stage by regressing scene flow from local neighborhoods using convolutions (Wu et al. 2022a, Cheng and Ko 2022). However, such approaches face challenges with large displacements, often requiring multistage refinements to incrementally estimate extensive motion.

This leads to a critical insight: high-quality point features are essential for the success of supervised 3D scene flow estimation. Our analysis indicates that these methods generally achieve optimal performance only after several rounds of flow refinement. Consequently, the question arises: *how to generate higher-quality point features efficiently?* This question is central to advancing the robustness and accuracy of these methods, as high-quality point features enable better local neighborhood understanding and finer motion detail extraction, thereby reducing the dependency on multiple refinement stages. Additionally, achieving rich point feature representations would help mitigate the challenges posed by large dis-

placements, allowing models to better generalize across varied dynamic scenes.

Building on the aforementioned advancements in point feature learning and scene flow estimation, recent work has extended attention-based models to scene flow estimation. Networks that employ spatio-temporal attention models further improves the performance on non-uniform point clouds (Wang et al. 2021c a). SAFIT (Shi and Ma 2022) proposes a segmentation-aware approach that aggregate the features of all points with both self-attention and cross-attention. RPPformer-Flow (Li et al. 2022c) is an approach embedded with transformer layer at all stage of estimating flow vector. Despite the effectiveness shown by these transformer-based approaches, most have overlooked the potential of transformers to capture long-range dependencies, primarily due to the high quadratic computational cost associated with the number of input points. This limitation has restricted the full capacity of transformers to model global interactions, particularly in large and complex scenes.

With this in mind, this chapter aims to balance efficacy (stable flow estimation) and efficiency (fast-feature aggregation), motivating the design of an improved Transformer architecture that captures both local and global information at a modest computational cost. The design principle is inspired by advances in Transformer-based methods for 3D tasks, as demonstrated in prior work (Wu et al. 2022b).

Given the current state-of-the-art in scene flow estimation, it is observed that prediction errors primarily stem from occlusions and invisible long-distance objects, which pose significant challenges in discriminating multi-scaled motion fields. From prior work, this thesis has the following observations: (1) Voxel-based representations can efficiently encode multi-scale features of 3D point clouds, which are then used for object detection or segmentation. However, the downside of voxel-based representation is that it degrades localization quality due to the

coarse voxelization. (2) Point-based representation could preserve accurate point positions with flexible receptive fields, which benefits flow estimation without heavy computation overhead. In light of these, the recent work PV-RAFT (Wei et al. 2021) integrates the voxel-based and point-based feature learning strategy. Meanwhile, SCTN (Li et al. 2022a) combines the point feature extracted through Transformers with voxel feature extracted via sparse 3D convolution. Despite yielding promising results, this integration of voxel-based and point-based feature representations poses two problems. The voxel-to-point encoding through voxel set abstraction operations introduces significant computational overhead, which is further exacerbated by the multi-stage point feature abstraction. On the other hand, the pooling operation in the voxel branch fails to fully harness the valuable dense points, resulting in little performance improvement for faraway or small objects with sparse points.

To address these challenges, this chapter introduces **GAMAFlow**, a Transformer-based scene flow estimation model that integrates point-voxel correlations. First, the model augments point features using a point Transformer layer, where grouped vector attention is employed to propagate pointwise features and guide the learning of discriminative patterns at the voxel level. Second, a context-aware global motion aggregation (GMA) module is proposed to enhance local motion features by aggregating global contextual cues. Extensive experiments demonstrate the superiority of the proposed method—**GAMAFlow**—on both synthetic and real-world datasets, outperforming existing approaches in accuracy and robustness.

3.2 Methods

Overview: The whole pipeline of the proposed method is depicted in Fig. 3.1. It takes two consecutive point clouds $\mathcal{X} \in \mathbb{R}^{N\times 3}$ and

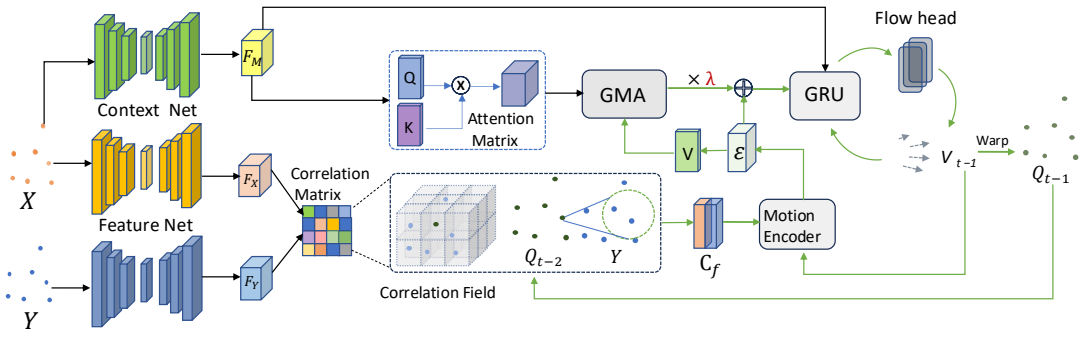


Figure 3.1: The pipeline of **GAMAFlow**: The input comprises two point clouds $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^3$ with 3D positions. The correlation field is constructed from point-level features computed by feature net and voxel-level features. These fine-grained correlations C_f, combined with the current flow estimate V_{t-1}, are processed by a motion encoder to produce a motion feature $\mathcal{E}$. GMA integrates context feature F_M and local motion feature $\mathcal{E}$ to refine motion representation. The GRU iteratively updates its hidden state using the concatenated context feature, local and global motion features. A flow head predicts the residual flow V_{t-1}. This refines the warped point cloud $Q_{t-1} = \mathcal{X} + V_{t-1}$ for subsequent iterations.

$\mathcal{Y} \in \mathbb{R}^{M\times 3}$ as input. **GAMAFlow** aims to predict a set of flow vectors $V = \{v_i \in \mathbb{R}^3\}_{i=1}^{N}$ that describe the motion field, which means point x_i in the source point cloud $\mathcal{X}$ is expected to move to $y_i = (x_i + v_i) \in \mathcal{Y}$. The proposed model initiates with the extraction of pointwise feature representations $F_\mathcal{X}$, $F_\mathcal{Y}$ from the source point cloud $\mathcal{X}$ and target point cloud $\mathcal{Y}$, respectively, leveraging a geometric-aware feature encoding framework (Sec 3.2.2). The context net shares same structure of feature net. Subsequently, a dense correlation lookup table is constructed by computing feature affinity scores between all pairs of points in $\mathcal{X}$ and $\mathcal{Y}$. This correlation field establishes preliminary correspondences for motion estimation (Sec 3.2.3). Following this, a global motion aggregator module is employed to refine the initial correlations by integrating contextual and structural dependencies across the point clouds, thereby enhancing the discriminative capacity of motion-related features (Sec 3.2.4). These enriched features are then processed through a Gated Recurrent Unit (GRU)-based recurrent network, which iteratively refines the scene flow

prediction through temporal feature propagation. At each iteration t, the network generates an incremental flow vector ΔV_t which updates the current flow estimate V_t as $V_t = V_{t-1} + \Delta V_{t-1}$ (Sec 3.2.5). Concurrently, the source point cloud $\mathcal{X}$ is progressively warped to an intermediate representation $Q_{t-1} = \mathcal{X} + V_{t-1}$, aligning it geometrically with the target $\mathcal{Y}$ at each step. This iterative alignment enables the model to resolve ambiguities in large displacements through coarse-to-fine refinement. After T iterations, the final flow estimate V_T represents the optimized displacement field that optimally deforms $\mathcal{X}$ to match the structural topology of $\mathcal{Y}$.

3.2.1 PointTransformer Layer

Before detailing the proposed method, a brief review of transformers is presented. PointTransformer (Zhao et al. 2021) employs self-attention during feature learning, which indeed brings efficiency down when the network becomes deeper and weight parameters increase. The attention mechanism introduced in PointTransformer is as follows:

$$\begin{aligned} &\mathbf{Q} = \mathbf{W}_q F_i, \quad \mathbf{K} = \mathbf{W}_k F_j, \quad \mathbf{V} = \mathbf{W}_v F_j, \\ &\mathbf{A}_{ij} = \mathrm{SoftMax}\left(\mathcal{M}\left(\mathbf{Q} - \mathbf{K} + \mathbf{P}_{emb}\right)\right), \\ &F_i' = \textstyle\sum_{F_j \in \mathbb{G}(i)} \mathbf{A}_{ij} \odot \left(\mathbf{V} + \mathbf{P}_{emb}\right). \end{aligned} \tag{3.1}$$

Let $\mathcal{F} = \{F_i\}_i$ be a set of feature vectors. F_i' is the output feature. $\mathbf{W}_q, \mathbf{W}_k$, and $\mathbf{W}_v$ are pointwise transformations. Here $\mathbb{G}(i) \in \mathcal{F}$ is k nearest neighbors of F_i. The attention weights $\mathbf{A}_{ij}$ in the above equation are the scalars computed by the scaled dot product of the query and the key elements (Wu et al. 2022b). $\mathcal{M}$ denotes a mapping function such as multilayer perceptron (MLP) that computes the attention vectors to reweight the value vector $\mathbf{V}$ before aggregation. $\mathbf{P}_{emb}$ is postional embedding.

As shown in Eq. 3.1, transformers and their variants rely on the encoded representation of input features. PointTransformer (Zhao et al. 2021) justifies the use of local attention by stating that computing global

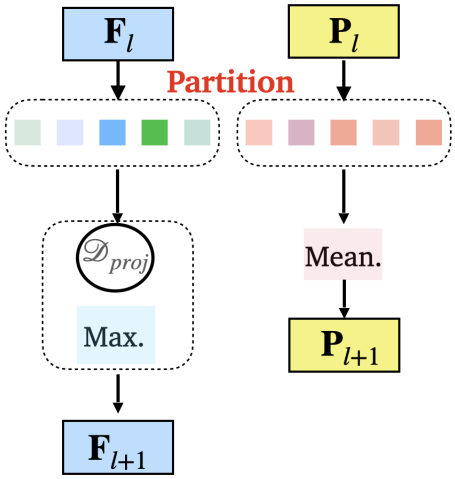


Figure 3.2: The detailed design of pooling module in point Transformer layer. The original point set is separated into non-overlapping partitions. Maxpooling is applied on the point-level feature. For each partition, the maximum feature value is selected among all points in the group. Meanpooling computes the average position of all points in the group.

attention on all input points is almost infeasible, especially when dealing with large-scale data, which indicates that the computational complexity of performing global attention on the entire point cloud is prohibitive.

Remarks: Existing attention blocks applied on 3D point cloud mainly follow two streams: scalar attention and vector attention, where the point features are projected by linear layers or MLPs to generate query, key, and value vectors. In contrast to scalar attention, *vector attention* introduces a weight encoding function to modulate the interaction between the query and key vectors. However, the parameters in weight encoding unit of grouped vector attention have been reduced, leading to a more powerful and efficient model.

3.2.2 Point Feature Extraction via Grouped Attention

Previous attention-based approaches (Li et al. 2022a, Shi and Ma 2022) compute features directly on the whole point cloud, ignoring the existence of multi scale motion fields in the scene. To address this issue, the

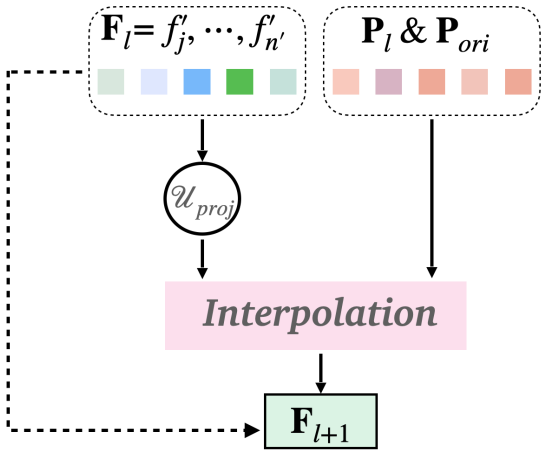


Figure 3.3: The detailed design of unpooling module in point Transformer layer. The widely adopted interpolation-based unpooling method can be extended to partition-based unpooling module, preserving the structural integrity of point features while maintaining computational efficiency.

feature and context networks are constructed using PointTransformerV2 (Wu et al. 2022b), enabling multi-scale point feature learning. The feature network is designed to generate 128-dimensional per-point descriptors for input point clouds, denoted as $F_X \in \mathbb{R}^{N\times D}, F_Y \in \mathbb{R}^{M\times D}$ in Fig. 3.1. The feature network architecture comprises four Point Transformer blocks operating at distinct resolutions. Each block begins with a downsampling layer, followed by grouped attention for localized feature aggregation. Subsequently, features are decoded and upsampled to restore the original point cloud resolution. To achieve this, partition-based pooling is employed to divide the point cloud into L non-overlapping subsets. Each subset is defined as $\mathcal{S}_i = (\mathcal{P}_i, \mathcal{F}_i)$, where $x_i = (p_i, f_i)$ belongs to $\mathcal{S}_i$. Feature f_i is updated via maxpooling operation and point position p_i is updated via meanpooling. Partition-based pooling is illustrated in Fig. 3.2 and grid uncoupling is illustrated in Fig. 3.3. Partition-based pooling performs separation on point clouds without overlapping. Given a point set $\mathcal{S} = (\mathcal{P}, \mathcal{F})$ from L subsets $[\mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_L]$, the point feature

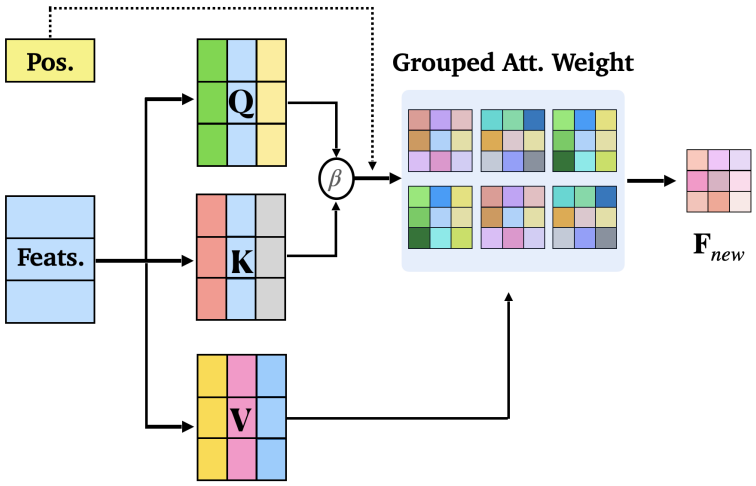


Figure 3.4: PointTransformerV2 attention module.

is updated via

$$\boldsymbol{f}_i' = \text{MaxPool}\left(\left\{\boldsymbol{f}_j \boldsymbol{U} \mid \boldsymbol{f}_j \in \mathcal{F}_i\right\}\right), \quad \boldsymbol{p}_i' = \text{MeanPool}\left(\left\{\boldsymbol{p}_j \mid \boldsymbol{p}_j \in \mathcal{P}_i\right\}\right). \tag{3.2}$$

The pointwise feature vector will go through a linear projection $\mathbf{U}$ before the max pooling operation. Collection of the updated position $\boldsymbol{p}_i'$ and point feature $\boldsymbol{f}_i'$ gives the contents for the next stage encoding. The unpooling operation follows a common practice by interpolation of the k = 3 nearest neighbors based on an inverse distance weighted average. The point feature in higher resolution are obtained by mapping point feature to all points from the same subset $\mathcal{S}_j$, which is given by

$$\boldsymbol{f}_i^{up} = \boldsymbol{f}_j', \quad \text{if}(\boldsymbol{p}_i, \boldsymbol{f}_i) \in \mathcal{S}_j. \tag{3.3}$$

Grouped Vector Attention (Fig. 3.4) efficiently learns spatial features across diverse regions of point clouds through a partitioned attention mechanism. Formally, for a point x_i, the attention weights and aggregated features are computed as:

$$\mathcal{A}_{ij} = \beta(\gamma(\mathbf{q}_i, \mathbf{k}_j)), \; f_i^a = \sum_{x_j}^{\mathcal{S}(p_i)} \sum_{l=1}^{g} \sum_{m=1}^{c/g} Softmax(\mathcal{A}_i)_{jl} \, v_j^{lc/g+m}, \tag{3.4}$$

where γ denotes a relational function encoding geometric or semantic dependencies, and $\beta(\cdot)$ generates grouped attention weights $\mathcal{A}_{ij}$. By di-

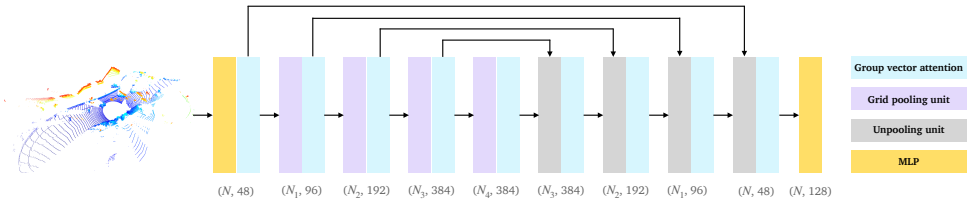


Figure 3.5: The structural components of Point Transformer v2, showcasing its multi-scale architecture designed to handle varying number: N4, N3, N2, N1, and N points. The model incorporates group vector attention to process point cloud data at different scales, utilizing grid pooling units for dimensionality reduction and unpooling units for upscaling. Additionally, it employs multi-layer perceptrons (MLP) for complex feature interactions. The sequence of these components facilitates efficient feature extraction and transformation across different scales within the point cloud.

viding the channels of the value vector into g groups, the number of parameters required for the aggregation of features is reduced by a factor of g. The aggregation operates locally over the reference set $\mathcal{S}(p_i)$, typically defined as the $k-$nearest neighbors of x_i (size $k \ll N$), avoiding global $O(N^2)$ complexity. Moreover, computing g independent Softmax operations on c/g channels rather than one over c channels further reduces computational overhead.

The full feature extraction network is illustrated in Fig. 3.5. The network input consists of raw 3D point coordinates, which also serve as the initialized feature vector. Tuples below each stage block denote the sampled point count and feature dimensions for each encoding layer. The downsampled resolutions N_1, N_2, N_3, N_4 are determined by grid sizes. In the experiments, the grid sizes for the four stages are set to [0.06, 0.12, 0.24, 0.48]. The local attention group sizes are configured as [12,24,48,64] for the encoder and [6,12,24,48] for the decoder.

Context Net: A context feature $\mathcal{F}_M \in \mathbb{R}^{N \times D}$ is extracted to enrich context information (e.g., category cues) during the flow estimation process. The context net shares the same structure as the feature net, without weight sharing.

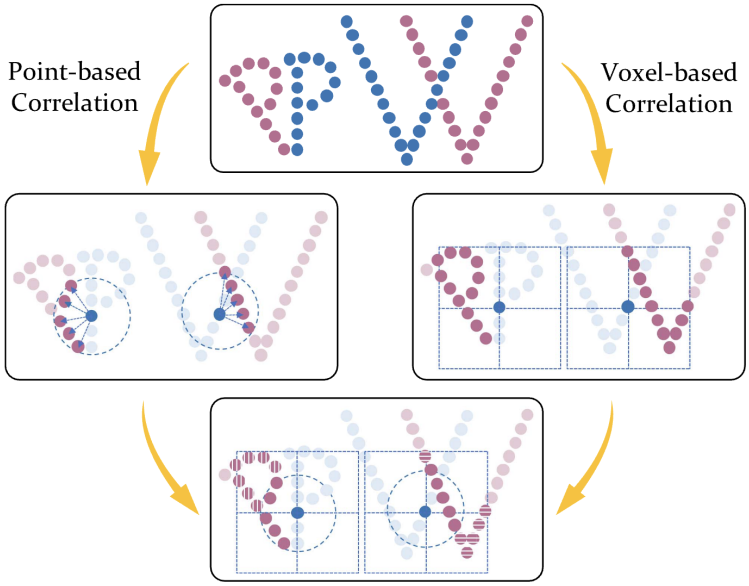


Figure 3.6: The illustration of correlation field, figure from (Wang et al. 2023). For a point in the source point cloud (blue), its k-nearest neighbors in the target point cloud (magenta) are identified to establish point-based correlations. Long-range interactions are further modeled by constructing voxel structures centered on the source point.

3.2.3 Point voxel correlation field

PV-RAFT (Wei et al. 2021) introduces point-voxel correlation fields based on feature similarities. Following this insight, GAMAFlow constructs two correlation volumes from the feature net introduced in Sec. 3.2.2. The correlation volume integrates features at **point-level** and **voxel-level**, as shown in Fig. 3.6. Let $\mathcal{N}_k = \mathcal{N}(\mathcal{Q}_t)_k$ represents the top-k nearest neighbors of $\mathcal{Q}_t$ in $\mathcal{Y}$.

Point-level correlation feature between $\mathcal{Q}_t$ and $\mathcal{Y}$ is defined as

$$\mathbf{C}_p\left(\mathcal{Q}_t, \mathcal{Y}\right) = \mathbf{\Gamma}\left(\mathrm{MLP}(\mathrm{concat}(\mathbf{C}_M(\mathcal{N}_k, \mathcal{Q}_t), \mathcal{N}_k - \mathcal{Q}_t))\right). \tag{3.5}$$

In Eq. 3.5, $\mathbf{C}_M \in \mathbb{R}^{N \times M}$ is the correlation value between Q_t and $\mathcal{N}_k$, which has been truncated to save memory. concat denotes concatenation of correlation and spatial information. $\mathbf{\Gamma}$ is a max pooling operation on k dimension.

Voxel-level correlation feature is defined as

$$\mathbf{C}_v\left(\mathcal{Q}_t, \mathcal{Y}\right) = \mathrm{MLP}\left(\underset{\mathbf{i}}{\mathrm{concat}}\left(\frac{1}{n_{\mathbf{i}}}\sum_{n_{\mathbf{i}}}\mathbf{C}_M\left(\mathcal{N}_r^{(\mathbf{i})}\right)\right)\right), \tag{3.6}$$

where n_i denotes the number of points in $\mathcal{Y}$ that located in a sub-cube of $\mathcal{Q}_t$ and $\mathcal{N}_r^{(\mathbf{i})}$ indexes all neighbor points of a sub-cube in $\mathcal{Q}_t$. In Eq. 3.5 and Eq. 3.6, $\mathbf{C}_M(\mathcal{N}_k)$ represents the corresponding truncated correlation values, which is computed through the pairwise dot-product between feature vectors $\mathbf{C}_M = \mathbf{F}_q^t \cdot \mathbf{F}_y$. The combination of $\mathbf{C}_v$ and $\mathbf{C}_p$ is denoted as $\mathbf{C_f}$. In the current paradigm, a correlation volume serves as the fundamental module for consecutive frame point matching. Conceptually, the correlation volume at the point-level focus on local regions while correlation volume at voxel-level compensates for large displacements.

3.2.4 Global Motion Aggregation Module

This chapter notices that the flow estimation is significantly degraded or even fails when dealing with large motions, which frequently occurs in non-local regions. To mitigate this issue, an enhanced global motion aggregation module is introduced, which reduces the number of isolated points and the need for masking operation (Song and Yang 2022). This chapter posit that temporal coherence exists between two consecutive point clouds, which can be utilized to build long-distance correlations among two point clouds.

To this end, this chapter first encodes motion feature $\mathcal{E} \in \mathbb{R}^{N \times D_m}$ using the previously estimated flow vector V_{t-1} and the correlation feature $\mathbf{C_f}$ generated by the point-voxel correlation field. Crucially, the motion motion encoder preserves the original motion signals by concatenating the processed features with the raw flow data (transposed for dimensional consistency), ensuring that both temporal relationships and low-level motion dynamics are retained. This architecture balances learned feature extraction with direct motion preservation. Let θ, ϵ, σ denote the projection functions to calculate query, key, and value vector in Fig. 3.7.

The projection functions for the context feature $f_i \in \mathcal{F}_M$ and motion feature $e_i \in \mathcal{E}$ are given by

$$\begin{aligned} \theta\left(\mathbf{f}_i\right) &= \mathbf{W}_{\mathrm{q}}\mathbf{f}_i, \\ \phi\left(\mathbf{f}_i\right) &= \mathbf{W}_{\mathrm{k}}\mathbf{f}_i, \\ \sigma\left(\mathbf{e}_i\right) &= \mathbf{W}_{\mathrm{v}}\mathbf{e}_i. \end{aligned} \tag{3.7}$$

The learnable parameters include weight $\mathbf{W}_{\mathrm{q}}, \mathbf{W}_{\mathrm{k}}, \mathbf{W}_{\mathrm{v}}$, and α. The key and query is from the context feature $\mathcal{F}_M$, while the value is from the motion feature $\mathcal{E}$. The aggregated motion feature is denoted as

$$\hat{\mathcal{E}} = \mathcal{E} + \alpha \sum_{j=1}^{N} h\left(\theta(\mathcal{F}_{M,j}), \epsilon(\mathcal{F}_{M,j})\right) \sigma\left(\mathcal{E}_j\right), \tag{3.8}$$

where α is a hyperparameter initialized to zero. The attention matrix (Eq. 3.9) is utilized for aggregating the value vector that represents temporal coherence. It dynamically weights and aggregates motion features based on their consistency and relevance across sequential frames.

$$h\left(\mathbf{q}_i, \mathbf{k}_j\right) = \frac{\exp\left(\mathbf{q}_i^{\top}\mathbf{k}_j/\sqrt{D}\right)}{\sum_{j=1}^{N}\exp\left(\mathbf{q}_i^{\top}\mathbf{k}_j/\sqrt{D}\right)}. \tag{3.9}$$

The final output is the concatenation $[\mathcal{E}, \hat{\mathcal{E}}]$. An illustration of the global motion aggregation module is provided in Fig. 3.7. Intuitively, concatenation enables flexible merging of motion vectors influenced by contextual attributes, potentially introducing uncertainty during the encoding process prior to decoding the combined motion vector. A critical question arises regarding the inclusion of positional encoding at this stage. Experimental results indicate that positional encoding offers limited value during motion enhancement, as positional information is already incorporated in the earlier feature extraction module. Furthermore, adding positional encoding increases computational overhead without significant performance gains.

To enhance motion feature representation, supervision from the correlation matrix is introduced, improving the model's ability to resolve ambiguities in motion estimation.

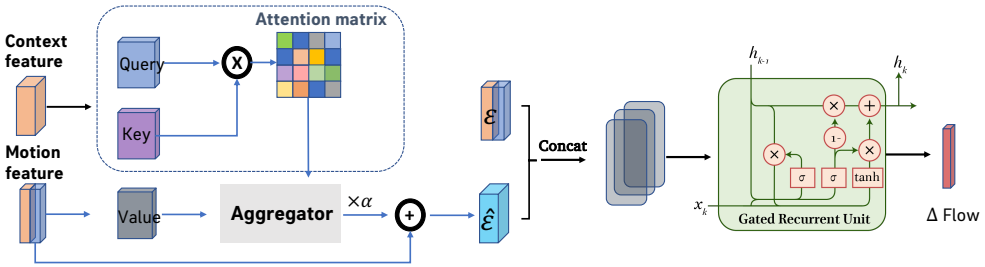


Figure 3.7: The detailed motion feature aggregator. $\mathcal{E}$ denotes the input motion feature for value vector. Context feature is the source for query vector and key vector. The aggregated motion feature is computed based on query, key, and value. The final output is the concatenation of original motion feature $\mathcal{E}$ and aggregated motion feature $\hat{\mathcal{E}}$.

3.2.5 Iterative update

The scene flow estimation is iteratively refined through a GRU-based framework that builds on PV-RAFT's approach (Wei et al. 2021) to integrate voxel and point feature representations. The GRU cell takes three inputs: the contextual feature $\mathcal{F}_M$ (initialized as the hidden state h_0), the concatenated global motion features $[\hat{\mathcal{E}}, \mathcal{E}]$. x_t is initialized by the concatenation of global motion features and $\mathcal{F}_M$ at current iteration step. At each iteration step, the hidden state h_t is updated by combining the previous state h_{t-1} with the input x_t, enabling progressive refinement of motion dynamics and spatial correlations. The components of GRU unit are as follows:

$$\begin{aligned} z_t &= \sigma\left(\mathrm{Conv}_{1\ \mathrm{d}}\left(\left[h_{t-1}, x_t\right], W_z\right)\right) \\ r_t &= \sigma\left(\mathrm{Conv}_{1\ \mathrm{d}}\left(\left[h_{t-1}, x_t\right], W_r\right)\right) \\ \hat{h_t} &= \tanh\left(\mathrm{Conv}_{1\ \mathrm{d}}\left(\left[r_t \odot h_{t-1}, x_t\right], W_h\right)\right) \\ h_t &= (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h_t} \end{aligned} \tag{3.10}$$

The final hidden state h_t is processed by a flow head to produce incremental flow. The updated flow V_{t-1} warps the source point cloud $\mathcal{X}$ into an intermediate translated point cloud $\mathcal{Q}_{t-1} = \mathcal{X} + V_{t-1}$, which serves as input for the subsequent iteration.

Refinement Step: In pursuit of enhanced performance in scene flow estimation networks (Fu et al. 2023), a subsequent refinement step is

implemented to produce the final refined flow prediction V_{ref}. Unlike the pre-training stage, the refinement module only utilizes the final predicted flow vector from iterative update stage and point feature F_Y of the targe frame. This refinement step promotes the smoothness and consistency of flow estimation.

3.3 Loss Terms

The proposed model is trained in a supervised manner, where the flow vectors are iteratively updated. The loss of early iterations is formed as:

$$\mathcal{L}_{iter} = \sum_{t=1}^{T} w_t \left\| (V_t - V_{gt}) \right\|_1 , \tag{3.11}$$

where V_t is the predicted flow vector from the t^{th} iteration in the first updating stage. V_{gt} denotes the ground-truth flow vector. T is the total number of iterations and the weight for t^{th} iteration is w_t. Let $\hat{V}_{ref}$ denotes the refined flow estimation. The loss of the refinement module is

$$\mathcal{L}_{ref} = \left\| \left(\hat{V}_{ref} - V_{gt} \right) \right\|_1 . \tag{3.12}$$

3.4 Experiments

3.4.1 Datasets and Performance Metrics

FlyingThings3D (Mayer et al. 2016) collects rendered stereo and RGB-D images from ShapeNet (Chang et al. 2015), which is the first synthetic benchmark to estimate scene flow. We follow (Wei et al. 2021) to pre-process and separate FlyingThings3D into a training set (19, 640 pairs) and a test set (3, 824 pairs). To evaluate the effectiveness of our model in a real dataset, we choose KITTI scene flow dataset (Menze et al. 2015 2018) and leverage the trained model on FlyingThings3D.

Implementation details. The proposed GAMAFlow is implemented in Pytorch. The number N, M of the input point clouds are set to 8192. The whole network is first trained for 50 epochs, with another 10 epochs for

Method	EPE3D ↓	Acc3DS ↑	Acc3DR ↑	Outliers ↓
PointPWC-Net (Wu et al. 2020)	0.059	0.738	0.928	0.342
FLOT (Puy et al. 2020)	0.052	0.732	0.927	0.357
FlowStep3D (Kittenplon et al. 2021)	0.046	0.816	0.961	0.217
SCTN (Li et al. 2022a)	0.038	0.847	0.968	0.268
PV-RAFT (Wei et al. 2021)	0.046	0.817	0.957	0.292
PT-FlowNet (Fu et al. 2023)	0.031	0.914	0.981	0.175
Ours	**0.027**	**0.929**	**0.981**	**0.146**

Table 3.1: Quantitative evaluation on Flyingthings3D dataset. Lower values are better for the error metrics including EPE3D and Outliers. Higher values are better for the accuracy metrics including Acc3DS and Acc3DR.

the refinement step. Experiments were conducted on a machine equipped with four NVIDIA A100-SXM4-80GB GPUs.

Evaluation Metrics. Several metrics are used for comprehensive comparison. 3D end-point-error (EPE3D) is the mean L2 distance between the ground truth scene flow and predicted result. End point error is calculated to compare the difference between a predicted scene flow vector and a ground truth scene flow vector. It is averaged over all points in meters:

$$\frac{1}{N}\sum_{p\in\mathcal{X}}\left\|\hat{\mathcal{V}}(p)-\mathcal{V}_{gt}(p)\right\|_2 \tag{3.13}$$

where $\mathcal{X}$ is the set of source point cloud with N points. $\hat{\mathcal{V}}(p)$ and $\mathcal{V}_{gt}(p)$ describes the predicted flow and ground truth flow. Strict accuracy (Acc3DS) is the percentage of points whose EPE3D < 0.05m or relative error $< 5\%$. Relaxed accuracy (Acc3DR) is the percentage of points whose EPE3D < 0.1m or relative error $< 10\%$. In benchmarks like FlyingThings3D and KITTI, an outlier is typically defined if absolute error (EPE3D) exceeds 0.3 meters per second or relative error exceeds 5% of the ground truth flow (Liu et al. 2019a).

3.4.2 Quantitative Analysis

This section report the performance of the proposed method compared to the state-of-the-art approches on both FlyingThings3D and KITTI

Method	EPE3D ↓	Acc3DS ↑	Acc3DR ↑	Outliers ↓
PointPWC-Net (Wu et al. 2020)	0.069	0.728	0.888	0.265
FLOT (Puy et al. 2020)	0.056	0.755	0.908	0.242
FlowStep3D (Kittenplon et al. 2021)	0.055	0.805	0.925	0.149
SCTN (Li et al. 2022a)	0.037	0.873	0.959	0.179
PV-RAFT (Wei et al. 2021)	0.056	0.823	0.937	0.216
PT-FlowNet (Fu et al. 2023)	0.023	0.958	0.979	0.121
Ours	**0.022**	**0.963**	**0.983**	0.122

Table 3.2: Quantitative evaluation on KITTI dataset. Lower values are better for the error metrics including EPE3D and Outliers. Higher values are better for the accuracy metrics including Acc3DS and Acc3DR.

dataset. The comparisons are shown in Table. 3.1 and Table. 3.2, respectively. The evaluation results demonstrate that the proposed model, which is trained using a synthetic dataset, exhibits strong generalization capabilities when applied to real KITTI scans. Specifically, **GAMAFlow** reduces EPE3D to 0.027m, achieved a 12% drop from PT-FlowNet (Fu et al. 2023) on FlyingThings3D. The evaluation results confirm the effectiveness of GAMAFlow on this dataset. GAMAFlow also presents superior performance on KITTI in terms of Acc3DS and Acc3DR.

3.4.3 Qualitative Comparison

As shown in Fig. 3.8, the proposed method shows excellent results in three scenes from the synthetic dataset: FlyingThings3D. The results are visualized through an error distribution map, where purple hues indicate minimal errors, transitioning to warmer colors (e.g., red) for larger deviations. The proposed GAMAFlow achieved satisfied *Acc* and avoid large $EPE3D$. The scene flow results on FlyingThings3D and KITTI are compared in Fig. 3.9. Large EPE3D error is highlighted with red rectangle. It is noticeable that GAMAFlow shows the minimum error commpared to the baseline methods: PV-RAFT (Wei et al. 2021) and PT-FlowNet (Fu et al. 2023).

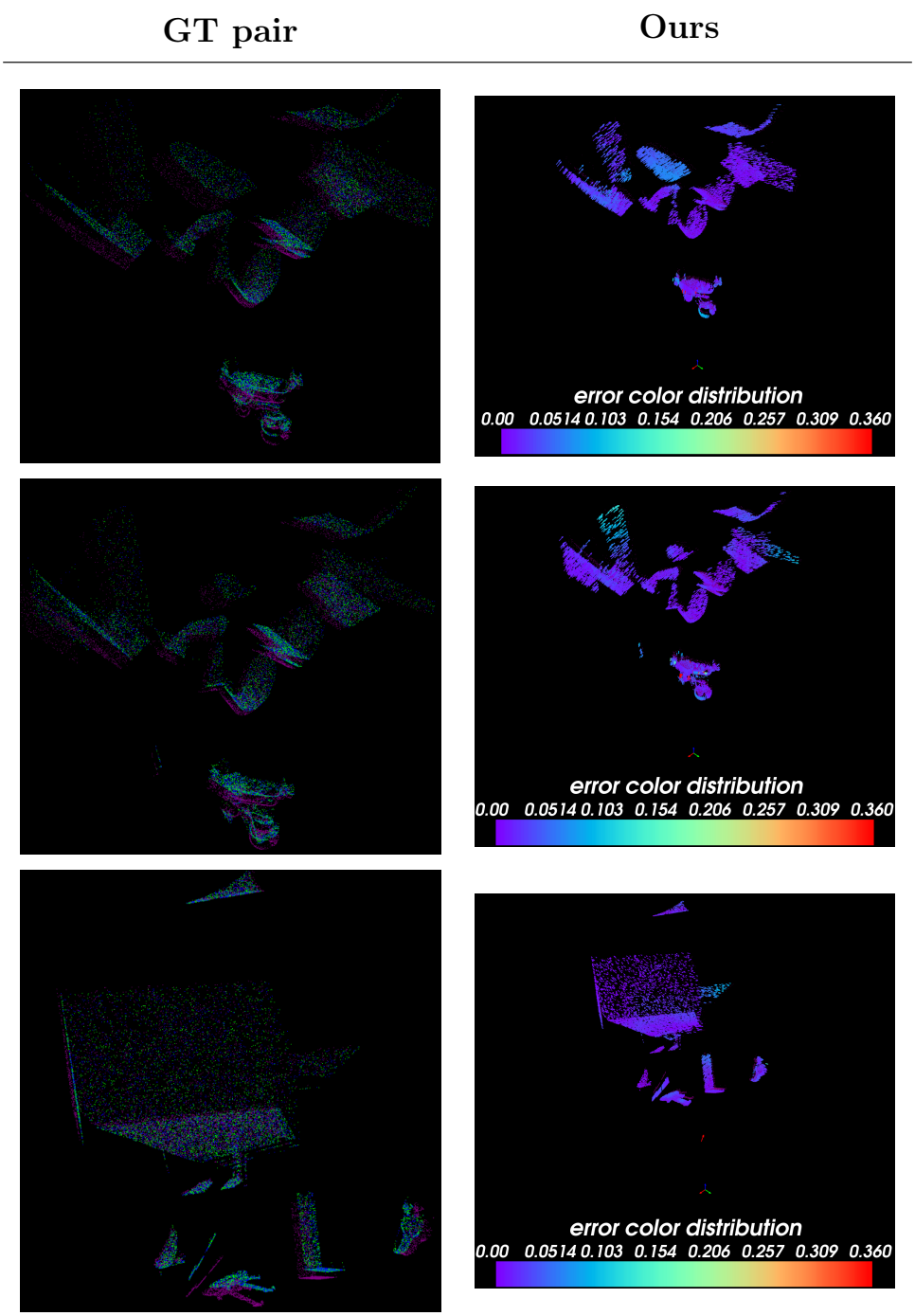
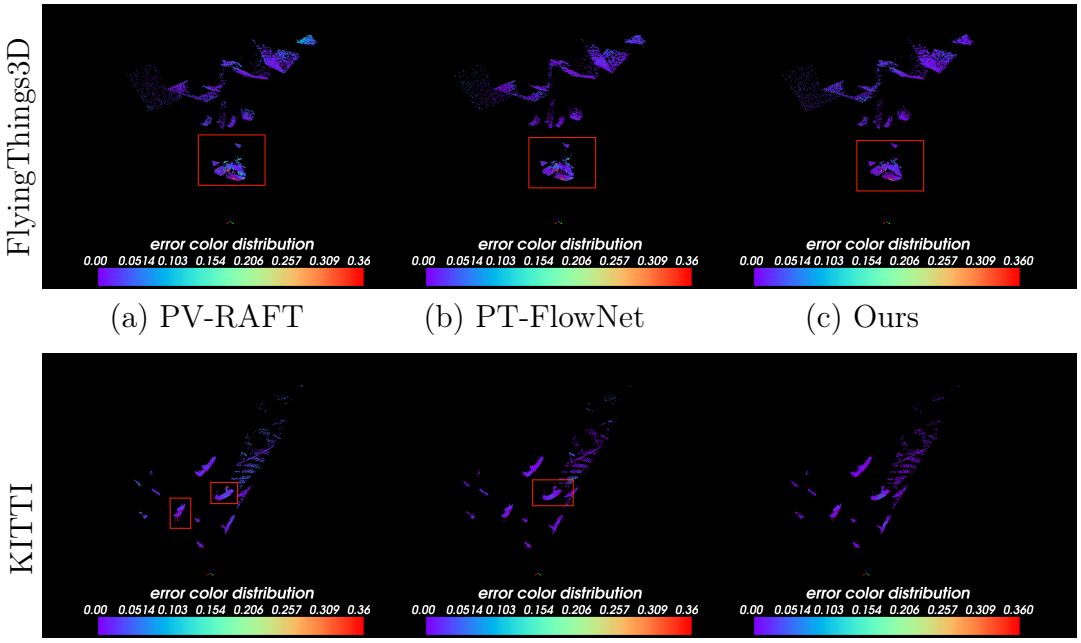


Figure 3.8: Visual Results on FlyingThings3D. Left figures are selected ground truth point cloud frames, where source point cloud is in red and target point cloud in blue. The transformed point cloud with GT flow is shown in green. The right panel illustrates the discrepancy between the target frame and the flow-warped source frame (generated by applying predicted scene flow vectors to the source frame). The error distribution is visualized using a colormap gradient, where purple hues represent minimal deviations and red indicates larger errors.



(a) PV-RAFT (b) PT-FlowNet (c) Ours

Figure 3.9: Visual comparison between PV-RAFT (Wei et al. 2021), PT-FlowNet (Fu et al. 2023), and our method on FlyingThings3D and KITTI dataset. Large error is highlighted in a red rectangle.

3.4.4 Ablation Study

The effectiveness of key components. Ablation experiments are conducted to verify the rationality of the proposed method. Variant I is trained on PT-FlowNet (Fu et al. 2023) without the refinement step. Subsequently, the core components are replaced with grouped attention for feature extraction (II) and global motion aggregation (III). Thirdly, the flow refinement module is incorporated into the plain model with the grouped attention mechanism (IV). The last variant (V) corresponds to the proposed method. As shown in Table 3.3, the grouped attention module improves performance by 13.5%, and the GMA module (Variant III) achieves a 16.2% improvement compared to Variant I.

3.4.5 Flow Refinement Module.

In the proposed framework, it independently integrate both the convolutional layers from (Puy et al. 2020) and a transformer-based refinement module for comparative analysis. Experimental results in Table. 3.3

ID	GA	GMA	FR	EPE3D
I				0.037
II	✓			0.032
III	✓	✓		0.031
IV	✓		✓	0.028
V	✓	✓	✓	0.027

Table 3.3: Ablation study results on grouped attention module and global motion aggregation module. These experiments are conducted on FlyingThings3D.

demonstrate that methods equipped with a refinement module consistently outperform those lacking it.

3.4.6 Running time comparison

Method	FlyingThings3D	
	T=8	T=32
PV-RAFT	293ms	719ms
PT-FlowNet	355ms	886ms
Ours	453ms	985ms

Table 3.4: Ablation study results on time consumption. All experiments are conducted on the same device and the number of points is set to 8192.

As shown in Table 3.4, the proposed method exhibits a higher running time compared to PV-RAFT (Wei et al. 2021) and PT-FlowNet (Fu et al. 2023). This is attributed to the inclusion of the global motion aggregation module, which introduces additional computational costs from its convolutional layers. Furthermore, the runtime scales with the number of iterations due to the iterative refinement process.

3.5 Concluding remarks

In this chapter, a new method: **GAMAFlow** is proposed for scene flow estimation between two point clouds. The core insight of this method is the integration of local motion feature and long-distance global information. Experimental results of **GAMAFlow** on the FlyingThings3D and KITTI datasets demonstrate its effectiveness. GAMAFlow is able to address the generalization challenge and accuracy challenge within reasonable computational cost.

4

Clustering-free unsupervised object segmentation via key points

This chapter introduces a clustering-free method for object segmentation with auxiliary supervision from scene flow. The proposed method tackles the problem of under-segmentation (two or more objects share the same label) and over-segmentation (a rigid object has multiple labels). In the following, the motivation and fundamentals are outlined. Then, the proposed method is introduced, which includes the framework and task-specific design choices. Moreover, two datasets are used for evaluating the proposed method: the Indoor Dynamic Room dataset (OGC-DR) and the outdoor KITTI-SF dataset. The Indoor Dynamic Room dataset is selected for its complex motion patterns and cluttered layouts, which inherently exacerbate segmentation challenges due to occlusions and partial object visibility, while its range of motion scales tests robustness in real-world scenarios. Experiments on two variants of the indoor dataset—one with extended point cloud frames and another with smaller motions—further verify the method's effectiveness.

4.1 Motivation

Originating nearly a century ago, Gestalt theory (Fussell 2023) has influenced countless applications that shape modern life. Rooted in the understanding that humans, as innate order-seekers, subconsciously organize visual elements into patterns and structures. Gestalt theory aims to dissect how this behavior manifests in the perception of images. Its fundamental principles have provided invaluable insights into how humans perceive and organize visual information in complex environments. Despite its true in the real world (where solid objects typically exhibit strong correlations in rigid motions), today's technology struggles to learn how to segment multiple objects all at once from a single point cloud.

The capacity to identify and segment moving objects in sequential data holds paramount importance in various applications (Marichal and Umeda 2003, Kenney et al. 2009, Chen et al. 2023). In particular, discerning and isolating dynamic entities gives insights in autonomous vehicles to navigate complex traffic scenarios (Chen et al. 2021a).

Dynamic shape segmentation in autonomous driving contexts faces two key challenges when processing sequential observations: Large positional shifts: Objects undergo significant displacements within the world coordinate system, complicating continuous tracking of their precise positions; Incomplete data capture: many observations of objects are incomplete or fragmented due to various positioning relative to sensors and heavy obstructions in cluttered environments. These challenges make it difficult to achieve accurate segmentation across frames.

Recent methods have sidestepped the dynamic object analysis, such as 4D Discovery (Wang et al. 2022e), which employ multi modality data to predict object mask in a joint optimization way. A bunch of work studied on grouping similar motion vectors and then predict bounding box and mask for moving objects only (Huang et al. 2021, You et al. 2022). In contrast, the proposed method operates without any training

process or clustering steps. Instead, it focuses on segmenting objects in dynamic point cloud sequences across both single-frame and multi-frame formats, providing a more direct approach to understanding object motion in complex environments.

To effectively segment moving objects, this study adopts a key point based approach rather than relying on the full set of object points. Initially, segmentation signals are predicted for a subset of key points and subsequently propagated across the entire point set using a kernel function. To refine the segmentation, a dynamic loss and a smoothness loss are used together. The full pipeline is illustrated in Fig. 4.1. The proposed method is compact and efficient, with only α as a learnable parameter. The per-point mask is derived through the dot product of the kernel function and key mask. Leveraging rigid object motion as supervisory information further enhances segmentation quality by enforcing geometric consistency constraints. Furthermore, considering the incomplete or occluded views of objects, the proposed method is extended to enable multi-frame segmentation. The results show that the proposed method indeed yields more accurate and robust segmentation outcomes.

This chapter presents a novel approach that employs classical kernel representations from point cloud only. This representation enables the proposed approach to describe the point feature even with dense point clouds while demonstrating exceptional segmentation accuracy —competitive among recent deep approaches. Another advantage of the proposed approach is its generalizability across various out-of-distribution scenarios. Through the positional encoding-based kernel, the proposed approach can effectively predict object mask in dynamic scenes. Systematic evaluations show that the proposed approach is significantly more powerful in predicting object masks, while being lightweight and highly stable in inference.

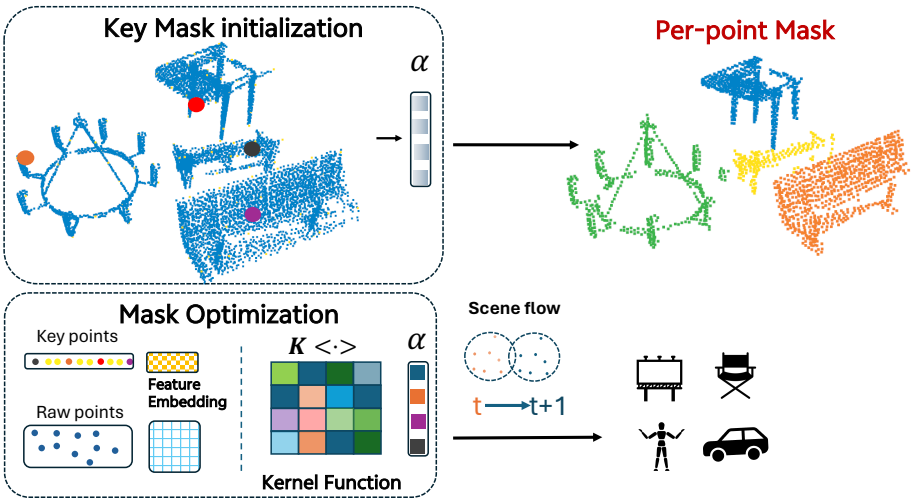


Figure 4.1: To construct kernel function, the proposed method begins by sampling key points from the input point cloud. Three sampling methods are compared in this chapter. Next, K key points and N raw points are embedded to extract corresponding features. The kernel function represents similarity matrix between these two feature embeddings. Finally, a linear coefficient vector α, which depicts the key mask, is optimized per sample to predict the per-point mask. Scene flow vectors are leveraged in the dynamic loss term to optimize the object masks.

4.2 Methods

Given that real-world dynamics encompass both observer motion and object motion, the proposed approach initially compensates for ego-motion before applying the segmentation method to sequence of point clouds. A core challenge in object segmentation lies in balancing under segmentation —where multiple rigid objects are mistakenly merged into a single cluster—and over-segmentation, where a single rigid object is divided into multiple clusters. To address these issues, an optimized segmentation technique is proposed with the following key objectives:

- Achieving segmentation without relying on traditional clustering techniques
- Mitigating the problem of over-segmentation to preserve the integrity of individual objects

- Enabling effective and simultaneous segmentation across multi-frame inputs for consistent dynamic scene understanding

4.2.1 Kernel function

The ability to automatically discover patterns and perform extrapolation is a key feature of intelligent systems. Kernel methods, such as Gaussian processes, are particularly well-suited for pattern extrapolation because the kernel governs the generalization properties of these methods in a flexible and interpretable way (Schölkopf and Smola 2002). However, extrapolating large-scale, multidimensional patterns is generally challenging, and developing Gaussian process models for such tasks presents several difficulties. Most kernels are effective primarily for smoothing and interpolation rather than true extrapolation (Seeger 2004). This challenge is further complicated by the fact that Gaussian processes are typically computationally feasible only for small datasets. Scaling a kernel learning approach that is expressive enough for complex patterns introduces additional difficulties compared to scaling a standard Gaussian process model (Wilson and Nickisch 2015).

In this work, no complex Gaussian process is involved in segmentation prediction. Instead, the proposed method applies a simple kernel function $\mathcal{K} =< x, x' >$, which is an inner product of feature vector. This kernel representation is inspired by FastKernelFlow (Li and Lucey 2024). Specifically, the kernel function is utilized to describe the weight between each key points to raw points in the source point cloud. The key points that are far apart have a lower probability of being picked than the key points that are close together. According to the experimental results, this data-dependent kernels brings more expressibility. The key novelty of the proposed approach lies in extending the classical kernel function, which explores the kernel representation of object mask, ergo achieving a disentanglement of object segmentation and object motion in an unsupervised manner.

In detail, the proposed method first identifies K key points through sampling methods such as random sampling, FPS, or grid sampling. Position embeddings are then extracted for both the key points (K points) and raw points (N points), where $K \leq N$. The position encoding is based on Random Fourier Feature (Li and Lucey 2024), it is defined as:

$$\mathcal{K}_{\mathbf{\Theta}} = \mathcal{K}_{\beta} = \mathcal{K}\left(\phi(\mathbf{p};\beta), \phi\left(\mathbf{p}^{*};\beta\right)\right) \tag{4.1}$$

This kernel function embeds the relationship of key points $\mathbf{p}^{*}$ and raw points $\mathbf{p}$. This chapter compared different kernel functions in Sec 4.5.4. In Equation. 4.1, ϕ denotes feature embedding function. The kernel has size $N \times K$. Formally, the predicted mask of point cloud P is defined as:

$$S = \sum_{i}^{K} \boldsymbol{\alpha}_i \mathcal{K}\left(\mathbf{p}, \mathbf{p}^{*}\right) \tag{4.2}$$

In Equation. 4.2, α denotes the initial parameters, also known as kernel coefficient. It is the optimization target. In our setting, we initialize α as the mask prediction of key points. It has size $K_1 \times K_2$, where K_1 is the number of key points $\mathbf{p}^{*}$ selected and K_2 is a predefined number of objects that is large enough for a specific dataset.

4.2.2 Weight Initialization

Weight initialization method could affect the training process, including the time required to reach the optimized solution and solve the problem of vanishing or exploding gradients. This section explains the criteria for a robust weight initialization method and the preferred choice.

Common initialization methods that satisfy these criteria include *He* initialization (He et al. 2015) (also known as Kaiming initialization) and Xavier initialization (Glorot et al. 2011) (also known as Glorot initialization). These methods scale the variance of the initial weights based on the number of input and output units of the layer to help maintain the magnitude of the gradients during training. The orthogonal initialization

and uniform initialization method are also included in the comparison experiments. Based on the empirical analysis, *He* initialization (He et al. 2015) is particularly well-suited to the proposed framework.

4.2.3 Shared key points

While short-term observations often fail to capture the complete shape of an object, combining multiple frames provides a more comprehensive view. We hypothesize that continuous observations over multiple frames are more beneficial for segmenting moving objects. In this study, partial representation of an object in a sequence are accumulated to form a more complete object shape. Key points is then computed on accumulated point set. To obtain the accumulated point set, each frame in the original sequence is first aligned within a common coordinate system. During this alignment, the proposed method utilizes the ground truth object pose for each frame, transforming each frame to match the first frame by applying the inverse of the respective transformation matrices. This process accumulates points from subsequent frames onto the first frame, ultimately yielding a more comprehensive 3D representation of the detected object, capturing both occluded and partially visible regions.

In this study, the goal is to learn an anchor mask that can propagate to any single frame in the sequence. Let $\mathcal{U}_p = \{\boldsymbol{P}_1, \ldots, \boldsymbol{P}_T\}$ denotes T frames of point clouds in a scene and $\mathcal{U}'_p = \{\boldsymbol{P'}_1, \ldots, \boldsymbol{P'}_T\}$ is the aligned point cloud frames. Each frame of point cloud $\boldsymbol{P}_t = \{(x, y, z)_j\}_{j=1}^{N_t}$ contains point coordinates, where $t \in [1, , T]$ denotes the frame index. N_t denotes the number of points in a single frame. Additionally, the scene flow vector extracted by the flow estimation network (Kittenplon et al. 2021) is define as $\mathcal{F}_p = \{\boldsymbol{F}_1, \ldots, \boldsymbol{F}_T\}$.

Same as Equation. 4.2, the predicted masks of point cloud P in a sequence $\mathcal{U}_p$ are obtained as follows:

$$S_{\mathbf{Pt}} = \sum_{i}^{K} \boldsymbol{\alpha}_i \mathcal{K}\left(\mathbf{P'_t}, \mathbf{P}^*\right), \mathbf{P'_t} \in \mathcal{U}'_p \tag{4.3}$$

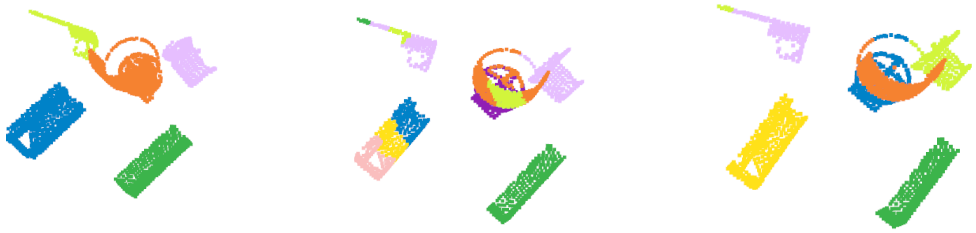


Figure 4.2: From left to right: GT, prediction without smooth loss, prediciton with smooth loss. Over-segmentation: a single rigid object is divided into multiple clusters, as shown in the middle. Smooth loss could address over-segmentation by adjusting neighboring number and searching radius.

where $\mathbf{P}^*$ is sampled from accumulated point set $\mathbf{A_p} = \{\mathbf{P'_1} \oplus \cdots \oplus \mathbf{P'_t}\}$ and is shared across frames.

The number of key points K is initialized to match the maximum expected object count M in the input scene ($K_{\text{init}} = M$). However, standard sampling methods—including Farthest Point Sampling (FPS), Random Sampling, and Grid Sampling—often fail to guarantee at least one key point per physical object due to undersampling or irregular spatial distributions. To address this, the framework employs oversampling by scaling K to $\alpha \cdot M$ ($\alpha \in \{4, 8, 16\}$), ensuring robust coverage even for occluded or densely clustered objects. A detailed comparison on different key point numbers is provided in Sec. 4.5.3.

4.2.4 Loss functions

As shown in Fig. 4.2, some object may have several different label assignments due to over segmentation. To address this issue, a smooth regularization controlled by neighboring number and searching radius is employed, which is defined as:

$$\mathcal{L}_{\text{s}} = \frac{1}{N} \sum_{n=1}^{N} \left(\frac{1}{H} \sum_{h=1}^{H} d\left(\boldsymbol{o}_n, \boldsymbol{o}_{n_h}\right) \right). \tag{4.4}$$

This geometry-aware loss enforces spatial connectivity between object points since physically neighboring points typically belong to the same object. o_n is the mask prediction of center point, followed by neighboring

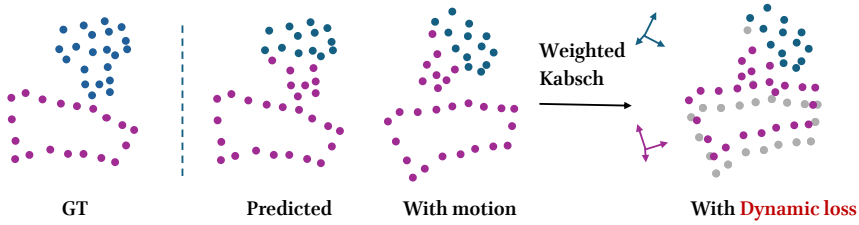


Figure 4.3: Leftmost: two ground truth objects in purple and blue. The right blocks show how the dynamic loss is computed.

point mask prediction $\boldsymbol{o}_{n_h}$. KNN is utilized to search H points from point p_n. This smooth loss then force the neighboring mask assignments to be consistent with the center point p_n. The distance function $d()$ is flexible to choose L1 or L2. A weighted combination of smoothness regularization terms is applied, where (n_1, n_1) represents the number of nearest neighbor point and (r_1, r_2) denotes the searching radius. The weights for the first and second smoothness losses are set to $(3.0, 1.0)$.

When processing point cloud data in dynamic settings, certain challenges arise. For instance, if an inferred object mask encompasses points from two distinct groups moving in different directions, the resulting transformed point cloud might only align with one of these directions, consequently increasing the error rate. The OGC model, as described in Song et al. (2022) (Song and Yang 2022), introduces a dynamic loss function designed to assign different labels to objects exhibiting varied dynamic behaviors. We adopt this dynamic loss to constrain the predicted point masks. The dynamic loss is defined as:

$$\mathcal{L}_{\mathrm{d}} = \frac{1}{N} \sum_{n=1}^{N} \left\| \left(\sum_{k=1}^{K} o_{nk} \cdot (\boldsymbol{T}_k \circ \boldsymbol{p}_n) \right) - (\boldsymbol{p}_n + \boldsymbol{a}_n) \right\|_2 . \tag{4.5}$$

Each point will be assigned a mask $o_n \in (0, 1)$, here o_{nk} denotes the probability of point belongs to the k-th object. $T_k \in \mathcal{R}^{4\times4}$ is estimated transformation matrix generated by Weighted Kabsch algorithm. The Weighted Kabsch algorithm is introduced in Appendix A.2.2. a_n represents the motion of point p_n. A visual illustration of dynamic loss is provided in Fig. 4.3.

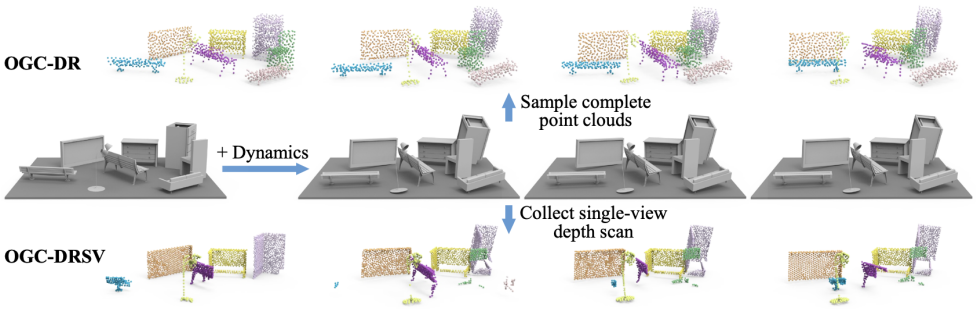


Figure 4.4: The data generation process of OGC-DR and OGC-DRSV, figure from the author of OGC (Song and Yang 2022).

As shown in Fig. 4.3, the blue and purple points belong to two estimated objects. After applying rigid transformations to the two objects, we can easily observe the inconsistency between the desired gray points and blue/purple points in the rightmost block of Fig. 4.3. Given fixed motion estimation, the target is to minimize inconsistency of mask predictions during optimization, the estimated object masks are expected to be better and better with this dynamic constraints. The full loss function is:

$$\mathcal{L} = \frac{1}{T}\sum_{t=1}^{T}\left(w_d\mathcal{L}_t^{\mathrm{d}} + w_s\mathcal{L}_t^{\mathrm{s}}\right) \tag{4.6}$$

The total optimization loss is a weighted sum of smooth loss and dynamic loss, the proposed method sets the weight w_s to 0.1 for smooth loss and $w_d = 10$ for dynamic loss.

4.3 Datasets & Metrics

OGC-DR (Song and Yang 2022), KITTI-SF (Menze and Geiger 2015) are used to evaluate the proposed approach. These datasets correspond to two application scenarios with three datasets: 1) OGC-DR for full-shape indoor furniture arrangements; 2) OGC-DR single-view counterpart OGC-DRSV and its two varaints; 3) KITTI-SF for real-world vehicular traffic.

4.3.1 OGC-DynamicRoom Single View

OGC-DR contains 3,750, 250, and 1,000 indoor scenes for training, validation, and testing, respectively. Each sequence contains four consecutive frames of point clouds (resolution: 2,048 points per frame), where rigid dynamics are simulated by applying random transformations to 4–8 objects selected from 7 ShapeNet categories: chair, table, lamp, sofa, cabinet, bench, display. To mimic real-world partial observations, partial scans are generated by rendering depth images from random viewpoints on the upper hemisphere and un-projecting depth pixels into 3D space. The single view version (OGC-DRSV) has a total of 5,000 scenes, adhering to the original dataset split (3,750 training, 250 validation, 1,000 testing). The generation algorithm for single view dataset OGC-DRSV is provided by OGC (Song and Yang 2022). To unveil the motion prediction and segmentation from long-sequence point clouds. Two varaints of the synthetic dataset OGC-DRSV (OGC-DRSV-$\mathcal{A}$) are created, resulting in OGC-DRSV-$\mathcal{B}$ and OGC-DRSV-$\mathcal{C}$. The OGC-DR and OGC-DRSV generation process is shown in Fig. 4.4.

Here are the details of how we build new dataset:

(1) OGC-DRSV-$\mathcal{B}$. This dataset is interpolated from frame-1 and frame-4 from original OGC-DR(Song and Yang 2022). We use linear interpolation, generating same amount of data frames while with smaller motion scale compared to the original version OGC-DRSV-$\mathcal{A}$. As shown in Fig. 4.5, two new frames are interpolated between the start frame and end frame in each sequence.

(2) OGC-DRSV-$\mathcal{C}$. In the extended version, the object categories and count remain consistent with the original OGC-DR dataset, but the room scale is enlarged to accommodate larger inter-frame motions, mitigating overlaps caused by proximal objects. This extension increases the sequence length to six dynamic frames. Configuration details for OGC-DRSV-$\mathcal{C}$ and OGC-DRSV-$\mathcal{A}$ are summarized in Table 4.1.

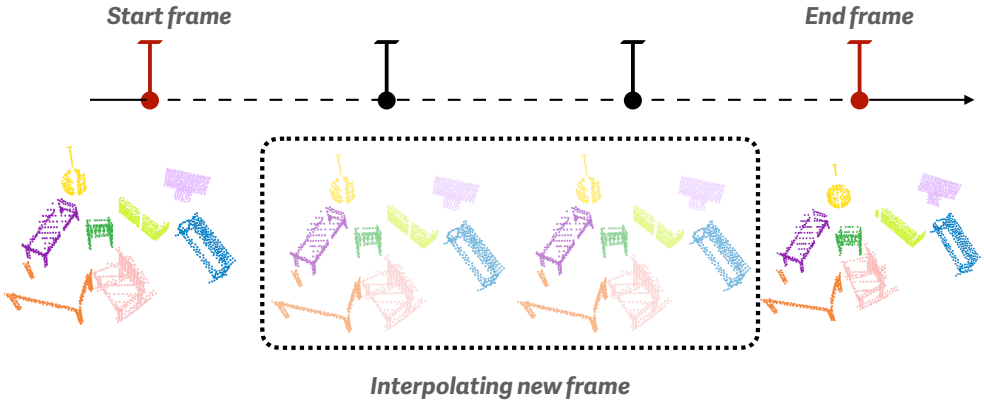


Figure 4.5: A variant of OGC-DRSV: OGC-DRSV-$\mathcal{B}$ which utilizes the original first frame and second frame to interpolate intermediate frames. This dataset has smaller motion scale than OGC-DRSV.

Dataset version	Object scale	Dynamic-rotation	Dynamic-translation	W&L of ground plane
OGC-DRSV-$\mathcal{A}$	0.2 $\sim$ 0.45	Angle[-10,10]	-0.04 $\sim$ 0.04	0.6 $\sim$ 1
OGC-DRSV-$\mathcal{C}$	0.2 $\sim$ 0.45	Angle[-50,50]	-0.2 $\sim$ 0.2	1 $\sim$ 2

Table 4.1: Dataset Configurations for OGC-DRSV $\mathcal{A}$ and its variant OGC-DRSV-$\mathcal{C}$.

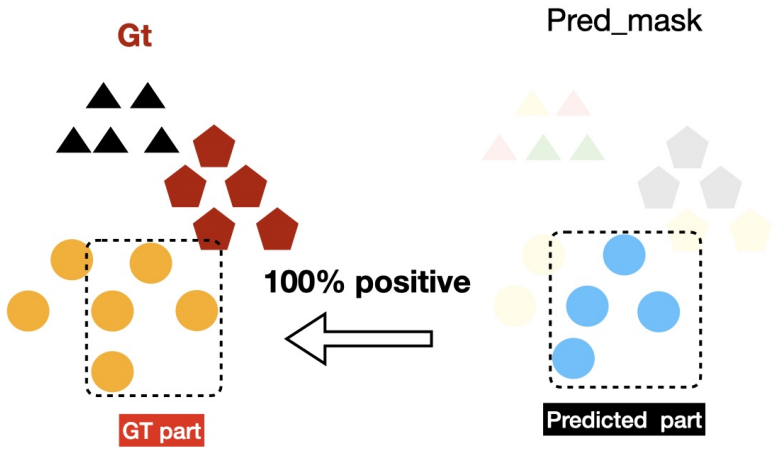
4.3.2 Metrics

The following metrics are used during evaluation: Average Precision (AP), Panoptic Quality (PQ), F1-score (F1), Precision (Pre), and Recall (Rec) at an Intersection over Union threshold of 0.5, in addition to the mean Intersection over Union (mIoU), Rand Index (RI), and Unknown Quality (UQ).

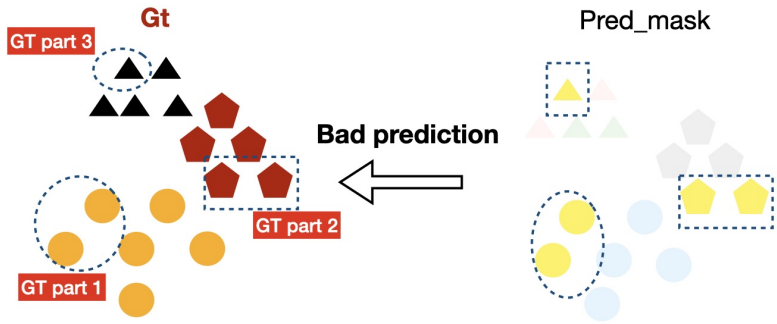
A new metric called proportion (PP) is proposed to evaluate the objectness prediction. The rationale behind this metric is that the predicted label should adhere to the underlying patterns of the objects being segmented. This metric is defined as:

$$pp = max(mask_{gt} \cap mask_{pred})/mask_{pred} \tag{4.7}$$

During inference, a point may be assigned to different labels. As shown in the Fig. 4.6, an object may be assigned with two or more labels, it will not affect the final prediction because the proposed method can refine it through the smooth loss. However, if it is not covered by at



(a) High proportion example.



(b) Low proportion example.

Figure 4.6: Partness metric for evaluating dynamic loss effectiveness. Correct mask predictions (blue labels) achieve high proportion values, while the yellow label demonstrates poor partness capability as it corresponds to three distinct subsets in the ground truth (GT) set.

least one label, or it shares same label with another object. This is an under segment phenomenon, which need to be addressed first by the segmentation algorithm.

4.4 Main Results

This section introduces the main results of the proposed method, compared to other baseline methods. In Sec. 4.4.1, it evaluates the proposed framework by comparing with state-of-the-art methods on indoor synthetic dataset OGC-DR and its single view variants. Additionally, this section presents the results of the proposed framework in real world

Method Category	Methods	AP ↑	PQ ↑	F1 ↑	Pre ↑	Rec ↑	mIoU ↑	RI ↑
Supervised Methods	OGC-sup (Song and Yang 2022)	90.7	82.6	87.6	83.7	92.0	89.2	97.7
	MBSE3-sup (Zhong et al. 2024)	92.8	86.9	91.0	88.8	93.2	91.2	98.7
Unsupervised Methods	TrajAffn (Ochs et al. 2013)	42.6	46.7	57.8	69.6	49.4	46.8	80.1
	SSC (Nunes and Demiris 2018)	74.5	79.2	84.2	92.5	77.3	74.6	91.5
	WardLinkage (Ward Jr 1963)	72.3	74.0	82.5	93.9	73.6	69.9	94.3
	DBSCAN (Ester et al. 1996)	73.9	76.0	81.6	85.8	77.8	74.7	91.5
	OGC (Song and Yang 2022)	92.3	85.1	89.4	85.6	93.6	90.8	97.8
	MBSE3 (Zhong et al. 2024)	93.9	87.0	91.1	87.0	95.6	92.4	98.1
	Ours	**96.6**	**92.4**	**96.5**	**96.4**	**97.2**	**93.2**	**98.5**

Table 4.2: Segmentation performance on OGC-DR. Parameters in our framework: (k_1, k_2): (32,32), (n_1, n_2): (32, 64), (r_1, r_2):(0.16, 0.32), learning rate: 0.004, early patience: 100.

datasets in Sec. 4.4.2.

4.4.1 Performance on OGC-DR and OGC-DRSV

4.4.1.1 OGC-DR.

The synthetic OGC-DR dataset is well-suited for joint point cloud sequence segmentation and scene flow estimation tasks. Comparisons are conducted against unsupervised baselines, including clustering-based methods (DBSCAN (Ester et al. 1996), WardLinkage (Ward Jr 1963)) and motion segmentation methods (TrajAffn (Ochs et al. 2013), SSC (Nunes and Demiris 2018)). As shown in Table 4.2, the proposed multi-frame segmentation framework outperforms all unsupervised baselines, achieving an average precision (AP) of 96.6—a significant improvement over the previous state-of-the-art (92.8). Notably, the method even surpasses fully supervised approaches, demonstrating its robustness in leveraging temporal consistency. This performance validates the effectiveness of using shared key points across frames to enforce spatial-temporal coherence in segmentation.

The OGC-DR dataset is high quality because the inclusion of complete object shapes enables a rigorous evaluation of dynamic scene understanding. The qualitative results (Fig. 4.7) further highlight the superiority of the proposed framework in addressing under/over-segmentation problem. For example, DBSCAN (Ester et al. 1996) has segmentation

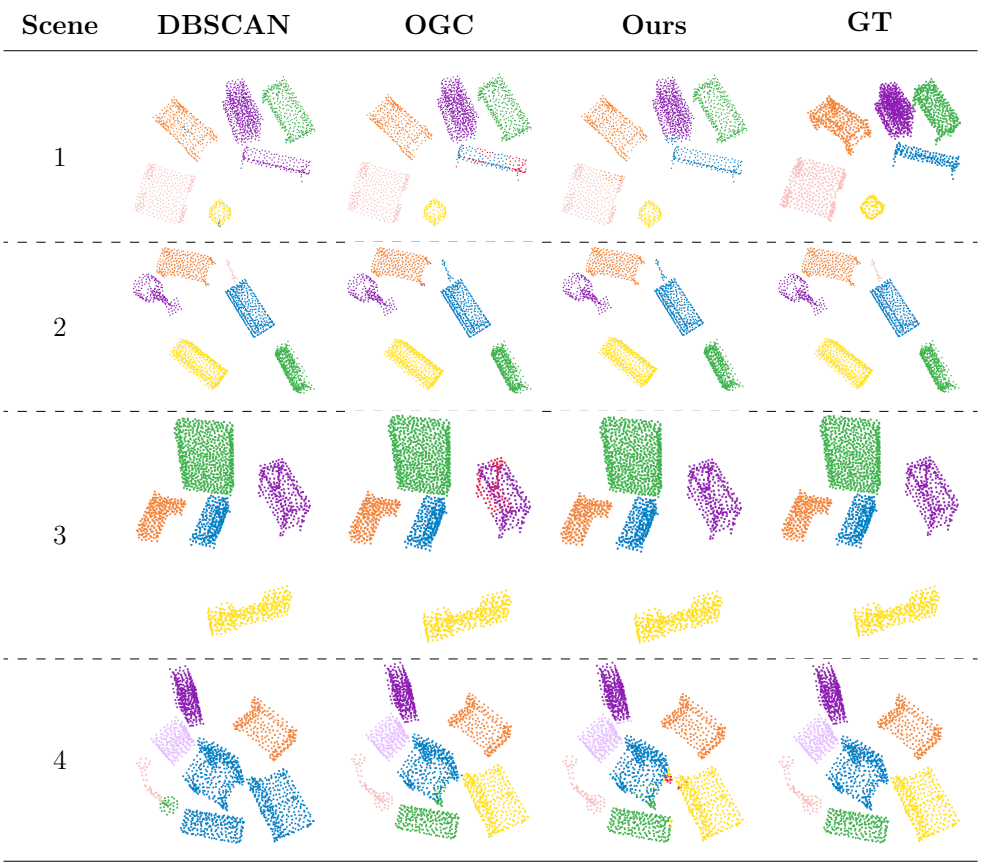


Figure 4.7: Visual Results on OGC-DR. Four distinct scenes with varying object counts are selected for visualization. For optimal clarity, images are best viewed in zoomed mode. From left to right: DBSCAN (eps = 0.05, nsample=10), OGC-R1, our multi-frame method, ground truth (GT). Parameters in the proposed framework: (k_1, k_2): (32, 32), (n_1, n_2): (32, 64), (r_1, r_2): (0.16, 0.32), learning rate: 0.004, early patience: 100.

and OGC (Song and Yang 2022) producess two different labels on the segmentation of a single object, while the proposed method could perfectly segment each object in the first scene.

4.4.1.2 OGC-DRSV

The single view version of OGC-DR presents a challenge due to the incomplete representation of furniture caused by occlusion, which makes it difficult to identify consistent rigidity. As shown in Table. 4.3, the proposed multi-frame mask prediction framework consistently outperforms state-of-the-art models across all metrics, even under the most chal-

Methods	**AP ↑**	**PQ ↑**	**F1 ↑**	**Pre ↑**	**Rec ↑**	**mIoU ↑**	**RI ↑**
TrajAffn (Ochs et al. 2013)	39.3	43.8	54.8	63.0	48.4	45.9	77.7
SSC (Nunes and Demiris 2018)	70.3	75.4	81.5	89.6	74.7	70.8	91.3
WardLinkage (Ward Jr 1963)	69.8	71.6	80.5	91.8	71.7	67.2	93.3
DBSCAN (Ester et al. 1996)	71.9	76.3	81.8	79.1	84.8	80.1	93.5
OGC (Song and Yang 2022)	86.8	77.0	83.9	77.7	91.2	84.8	95.4
MBSE3 (Zhong et al. 2024)	88.1	80.0	86.1	80.8	92.2	86.7	96.6
Ours	**88.9**	**82.1**	**86.6**	79.9	**96.3**	**87.6**	96.2

Table 4.3: Rigid segmentation results on OGC-DRSV-$\mathcal{A}$ compared with state-of-the-art approaches. Parameters in our framework: (k_1, k_2): (256,32), (n_1, n_2): (16, 32), (r_1, r_2): (0.16, 0.32), learning rate: 0.001, early patience: 50.

Config	AP ↑	PQ ↑	F1 ↑	Pre ↑	Rec ↑	mIoU ↑	RI ↑	PP ↑
OGC-R1	87.4	76.2	83.4	75.8	92.5	85.5	95.7	98
OGC-R1 voted	88.4	78.6	85.3	79.4	92.1	85.7	96	97.6
Ours-single	82.8	70.8	78.9	71.9	90	82.1	94.9	97
Ours-voted	**89.6**	78.5	**85.5**	**81.4**	91.8	84.9	**96.1**	96.7

Table 4.4: Ablation results about the voting mechanism in our single-frame input optimization framework on the OGC-DRSV dataset. The configuration of optimization algorithm: lr = 0.004, early patience= 100, (k_1, k_2): (32, 32), (n_1, n_2): $(16, 32)$, (r_1, r_2) : $(0.08, 0.16)$. The OGC results are derived from the initial training round to ensure a fair comparison.

lenging condition—unsupervised single-view. The proposed approach achieves an AP of 88.9%, underscoring its robustness in handling incomplete observations.

To rigorously evaluate the capabilities of the proposed framework, an ablation study is conducted comparing single-frame segmentation performance against the baseline method OGC (Song and Yang 2022) and its voting-enhanced variant. As shown in Table 4.4, the proposed method achieves 82.8% AP in the single-frame setting, trailing the OGC benchmark by 4 percentage points (87.4% AP). However, after integrating the voting mechanism, which enforces the consistency of the label by averaging the predictions in a 3-frame sliding window, the proposed framework

	Config	Object Segmentation Metrics							
		AP ↑	**PQ** ↑	**F1** ↑	**Pre** ↑	**Rec** ↑	**mIoU** ↑	**RI** ↑	**UQ** ↑
OGC-DRSV-$\mathcal{A}$	GT flow-Ours	95.5	86.9	91.5	86.9	97.8	92.3	98	92.5
	Pred flow-Ours	89.4	75.5	82.3	73.8	95.4	87.8	96.2	87.0
	Pred flow OGC	88.4	78.6	85.3	79.4	92.1	85.7	96.0	-
OGC-DRSV-$\mathcal{B}$	GT Flow	95.5	90.7	95.1	94.6	96.3	91.4	98.2	98.0
	Pred flow-Ours	80.9	67.0	74.7	66.4	88.1	81.3	94.1	78.7
	Pred flow OGC	83.2	78.6	85.0	84.7	85.3	81.7	96.4	-
OGC-DRSV-$\mathcal{C}$	GT Flow-Ours	96.7	88.0	91.7	86.6	98.6	94.1	98.2	99.4
	Pred flow-Ours	93.0	81.9	86.8	79.6	97.1	91.6	97.3	91.3
	Pred flow OGC	90.4	83.0	88.2	84.7	92.0	87.7	96.5	-

Table 4.5: Multi frame segmentation results on two variants of OGC-DRSV: Ver-$\mathcal{B}$ with motion scale = 0.024 and Ver-$\mathcal{C}$ with motion scale = 0.043. All groups share the same optimization configuration (lr = = 0.004, early stopping patience = 100, $(k_1, k_2) : (32, 32)$, $(n_1, n_2) : (16, 32)$, $(r_1, r_2) = (0.08, 0.16)$). Object segmentation performance and scene flow quality are reported on the testing set. For fair comparison, the OGC baseline employs unsupervised training limited to a single iteration. The results of OGC is trained in unsupervised manner for only one round.

achieves 89.6% AP, outperforming OGC's refined voting results (88.4%).

The results for OGC-DRSV-$\mathcal{B}$ and OGC-DRSV-$\mathcal{C}$ are presented in Table. 4.5. In both datasets, predicted flow is generated using FlowStep3D (Kittenplon et al. 2021), while ground truth flow is computed via rigid transformations based on ground truth object poses. OGC-DRSV-$\mathcal{C}$, which extends sequence length while preserving the motion scale of OGC-DRSV-$\mathcal{A}$, challenges methods with prolonged temporal dependencies. Despite this complexity, the proposed framework achieves competitive segmentation accuracy (AP: 90.4%), closely matching the OGC baseline (Song and Yang 2022) (AP: 93%). This highlights the method's robustness to extended sequences without sacrificing spatial coherence. However, the performance on OGC-DRSV-$\mathcal{B}$ is less satisfactory in the AP and PQ metrics. This disparity likely stems from the reduced motion scale in OGC-DRSV-$\mathcal{B}$, which amplifies ambiguity in motion cues—critical for the proposed method's loss module. This suggests that instance-aware metrics (AP, PQ) are more sensitive to subtle

Method Category	Methods	AP ↑	PQ ↑	F1 ↑	Pre ↑	Rec ↑	mIoU ↑	RI ↑
Unsupervised Methods	TrajAffn (Ochs et al. 2013)	24.0	30.2	43.2	37.6	50.8	48.1	58.5
	SSC (Nunes and Demiris 2018)	12.5	20.4	28.4	22.8	37.6	41.5	48.9
	WardLinkage (Ward Jr 1963)	25.0	16.3	22.9	13.7	**69.8**	**60.5**	44.9
	DBSCAN (Ester et al. 1996)	13.4	22.8	32.6	26.7	42.0	42.6	55.3
	OGC (Song and Yang 2022)	**36.0**	24.6	**35.4**	26.4	53.8	53.7	57.8
	Ours	25.2	**34.9**	**36.8**	**96.2**	24.7	23.3	**87.0**

Table 4.6: Quantitative results on KITTI-SF. Compared baseline methods include unsupervised algorithms: TrajAffn, SSC, WardLinkage, DBSCAN, and OGC. The proposed method is evaluated in unsupervised manner. The results of OGC are collected from the first round training for a fair comparison.

motion variations. In contrast, mIoU's reliance on region-based overlap rather than instance-level precision allows it to remain stable.

4.4.2 Performance on KITTI-SF

To validate the generalization capability of the proposed method, additional experiments on the challenging real-world outdoor KITTI Scene Flow (KITTI-SF) dataset are conducted. This dataset consists of 200 training pairs captured from real-world traffic scenes and an online hidden test for scene flow estimation (Song and Yang 2024). In the experiments, the last 100 pairs from the dataset were selected to form the testing set, comprising 200 individual point cloud frames. The quantitative results is presented in Table. 4.6.

Empirically, the human annotations of cars and trucks are kept in each frame to compute the segmentation scores. Other objects are grouped into ground points. Due to the extreme imbalance of 3D points between foreground objects and background, KITTI-SF is considered as an challenging dataset in the literature. For instance, clustering based algorithms like DBSCAN tends to favor objects that distributed far apart. Even in OGC (Song and Yang 2024), prior knowledge on ground planes is required to assist the segmentation of objects in the foreground. KITTI-SF is not a strictly defined multi-body rigid dataset, given that the background elements within its point clouds have the potential to undergo

deformation. Nevertheless, the proposed method still surpasses classical clustering-based algorithms such as DBSCAN (Ester et al. 1996) and achieves comparable results to other baseline methods.

The performance gap relative to the learning-based OGC method (Song and Yang 2022) may stem from two factors: (1) Sparsity of moving objects: Autonomous driving datasets often contain fewer than 20% moving objects (Khatri et al. 2025), posing challenges for frameworks relying on object motion analysis through dynamic loss components. This limitation is amplified in scenes dominated by static structures, where motion cues are insufficient for robust segmentation. (2) Optimization scope: Unlike learning-based approaches that leverage full-dataset training to iteratively refine predictions across sequences, the proposed framework employs per-scene optimization. While this avoids overfitting to dataset-specific biases, it sacrifices the error-correction benefits of end-to-end training, where coherent temporal predictions across frames improve generalization.

4.5 Ablation Studies

This section studies the contributions of different components described in Sec. 4.2 , including the selection of key points, number of key points, kernel function type, effect by flow source, as well as smooth loss settings.

4.5.1 Flow source

There are two groups of flow source used to compare the performance:
Group (i): We simulate the scattered signal of the background environment where the target is located by adding random Gaussian noise.
Group (ii): We substitute the best flow estimation with a flow estimation generated by a flow model that is only trained for 30 epochs.

		Object Segmentation								Scene Flow	
	Flow Source	**AP** ↑	**PQ** ↑	**F1** ↑	Pre ↑	**Rec** ↑	**mIoU** ↑	**RI** ↑	UQ ↑	**EPE3D** ↓	**AccR** ↑
(i)	GT + Gaussian (std=0.1)	86.1	85.5	90.6	95.1	87.7	82.8	95.9	82.9	0.16	1
	GT + Gaussian (std=0.05)	90.4	88.9	93.3	95.8	91.8	87.2	97.3	87.6	0.08	14
(ii)	FlowStep3D (epoch= 30)	45.5	29.1	37.9	27.3	65.2	61.8	85.7	49.7	0.11	24.5
	FlowStep3D (epoch=50)	89.4	75.5	82.3	73.8	95.4	87.8	96.2	87	0.02	72.2
(iii)	GT flow	95.5	86.9	91.5	86.9	97.8	92.3	98	92.5	0	100

Table 4.7: Ablation results about the robustness to scene flow distortions on OGC-DRSV. The robustness of the proposed multi-frame approach to scene flow distortions on OGC-DRSV. All groups share same configurations of optimization: lr = 0.004, early patience= 100, $(k_1, k_2) : (32, 32)$, $(n_1, n_2) = (16, 32)$, $(r_1, r_2) = (0.08, 0.16)$. This table reports the object segmentation performance on the testing set and scene flow quality on testing set.

According to Table. 4.7, the proposed method is robust to Gaussian noise in scene flows. The AP of noise flow with std 0.05 maintains 90.4 even the quality of flow degrades. In contrast, flow distortions caused by undertrained estimators lead to a significant decrease in segmentation performance, where AP drops from 89.4 to 45.5. Based on this comparison, the proposed method demonstrates robustness against noisy flows with significant variance but is more susceptible to large biases in the estimated scene flows.

4.5.2 Key points selection

Point sampling is a frequently employed technique in the field of point cloud analysis, particularly when the original point cloud set contains an enormous quantity of points. As per the existing literature, three of the most commonly utilized sampling methods—Farthest Point Sampling, Random Sampling, and Grid Sampling—are selected as plug-in-play modules for the proposed framework. Except sampling methods, other parameters remain unchanged to achieve a fair comparison. The results are shown in Table. 4.8. RS and FPS perform similarly, showing comparable results across all metrics. FPS is used in other experiments as it can balance the inference speed and segmentation performance.

Sample Method	AP ↑	PQ ↑	F1 ↑	Pre ↑	Rec ↑	mIoU ↑	RI ↑	PP ↑	UQ ↑
RS	89.3	75.4	82.2	73.7	95.3	87.7	96.2	98.3	87.0
FPS	89.5	75.6	82.5	74.0	95.5	87.7	96.2	98.3	87.0
Grid	83.7	72.2	81.0	76.4	88.5	80.2	94.6	95.8	78.6

Table 4.8: Sampling methods comparison in multi-frame approach. RS denotes Random sampling. FPS represents Farthest Point Sampling. Grid is Grid Sampling. Configurations of optimization: lr = 0.004, early patience= 100, $(n_1, n_2) = (16, 32)$, $(r_1, r_2) = (0.08, 0.16)$.

Key & Object	**AP**	**PQ**	**F1**	**Pre**	**Rec**	**mIoU**	**RI**	**UQ**	**PP**	**Time**
(8, 8)	65.9	55.1	70.3	69.1	73.4	63.2	87.6	57.4	87.8	14.6
(32, 32)	89.5	75.6	82.5	74.0	95.5	87.7	96.2	87.0	98.3	20.2
(64, 64)	87.9	76.9	82.6	73.7	96.3	89.6	96.8	89.1	99.2	24.0
(128, 128)	83.5	75.9	81.4	72.1	96.3	89.7	96.7	89.3	99.2	22.2
(256, 256)	79.7	72.7	78.4	68.1	95.5	88.7	96.3	87.9	99.1	32.7
(512, 512)	77.1	70.0	76.3	65.2	95.2	87.6	95.8	86.8	99.0	46.5

Table 4.9: Ablation of key number and maximum object number on OGC-DRSV dataset. Configurations of optimization: lr = 0.004, early patience= 100, $(n_1, n_2) = (16, 32)$, $(r_1, r_2) = (0.08, 0.16)$.

4.5.3 Key points &max object number

Five different sets of parameters are used to compare the performance of the proposed segmentation algorithm. Notably, the number of key points significantly impacts the evaluation, as both recall and precision are closely tied to how well the predicted masks match the ground truth. As shown in Table. 4.9, the configuration with 32 key points and a maximum of 32 possible objects achieves the best overall performance. Although the actual number of objects in the dataset is considerably smaller than 32, this configuration enhances performance by allowing the model to estimate a greater number of potential object masks, leading to more accurate segmentation. The increased mask quantity likely improves the model's capacity to capture finer details and better handle object occlusions or overlaps. This flexibility in segment mask prediction seems to provide a balanced trade-off between precision and recall, particularly in complex scenes where under-segmentation would reduce the

Method	AP ↑	PQ ↑	F1 ↑	Pre ↑	Rec ↑	mIoU ↑	RI ↑	UQ ↑
Softmax	84.7	72.1	79.6	70.0	95.3	86.8	95.7	86.0
RBF kernel	89.5	75.6	82.5	74.0	95.5	87.7	96.2	87.0

Table 4.10: Ablation of kernel function. Results on OGC-DRSV test set. Parameters: lr = 0.004, early stopping patience = 100, $(k_1, k_2) : (32, 32)$, $(n_1, n_2) = (16, 32)$, $(r_1, r_2) : (0.08, 0.16)$.

recall, and over-segmentation would hurt precision.

4.5.4 Kernel function

Table. 4.10 compares the segmentation performance of the proposed framework using two distinct kernel functions: the standard Softmax and the Radial Basis Function (RBF) kernel. Results are evaluated on the OGC-DRSV test set. The RBF kernel demonstrates consistent superiority over Softmax, achieving 89.5% AP and 75.6% PQ—a significant improvement of +4.8 AP and +3.5 PQ compared to the baseline. Notably, the RBF kernel enhances robustness in boundary-aware metrics (e.g., +1.7 mIoU), likely due to its capacity to model non-linear spatial relationships. This comparison validates the effectiveness of RBF-based feature aggregation for dynamic point cloud segmentation.

4.5.4.1 Softmax kernel

The softmax kernel is a kernel function applied directly to the inner product $< \cdot >$ of two feature vectors. This approach simplifies the parameter space by reducing the number of hyperparameters from two (the positional encoding scale and the kernel scale) to just one (the positional encoding scale). By leveraging the softmax function, the kernel acts as a selection matrix, emphasizing the most dominant features from the similarity matrix, thus enhancing the most relevant features for improved representation. It is defined as:

$$\mathcal{K}_{\boldsymbol{\beta}} = \text{Softmax}\left(\langle \phi(\mathbf{p}; \boldsymbol{\beta}), \phi\left(\mathbf{p}^*; \boldsymbol{\beta}\right)\rangle\right) \tag{4.8}$$

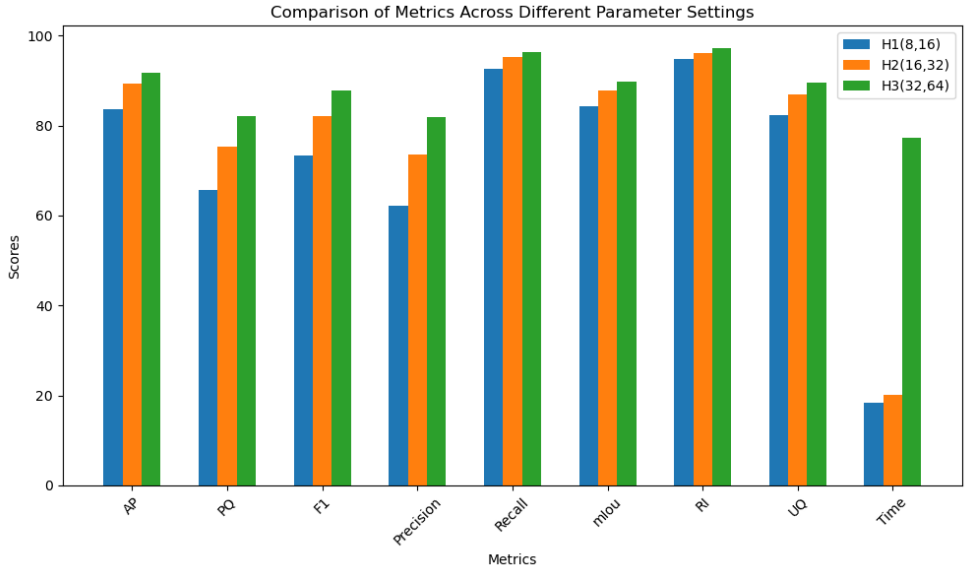


Figure 4.8: Segmentation Results for three different smooth regularization settings. H_1 : $(8, 16)$, $(0.02,\ 0.04)$, H_2 : (16,32), (0.08, 0.16), H_3: (32,64), (0.08,0.16). Results are evaluated on the OGC-DRSV test set across eight metrics, including Average Precision (AP), Panoptic Quality (PQ), F1 score, Precision (Pre), Recall (Rec), mean Intersection-over-Union (mIoU), Rand Index (RI), Uncertainty Quality (UQ), and processing time.

4.5.4.2 RBF Kernel

When dealing with data that has a nonlinear structure, the radial basis function (RBF) kernel is a good choice to learn relationships between point sets. The radial basis function kernel is defined as:

$$\mathcal{K}\left(\mathbf{p}, \mathbf{p}^*\right) = \exp\left(-\frac{\left\|\mathbf{p} - \mathbf{p}^*\right\|^2}{2\sigma^2}\right) \tag{4.9}$$

We use it to capture the subtle variations in point sets. RBF kernel is able to focus on and learn similarities in local regions of a point set, which can help improve accuracy in distinguishing between similar and dissimilar sets. The RBF kernel primarily depends on a single parameter (the bandwidth or scale parameter), which controls the spread of the kernel. This single parameter can be adjusted to tune the sensitivity to variations within the point set, making it relatively simple to optimize compared to other more complex kernels with many hyperparameters (Cortes et al. 2002, Parra and Tobar 2017).

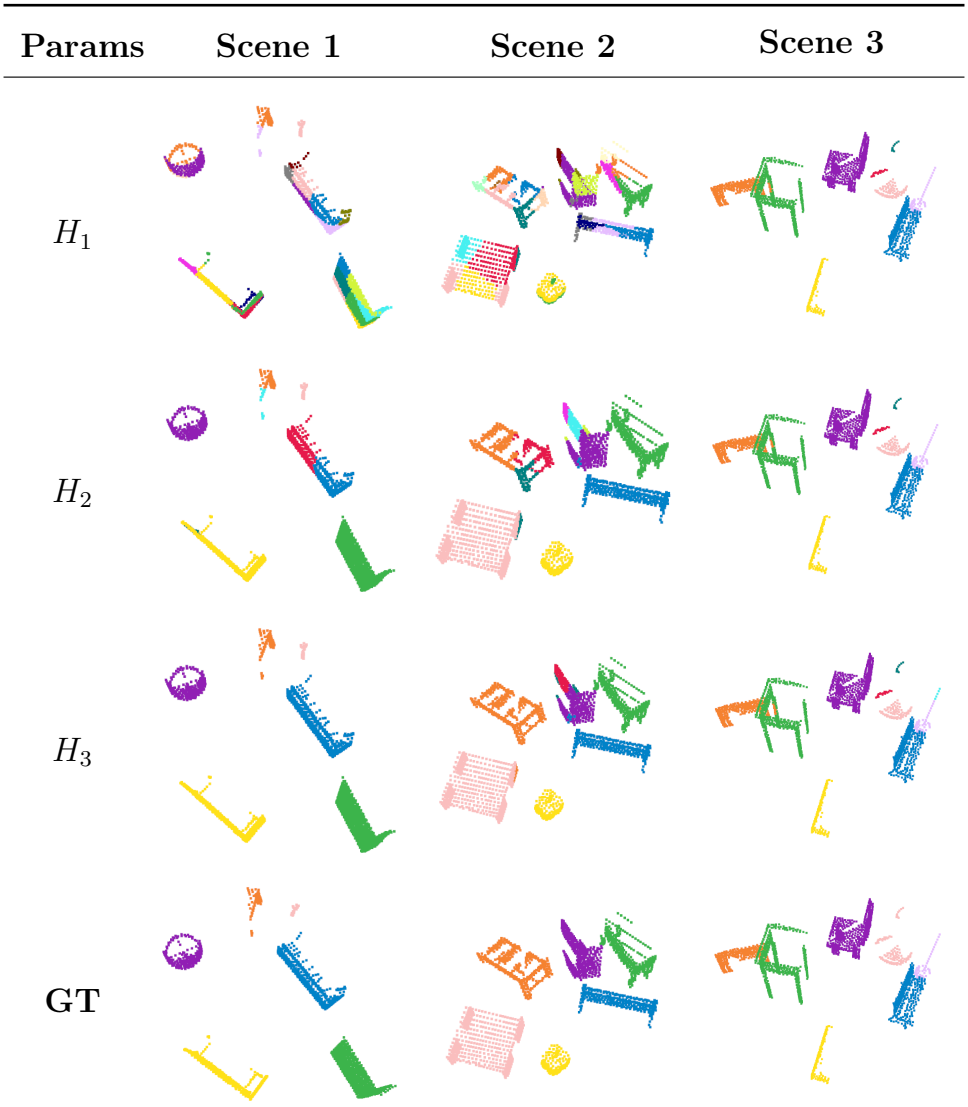


Figure 4.9: Visual Results on OGC-DRSV for three smooth regularization settings. Parameters held constant across comparisons for H_1, H_2, H_3 include: key points and max object number: (32,32), learning rate: 0.004, early patience: 100.

4.5.5 Smooth loss

This section evaluates the influence of smoothness regularization hyperparameters on the OGC-DRSV dataset. Smooth loss address over-segmentation through enforcing the mask to be consistent within an geometrical area. Moreover, when the regularization is strengthened by enforcing smoothness in a larger local neighborhood, the Precision score is improved with less over-segmentation. Fig. 4.8 shows qualitative re-

Figure 4.10: Failure case when two objects have the same mask prediction (red ellipse in the rightmost figure).

sults. It can be seen that larger neighboring area brings higher performance. Expanding the neighboring region (e.g., increasing search radius or neighbor count) allows the model to aggregate richer geometric and semantic features from a broader spatial context. As shown in Fig. 4.9, the smooth parameter H_3 effectively reduces over-segmentation artifacts by enforcing stronger spatial coherence. However, this enhancement entails a computational cost: H_3 significantly increases runtime due to its reliance on larger neighborhood sizes. Algorithms like K-nearest neighbors (KNN) or radius search scale non-linearly with neighborhood size. According to Fig. 4.8, optimal parameters (H_2 setting) are often determined empirically to balance accuracy and latency for target hardware. The parameter set H_2 is used in other ablation studies to save time.

4.6 Concluding remarks

Contributions. This chapter shows that the classical kernel function provides a direct object mask prediction in a non-learning setting. Key mask initialization and key mask propagation are implemented in a compact and efficient manner, which could be used in various architectures. Dynamic loss and smooth loss are employed to address the under segmentation and over segmentation problem, respectively. Experimental results confirm that the proposed method has adaptability to both a small motion sequence and a longer sequence. The proposed method succeeds in generating an accurate object mask for dynamic scene analysis.

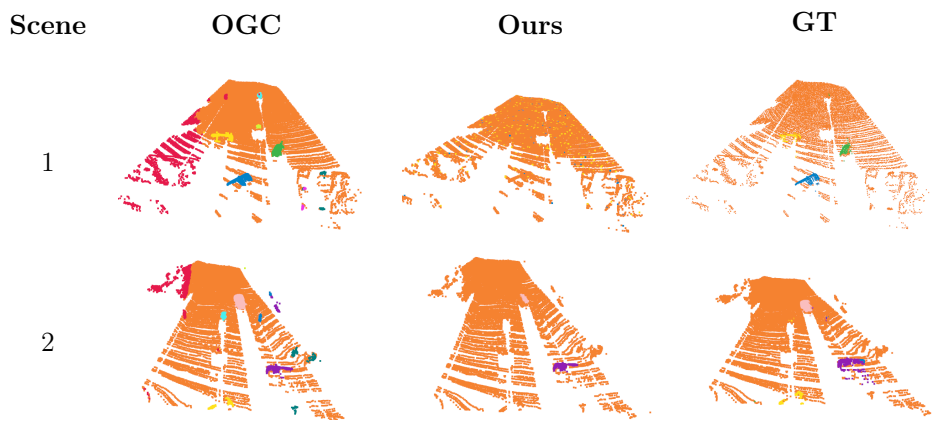


Figure 4.11: Visual comparisons are shown for three methods (left to right): OGC-R1, the proposed single-frame method, and ground truth (GT). For optimal clarity, images should be viewed in zoomed mode. In the first scene, the proposed method (middle column) struggles to generate coherent object masks, likely due to insufficient motion cues. In contrast, OGC-R1 (left) successfully identifies most instances, albeit with minor false positives (e.g., fragmented masks in static regions).

Limitations. One issue is that the loss functions are not always sufficient to ensure distinct mask predictions for object parts that are close to each other (see Fig. 4.10). Furthermore, the optimization results on the KITTI-SF dataset did not meet expectations. As shown in Fig. 4.11, the proposed method failed to produce reasonable segmentation results compared to OGC (Song and Yang 2022). The proposed method inherently prioritizes scene-specific adaptation, which may limit scalability in large-scale applications. In contrast, conventional learning-based methods leverage full-dataset training to iteratively refine predictions across sequences, balancing errors and improving average performance through holistic optimization. To bridge this gap and enhance the framework's generalizability while retaining its strengths, Chapter 5 introduces a novel training strategy that integrates key points masks with dataset-wide learning. This approach systematically validates the current methodology while extending it to address scalability challenges, enabling efficient adaptation across diverse scenarios without sacrificing temporal coherence or segmentation precision.

5

Learning to Segment 3D Objects from Multiple Point Cloud Frames

This Chapter introduces a learning framework for object segmentation from multiple frames. Morepreciselly, the framework is validated in both fully supervised manner and unsupervised manner. Based on the limitations discussed in Chapter 4, this chapter improves the segmentation results on KITTI-SF dataset through a time-independent query. In the following, a brief motivation is given why multiple frames promise to achieve increased performance. Then, the methodology and network architectures are presented. Finally, extensive experimental results are illustrated.

5.1 Motivation

Recently, significant progress has been made in 3D instance segmentation (Vu et al. 2022, Wang et al. 2018, Liu et al. 2020, Yang et al. 2024), enhancing both the accuracy and efficiency of object recognition and segmentation. However, fully-supervised training for this task often requires extensive human annotations, making it a costly and labor-intensive pro-

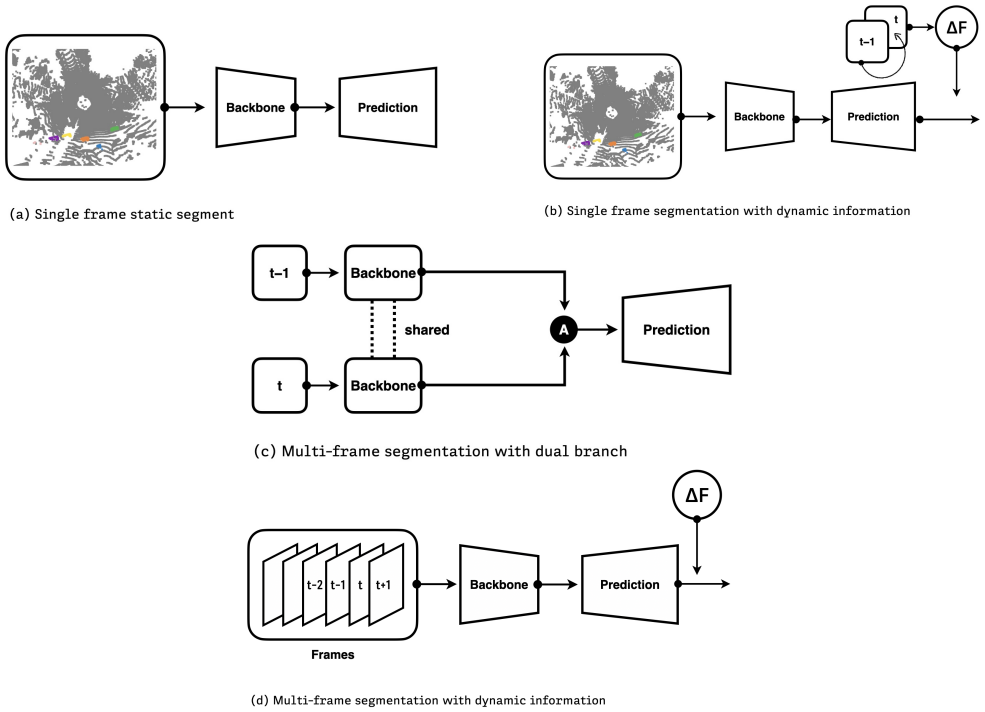


Figure 5.1: Different frameworks to obtain instance segmentation in static input and dynamic input. (a) Single frame static segmentation: segment individual scans (Triess et al. 2020, Hui et al. 2022, Ren et al. 2024). (b) Single frame segmentation with dynamic supervision (Song and Yang 2024, Zhong et al. 2024). (c) Multi-frame: segment individual scans and associate predictions over time (Marcuzzi et al. 2022, Hong et al. 2021). (d) Ours: directly segment multi-frame data without association between individual predictions.

cess. In the context of point cloud processing, single frame static segmentation focuses on segmenting objects within a single point cloud (Triess et al. 2020, Hui et al. 2022, Ren et al. 2024), while dynamic segmentation deals with the temporal changes across multiple frames to track and predict the motion of objects (Marcuzzi et al. 2022, Hong et al. 2021). Static segmentation is crucial for tasks where the scene is captured in a single shot, such as in architectural modeling or in a stationary scene. On the other hand, dynamic segmentation is vital for applications that involve motion, such as in robotics or surveillance systems where the movement of objects over time is critical.

Exploring objectness in given dynamic sequences is quite tricky without any annotations. This study argues that geometry consistency and

motion pattern consistency are the most obvious cues to distinguish moving object in dynamic sequences, which is to say, points that belongs to the same rigid object should hold the same geometry characteristics, and at least share same moving direction and speed.

Then it comes to the key question in this hypothesis. *How to learn motion patterns between consecutive frames?* To this date, most segmentation methods in the literature relies on object bounding boxes to detect object. One solution is first detect object then estimate object-level scene flow to explicitly learn the motion patterns (Khatri et al. 2024). In this way, a perfect detector and tracker will produce perfect flow. However, it requires strong supervision on the object bounding box to detect interested objects. The detection framework consists of clustering local points into several segment, then apply Kalman filters to track segmented objects across time. Kalman filters is used to predict the state of an object based on its previous state and measurements. Another attempt, as proposed in (Wang et al. 2022e), trains a ClusterNet with supervision from motion cues. The 3D instance segmentation output from ClusterNet can guide the localization network when projected onto a 2D image. In turn, leveraging appearance information from 2D detection can refine the 3D instance segmentation process.

Previous work such as OGC (Song and Yang 2022) and MBSE3 (Zhong et al. 2024) integrates scene flow estimation with object segmentation; however, they only leverages scene flow predictions from two point cloud frames for segmentation. Existing methods for multi-frame segmentation adopt a shared backbone for all frames (Marcuzzi et al. 2022). Subsequently, VCSF (Vogel et al. 2014) introduced a view-consistent multi-frame scene flow estimation architecture specifically for stereo video. Both OGC (Song and Yang 2024) and VCSF utilize a sliding temporal window to enforce consistency across frames, albeit through different approaches. VCSF, for instance, represents a scene as a set of

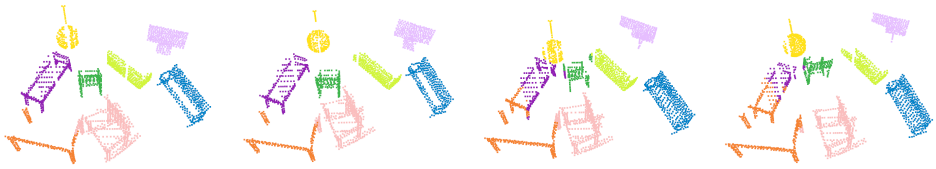
Frame 1	Frame 2	Frame 3	Frame 4
			

Figure 5.2: Visual Results on OGC-DRSV for inconsistent mask predictions.

planar patches that remain consistent across views, with each patch undergoing an approximately constant rigid motion over time. It formulates the matching of these patches and their motion as the minimization of an energy function, which optimizes over both continuous plane and motion parameters and the discrete pixel-to-plane assignments.

This chapter introduces a method to ensure instance consistency across sequences by linking predictions through keypoint-driven association. Unlike conventional approaches that depend on post-processing (e.g., clustering or association steps), the proposed end-to-end framework operates scan-to-scan, directly generating predictions for the input sequence while accommodating variable frame numbers.

5.1.1 Inconsistency between object mask across frames

According to the results in Fig. 5.2, inconsistencies are observed between frames. To investigate this further, a statistical analysis of prediction accuracy gaps within sequences is conducted. As shown in Fig. 5.3, significant inter-frame discrepancies arise in the absence of post-processing techniques like voting. This observation motivates the exploration of solutions to enforce consistency in object mask predictions across frame sequences. The problem necessitates addressing both temporal point-frame combination and robustness in segmentation.

Is multi-frame input more effective for object segmentation than using just two frames? The intuition behind this question is that continuous or, at the very least, multiple observations help us as

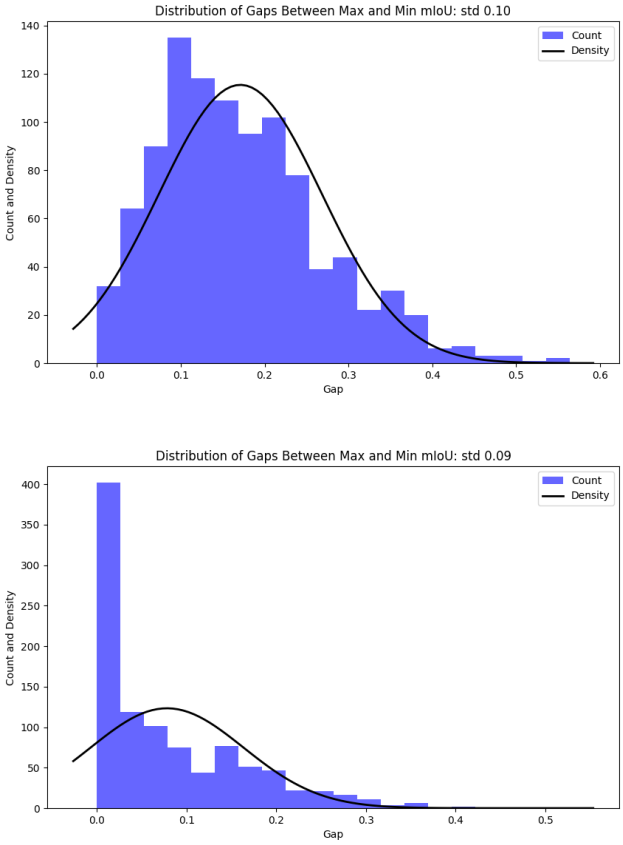


Figure 5.3: mIou gap between frames in a sequence on OGC-DRSV. Upper: before voting. Lower: after voting.

human beings perceive the world and make safer, more accurate decisions. This concept becomes particularly relevant in 3D vision, where robots and autonomous agents must predict the outcomes of potential obstacles or changes in their environment. For these systems, relying on just two frames often isn't enough to capture the full context, especially in dynamic settings.

The analysis on existing benchmarks supports this assumption. As shown in Fig. 5.4, the motion of an object becomes more distinct as more frames are detected, making segmentation of moving objects more accurate. This observation naturally raises the question: how does multi-frame perception impact the completeness of object shape? Indeed, multiple frames contribute to a more complete geometric representation of objects, aligning with the structure-from-motion (SfM) principle from

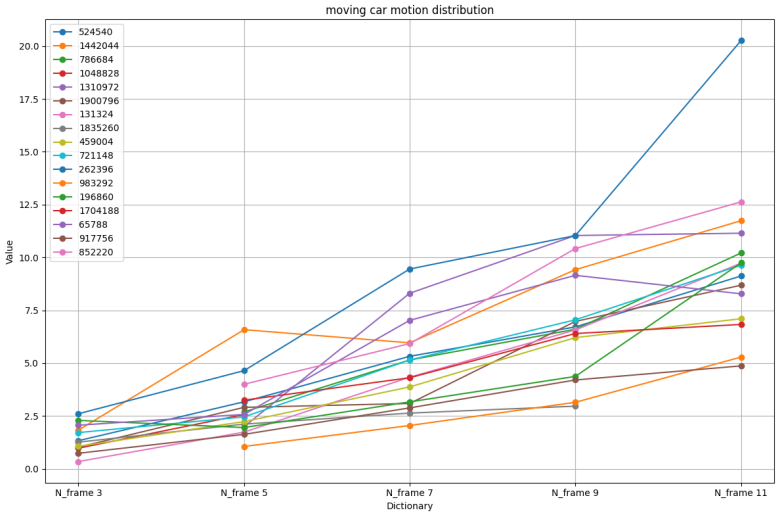


Figure 5.4: Vehicle motion distribution with different frame numbers in SemanticKITTI-Seq 08.

T	**AP** ↑	**PQ** ↑	**F1** ↑	**Pre** ↑	**Rec** ↑	**mIoU** ↑	**RI** ↑
0	87.4	76.2	83.4	75.9	92.5	85.5	95.7
1	88.1	78.1	84.9	78.2	92.7	86.0	96.0
2	88.3	78.5	85.2	79.1	92.3	85.8	96.0
3	88.4	78.6	85.3	79.4	92.1	85.7	96.0

Table 5.1: The OGC-DRSV dataset employs multi-frame co-segmentation, where the hyperparameter T controls the temporal window for consistency: adjacent frames within $[t-T, t+T]$ are leveraged to compute segmentation coherence for the anchor frame t. The baseline method ($T = 0$, OGC (Song and Yang 2022)) is trained once without object-aware optimization. Test set results demonstrate the impact of varying T on segmentation performance.

computer vision, which describes how the 3D structure of an object can be inferred from sequential 2D images. This chapter doesn't delve deeply into multi-view geometry. Instead, it focus on enhancing data for unsupervised object segmentation. The proposed approach is rooted in the Gestalt principles, which suggest that moving agents often appear in groups. This insight means that an anchor point cloud can effectively represent the movement of points within its vicinity. Building on this, this chapter explores the use of key points in multi-frame segmentation tasks. Specifically, key point masks for the input sequence are predicted,

which is then extrapolated to create a full point mask with linear computational complexity.

5.1.2 Difference to co-segmentation of multiple frames

In the study presented in OGC-PAMI (Song and Yang 2024), a multi-frame co-segmentation algorithm was introduced to enhance segmentation accuracy across temporal sequences. The core of this algorithm lies in harnessing scene flows to align estimated object segmentation masks across frames, enforcing geometric consistency and temporal coherence. Table 5.1 shows the co-segmentation algorithm's improvements in object segmentation across sequential frames. These results align with the principle of maintaining geometric consistency over time. The algorithm effectively ensures stable, coherent segmentation throughout the sequence. While effective, this approach relies heavily on precomputed scene flow vectors to guide mask alignment, which can propagate errors from inaccurate flow estimations.

By contrast, the proposed method is an end-to-end learning framework that balances temporal coherence with the flexibility to refine masks based on intra-frame geometric cues. The proposed approach mitigates error accumulation across frames without additional masks alignment operation.

5.1.3 Difference to co-part segmentation

Object co-segmentation is aimed at segmenting common objects from the background, whereas co-part segmentation focuses on jointly decomposing these common objects into semantically consistent parts across point clouds. In other words, object co-segmentation targets the entire object, while co-part segmentation is concerned with specific parts of the object. For example, object co-segmentation would aim to segment all cars in a scene, while co-part segmentation would be more interested in

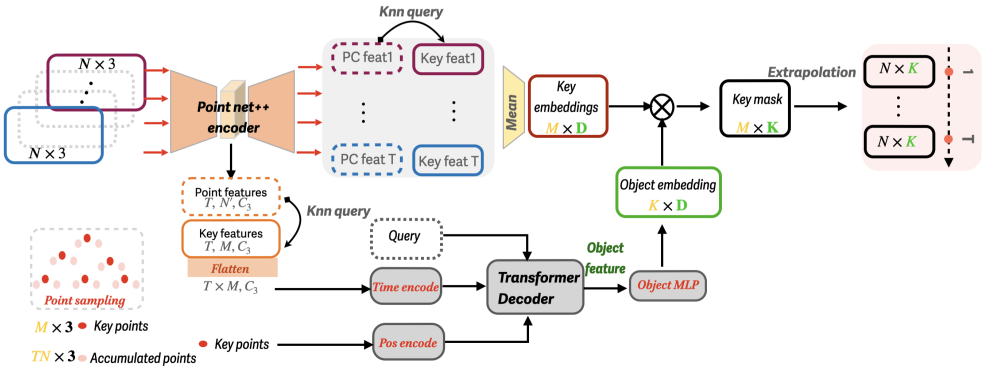


Figure 5.5: Architecture of the Proposed Multi-Frame Segmentation Network. The network processes 2–4 sequential point clouds as input and predicts a set of temporally consistent object masks. It comprises three core components: (1) a PointNet++ encoder for hierarchical feature extraction, (2) a transformer decoder to refine object queries using spatiotemporal context, and (3) a key feature query head via k nearest neighbor (knn) search. The per-point masks are extrapolated from the key point mask by leveraging spatial relationships.

segmenting the wheels, windows, and doors of the cars. Due to the difficulty caused by the absence of labels, (Umam et al. 2024) investigate the co-part segmentation task by two subnetworks, superpoint generation network (SG-Net) and part aggregation network (PA-Net), respectively. However, their target is static object segmentation.

5.2 Methods

This section provides a detailed description of the proposed object segmentation network. The whole pipeline is illustrated in Fig. 5.5. The point feature extraction module first generates point embeddings and their corresponding features. Subsequently, transformer decoders employ K learnable queries alongside key point features—enhanced with positional and temporal encodings—to compute K distinct object embeddings. Each object embedding is designed to encode the characteristics of a specific object within the input point cloud. Following the key point embedding query process, the key mask is derived by calculating the dot product similarity between each object embedding and the key

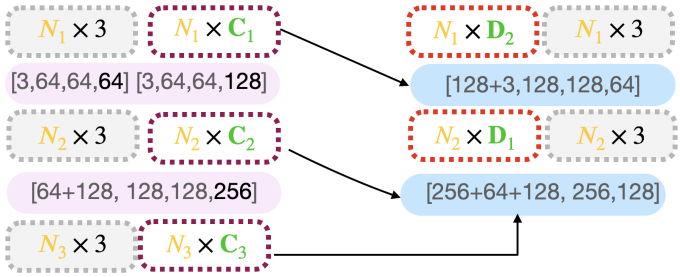


Figure 5.6: The detailed architecture of PointNet++ for OGC-DR/OGC-DRSV dataset.

point embeddings. This mask identifies regions of the point cloud associated with each detected object. The framework leverages scene flow vectors as a self-supervised signal, combining a dynamic consistency loss (derived from scene flow alignment) and a motion smoothness loss to enforce temporal coherence in predictions.

5.2.1 Pointnet++ backbone

PointNet++ (Qi et al. 2017b) is widely used as a feature backbone for processing point cloud data. Its hierarchical structure is leveraged to capture point features across multiple scales. By grouping points in progressively larger neighborhoods, PointNet++ enables the network to downsample the point cloud and learn features from coarse to dense resolutions. This hierarchical downsampling captures both local and global information, enabling the model to effectively represent complex geometries. At each level, the network aggregates features from increasingly larger regions, capturing fine-grained details in early layers and broader spatial structures in deeper layers. It is highly effective for various point cloud processing tasks, such as segmentation and classification. In our framework, PointNet++ serves as the backbone for point feature extraction, learning features from each point cloud frame simultaneously. The detailed layers of PointNet++ backbone for OGC-DR/OGC-DRSV is shown in Fig. 5.6.

Table. 5.2 lists the network layer parameters. The Set Abstraction

	OGC-DR / OGC-DRSV					KITTI-SF			
	level	s	k	r	c	s	k	r	c
SA	1-1	1/2	64	0.1 (0.05)	{3,64,64,64}	1/4	64	1.0	{3,32,32,32}
	1-2	1/2	64	0.2 (0.1)	{3,64,64,128}	1/4	64	2.0	{3,32,32,64}
	2	1/4	64	0.4 (0.2)	{192,128,128,256}	1/8	64	4.0	{96,64,64,128}
	3					1/16	64	8.0	{128,128,128,256}
FP	3	1/8				1/8			{384,128,128}
	2	1/2			{448,256,128}	1/4			{224,64,64}
	1	1			{131,128,128,64}	1			{67,64,64,64}
KP	1				{256,256}	1			{256,256}

Table 5.2: Comparison of configurations for OGC-DR / OGC-DRSV and KITTI-SF. In practice, the downsampling rate s and point neighborhood selection in the PointNet++ backbone are adapted to the point densities and sizes of different datasets. The parameter k determines the number of nearest neighbors sampled within a spherical region of radius r. Meanwhile, c specifies the input channel dimension of the first MLP layer and the output channel dimensions of subsequent layers in the MLP block.

(SA) modules employ a multi-scale grouping (MSG) strategy (Qi et al. 2017a), where levels 1-1 and 1-2 are combined and their outputs concatenated to capture hierarchical spatial features. In the Feature Propagation (FP) modules, multi-level point features generated by the SA modules are concatenated to form enriched input representations for upsampling and feature refinement. The Kernel Propagation (KP) Block—a non-parametric component within the key feature query head—maintains fixed input and output feature dimensions, which preserves structural relationships during feature propagation.

5.2.2 Point cloud accumulation

Scene flow predictions are utilized to accumulate a sequence of point clouds. In the proposed approach, the input sequence lacks one-to-one correspondences, making it challenging to directly superimpose one point cloud onto another. However, the flow can be utilized to accumulate these clouds through a process where the point cloud is first aligned to a target frame and then concatenated to form a consolidated set of points.

Algorithm 1 Accumulate Point Clouds According to Flow Vector

Require: `pcs` $\mathcal{U}_p$: $(T, N, 3)$
`flows` $\mathcal{F}_p$: $(T, N, 3)$
Ensure: A_p: $(T \times N, 3)$; $\mathcal{U}'_p$: $(T, N, 3)$
$T \leftarrow$ length of `pcs`
$\mathcal{U}'_p \leftarrow$ empty list
{Process each frame}
for $i \leftarrow 0$ to $T-2$ **do**
  $\boldsymbol{P'}_i \leftarrow \boldsymbol{P}_i + \mathcal{F}_i$
  for $j \leftarrow i+1$ to $T-2$ **do**
    `ids` $\leftarrow$ `find_knn(`$\boldsymbol{P'}_i$`,` $\mathcal{P}_j$`)`
    `nn_flow` $\leftarrow$ `index_flow(`$\mathcal{F}_j$`, ids)`
    $\boldsymbol{P'}_i \leftarrow \boldsymbol{P'}_i +$ `nn_flow`
  end for
  Append $\boldsymbol{P'}_i$ to $\mathcal{U}'_p$
end for
{Add the last frame}
Append $\mathcal{P}_{T-1}$ to $\mathcal{U}'_p$
$A_p \leftarrow$ `torch.cat(`$\mathcal{U}'_p$`, dim=0)`
return $A_p, \mathcal{U}'_p$

Let $\mathcal{U}_p = \{\boldsymbol{P}_1, \ldots, \boldsymbol{P}_T\}$ denotes T frames of point clouds in a scene and $\mathcal{U}'_p = \{\boldsymbol{P'}_1, \ldots, \boldsymbol{P'}_T\}$ is the aligned point cloud frames. Each frame of point cloud $\boldsymbol{P}_t = \{(x, y, z)_j\}_{j=1}^{N_t}$ contains point coordinates, where $t \in [1, , T]$ denotes the frame index. N_t denotes the number of points in a single frame. Additionally, we define $\mathcal{F}_p = \{\boldsymbol{F}_1, \ldots, \boldsymbol{F}_T\}$ as the scene flow vector extracted by the flow estimation network (Kittenplon et al. 2021). The accumulated point set, defined as A_p is generated through Alg. 1.

The flow is propagated through the sequence; the flow vector between the first and second frames is employed to align the initial frame with the subsequent one. Subsequently, once the first frame's points are identified within the second frame, the flow vector in the second frame is indexed to transition towards the subsequent frame. In this manner, each frame becomes aligned with the last frame in the sequence. The accumulated point cloud is then utilized to sample key points for the subsequent mask prediction process. The accumulated point sets and sampled key points are illustrated in Fig. 5.7, which demonstrates the ability of the FPS

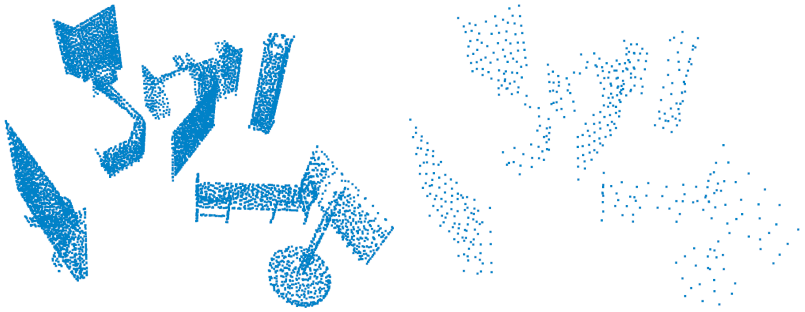


Figure 5.7: Dense accumulated point cloud (left) and sampled key points (right).

algorithm to preserve the geometric structure of objects.

5.2.3 Key points feature

Key points are represented as a shared anchor set across all frames in the sequence: $\mathcal{H} = \{h_1, \cdots, h_M\}$. These key points can be either real or virtual. Farthest Point Sampling (FPS) is employed to ensure optimal coverage of the original point distribution, with M denoting the number of key points. While a key point may not align precisely with specific spatial locations across frames, it serves as a critical component for temporal point mask prediction. Key embeddings are aggregated by independently searching for corresponding features in each frame's original point embeddings. This cross-frame aggregation captures subtle temporal variations, enriching the key embeddings with contextualized motion and structural dynamics. The resulting temporally informed embeddings enhance robustness and consistency, establishing reliable anchors for cross-sequence coherence. Key point features are computed via k-nearest neighbors (KNN) queries on the aligned point cloud. For each key point h_t^i, three nearest neighbors $(\widetilde{h}_t^{p_1}, \widetilde{h}_t^{p_2}, \widetilde{h}_t^{p_3})$ are sampled from $\boldsymbol{P'}_t$, and an inverse distance-weighted average is applied to their features $z_t^{p_k}$. The aggregated feature for key points $\mathcal{H}$ at frame t is defined as :

$$\mathcal{Z}_{Ht} = \mathcal{F}(\mathcal{Z}_{\boldsymbol{P'}_t}, \boldsymbol{P'}_t, \mathcal{H}). \tag{5.1}$$

$\mathcal{Z}_{\boldsymbol{P'}_t}$ is raw point feature vector at frame t, with corresponding point set represented by $\boldsymbol{P'}_t$. The feature query operation is as folllows:

$$\mathcal{Z}_t^i = \sum_{k=1}^{3} \frac{u_k \times z_t^{p_k}}{\sum_{k=1}^{K} u_k}, i \leq M \tag{5.2}$$

where u_k is distance based weight defined as:

$$u_k = \frac{1}{d\left(h_t^i, \widetilde{h}_t^{p_k}\right)} \tag{5.3}$$

The interpolated key point features from each frame $\{\mathcal{Z}_t^i \in \mathbb{R}^c \mid i = 1, \ldots, M\}$ is then stacked to fuse multi-frame feature. The final key feature is:

$$\mathcal{Z}_H = [\mathcal{Z}_{H1} \oplus \mathcal{Z}_{H2} \oplus \mathcal{Z}_{H3} \oplus \mathcal{Z}_{HT}] \in \mathbb{R}^{TM \times C}. \tag{5.4}$$

Time encode head. Following KNN-based key feature retrieval, a time encoding head integrates temporal information into each key point using a sinusoidal encoding scheme, akin to positional embeddings in sequence processing. This process augments each key point feature by appending a three-dimensional temporal feature, yielding an enriched representation $\mathcal{Z}'_H \in \mathbb{R}^{TM \times (C+3)}$. The sinusoidal encoding captures cyclical temporal patterns, embedding consistent contextual information about each key point's position in the sequence. Structured temporal differentiation enables the model to identify motion patterns and variations across frames effectively. By augmenting key features with temporal signals, the encoding enhances cross-frame coherence through improved tracking of key points over time. These temporally enriched features thereby provide a robust foundation for downstream tasks requiring dynamic pattern analysis across frames.

5.2.4 Maskformer decoder

The Transformer decoder initializes with K instance queries and iteratively refines them through L stacked decoder layers, producing a set of precise, context-aware instance embeddings. Each layer refines the

queries by cross-attending to key point features and performing self-attention to model inter-query relationships. The learnable queries act as Q (query vectors), while key point features serve as K (keys) and V (values) in the multi-head attention mechanism. Following the cross-attention step, the instance queries undergo self-attention, where the queries, keys, and values are computed from linear projections of the instance queries themselves. This self-attention facilitates communication between the instance queries, preventing multiple queries from focusing on the same object, which would otherwise lead to duplicate instance masks. An MLP layer is added to reduce the dimensionality of the object embeddings, aligning them with the dimensionality of the point embeddings produced by the PointNet++ backbone.

This dual-attention design enables the decoder to adaptively focus on geometrically salient regions while reasoning about instance-level interactions.

Types of Queries: Parametric query approaches (Cheng et al. 2022, Chen et al. 2024) learn both query features and positional encodings during training. This requires optimizing a fixed set of K queries to generalize across inference-time scene instances.

In contrast, methods like (Misra et al. 2021) employ non-parametric queries. Instead of training queries, they create them by sampling 3D points from the input (using a farthest-point sampling strategy). These queries start with zero features and only use the sampled points' positions for encoding. Non-parametric queries offer flexibility: the query count can vary between training and inference, enabling speed-performance trade-offs without retraining. However, the proposed framework showed no performance improvement with non-parametric queries. This result motivated the adoption of parametric queries in this work.

Mask prediction: Each (soft) binary mask is derived by computing the dot product between the object embedding and the embeddings of M key points within the scene. This operation generates raw scores for every

key point, indicating its affinity (or likelihood) of belonging to the k^{th} object. To interpret these scores as probabilities, a softmax activation function is applied across all objects for each key point, normalizing the scores to produce a probability distribution. This ensures that each key point has a soft probability of being assigned to different objects, with values that sum to 1 across all possible object classes. Per-point mask is extrapolated from the key point mask without extra learning layers.

5.2.5 Loss functions

5.2.5.1 Smooth loss

The computed smoothness loss gives us a scalar value that indicates how consistent or smooth the features are across neighboring points in the point clouds. The lower the loss, the smoother the point cloud features are, meaning that neighboring points have more similar features. Mathmatically, the smooth loss is defined as:

$$\ell_s = \frac{1}{N}\sum_{n=1}^{N}\left(\frac{1}{Q}\sum_{q=1}^{Q} d\left(\boldsymbol{o}_n, \boldsymbol{o}_{n_q}\right)\right) \tag{5.5}$$

where $\boldsymbol{o}_n \in \mathbb{R}^{1\times K}$ denotes the object assignment of center point p_n. The neighboring point set around the center point p_n is $\mathcal{Q}_n = \left\{\boldsymbol{p}_n^1, \cdots, \boldsymbol{p}_n^Q\right\}$ and $\boldsymbol{o}_{n_q}$ is mask of the q^{th} point. We choose $L1$ distance function during loss computation.

5.2.5.2 Dynamic loss

The dynamic loss is defined as:

$$\ell_d = \frac{1}{N}\sum_{n=1}^{N}\left\|\left(\sum_{k=1}^{K} o_{nk}\cdot\left(\boldsymbol{T}_k \circ \boldsymbol{p}_n\right)\right) - \left(\boldsymbol{p}_n + \boldsymbol{a}_n\right)\right\|_2 . \tag{5.6}$$

Each point will be assigned a mask $o_n \in (0, 1)$, here o_{nk} denotes the probability of point belongs to the k-th object. $T_k \in \mathcal{R}^{4\times 4}$ is estimated transformation matrix generated by Weighted Kabsch algorithm. The Weighted Kabsch algorithm is introduced in Appendix A.2.2. a_n represents the motion of point p_n. Given fixed motion estimation, our target

Method Category	Methods	AP ↑	PQ ↑	F1 ↑	Pre ↑	Rec ↑	mIoU ↑	RI ↑
Supervised Methods	OGC-sup (Song and Yang 2022)	90.7	82.6	87.6	83.7	92.0	89.2	97.7
	MBSE3-sup (Zhong et al. 2024)	92.8	86.9	91.0	88.8	93.2	91.2	98.7
	Ours-sup	91.7	85.1	89.6	86.6	92.9	90.3	98.1
Unsupervised Methods	TrajAffn (Ochs et al. 2013)	42.6	46.7	57.8	69.6	49.4	46.8	80.1
	SSC (Nunes and Demiris 2018)	74.5	79.2	84.2	92.5	77.3	74.6	91.5
	WardLinkage (Ward Jr 1963)	72.3	74.0	82.5	93.9	73.6	69.9	94.3
	DBSCAN (Ester et al. 1996)	73.9	76.0	81.6	85.8	77.8	74.7	91.5
	OGC (Song and Yang 2022)	92.3	85.1	89.4	85.6	93.6	90.8	97.8
	MBSE3 (Zhong et al. 2024)	93.9	87.0	91.1	87.0	95.6	92.4	98.1
	Ours	**94.7**	**88.7**	**92.1**	**88.7**	**95.9**	**93.5**	**98.3**

Table 5.3: Segmentation performance on OGC-DR. The proposed method outperforms all unsupervised baselines across eight evaluation metrics. Furthermore, its fully-supervised variant achieves performance competitive with state-of-the-art supervised approaches.

is to minimize inconsistency of mask predictions during training, the estimated object masks are expected to be better and better with this dynamic constraints. The total loss is a weighted combination of smooth loss and dynamic loss, we set the weight to 0.1 for smooth loss and 10 for dynamic loss.

The proposed method can also be trained in a fully supervised manner. Let $\hat{o}_n$ denotes the predicted object mask and o_n denotes the ground truth label, cross entropy loss is computed as:

$$\ell_{ce}(o_n, \hat{o}_n) = -o_n \log(\hat{o}_n) - (1 - o_n) \log(1 - \hat{o}_n). \tag{5.7}$$

5.3 Experiments

The proposed method is evaluated on the following datasets:

- DynamicRoom synthetic dataset: object segmentation of indoor scenes
- KITTI-SF dataset: object segmentation of real-world outdoor scenes

5.3.1 Training details

The initial learning rate is established at $1.0 \times 10^{-4}$and undergoes a decay at a rate of 0.7, with a minimum threshold set at 1.0×10^{-5}. The

Method Category	Methods	AP ↑	PQ ↑	F1 ↑	Pre ↑	Rec ↑	mIoU ↑	RI ↑
Supervised Methods	OGC-sup (Song and Yang 2022)	86.3	78.8	85.0	82.2	88.0	83.9	97.1
	MBSE3-sup (Zhong et al. 2024)	89.3	82.6	87.9	85.5	90.4	86.6	97.9
	Ours-sup	**90.2**	**84.7**	**89.5**	**87.6**	**91.4**	**88.7**	97.7
Unsupervised Methods	TrajAffn (Ochs et al. 2013)	39.3	43.8	54.8	63.0	48.4	45.9	77.7
	SSC (Nunes and Demiris 2018)	70.3	75.4	81.5	89.6	74.7	70.8	91.3
	WardLinkage (Ward Jr 1963)	69.8	71.6	80.5	91.8	71.7	67.2	93.3
	DBSCAN (Ester et al. 1996)	71.9	76.3	81.8	79.1	84.8	80.1	93.5
	OGC (Song and Yang 2022)	86.8	77.0	83.9	77.7	91.2	84.8	95.4
	MBSE3 (Zhong et al. 2024)	88.1	80.0	86.1	80.8	92.2	**86.7**	**96.6**
	Ours	**88.1**	78.4	85.1	78.1	**93.5**	85.7	96.3

Table 5.4: Segmentation performance on OGC-DRSV. Minor performance differences (PQ 80.0% *vs* 78.4%) arise between MBSE3 and the proposed method. This discrepancy stems from MBSE3 updating flow vectors during network training, whereas the proposed method employs fixed flow vectors.

momentum for batch normalization is configured to 0.9, which influences the stabilization of the batch normalization metrics. The decay step is established at 2.0×10^5, dictating how often the learning rate is reduced.

During the training of segmentation network, the Adam optimizer is used. The learning rate is 0.001. The epochs for training on OGC-DR/KITTI-SF is 150/400 epochs respectively.

The batch size is set as 4/2 on each dataset to fill in the whole memory of a single RTX3090 GPU. The smooth loss is enabled after first 4000/400 sampleds on OGC-DR/KITTI-SF datasets. Data augmentation is introduced in fully supervised training on KITTI-SF dataset, which enhance the generalization ability of our network. The details of data augmentation is illustrated in Appendix A.2.1.

5.3.2 Results on OGC-DR and OGC-DRSV

As presented in Table. 5.3, the proposed method outperforms all classical unsupervised methods including the clustering based and the motion segmentation based methods on OGC-DR. Fig. 5.8 shows qualitative results. The proposed approach generates comprehensive object masks with minimal over-segmentation.

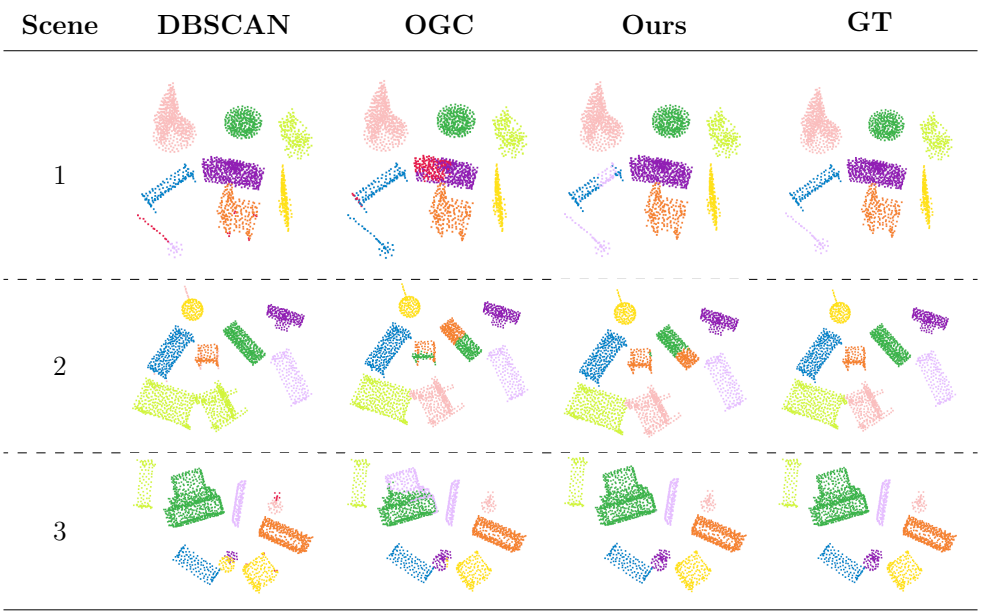


Figure 5.8: Visual Results on OGC-DR. We have selected three single scenes. Our method is compared with DBSCAN and OGC. For the best clarity, view these images in zoomed mode.

On the single-view OGC-DRSV dataset, the proposed method demonstrates superior performance and robustness to incomplete point clouds, as shown in Table. 5.4. Compared to the baseline OGC (Song and Yang 2022), the method achieves an AP score of 88.1 (+1.3 improvement) without post-processing or iterative training. This contrasts with OGC, which requires an object-aware ICP algorithm for flow refinement and an additional training round to reach an AP of 86.8.

A visual comparison of mask predictions across three representative scenes is provided in Fig. 5.9, evaluated under the challenging single-view OGC-DRSV benchmark. While the framework achieves reasonable segmentation accuracy overall, Scene 1 reveals a limitation where two adjacent objects with overlapping geometries are erroneously assigned the same mask, likely due to insufficient feature disentanglement in regions of high similarity. In contrast, Scenes 2 and 3 demonstrate robust temporal consistency, with mask predictions remaining coherent across all four frames despite viewpoint shifts and partial occlusions. This disparity highlights the method's sensitivity to object interaction complexity

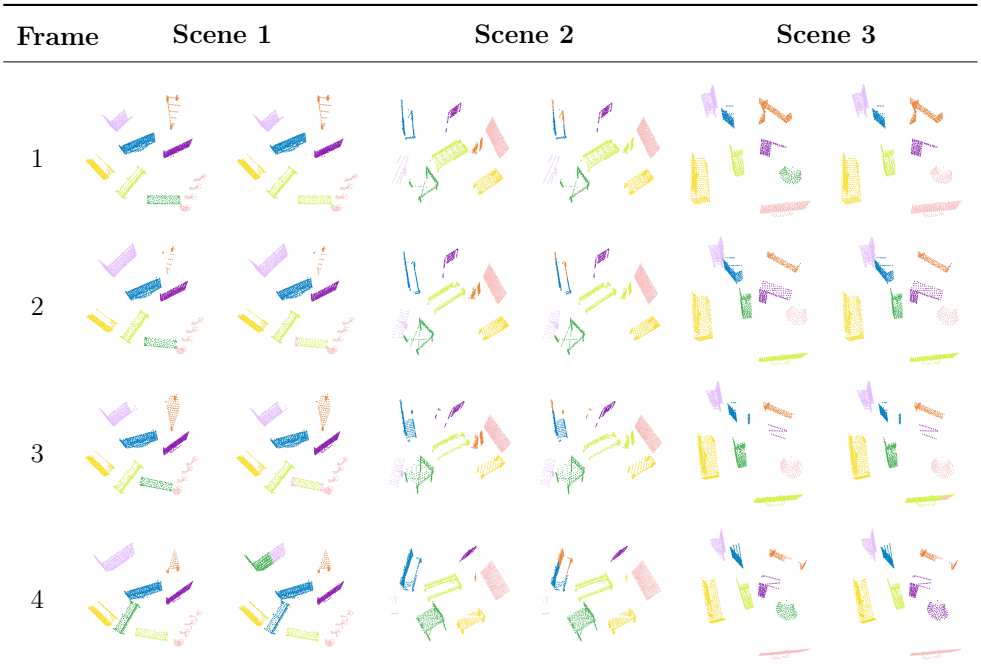


Figure 5.9: Visual Results on OGC-DRSV. We have selected three sequences, with each containing four frames. For the best clarity, view these images in zoomed mode. The left subfigure presents the ground truth (GT), while the right subfigure is our prediction.

Method Category	Methods	AP ↑	PQ ↑	F1 ↑	Pre ↑	Rec ↑	mIoU ↑	RI ↑
Supervised Methods	OGC-sup (Song and Yang 2022)	62.4	52.7	65.1	63.4	67.0	67.3	95.0
	MBSE3-sup (Zhong et al. 2024)	65.1	56.3	68.6	69.4	67.8	69.5	95.7
	Ours-sup	**67.5**	**58.3**	**72.2**	**74.7**	**69.9**	68.6	95.3
Unsupervised Methods	TrajAffn (Ochs et al. 2013)	24.0	30.2	43.2	37.6	50.8	48.1	58.5
	SSC (Nunes and Demiris 2018)	12.5	20.4	28.4	22.8	37.6	41.5	48.9
	WardLinkage (Ward Jr 1963)	25.0	16.3	22.9	13.7	**69.8**	60.5	44.9
	DBSCAN (Ester et al. 1996)	13.4	22.8	32.6	26.7	42.0	42.6	55.3
	Kernel-opt (Chapter 4)	25.2	**34.9**	**36.8**	**96.2**	24.7	23.3	**87.0**
	OGC (Song and Yang 2022)	36.0	24.6	35.4	26.4	53.8	53.7	57.8
	Ours	**36.5**	23.0	33.7	24.8	52.7	51.9	56.2

Table 5.5: Segmentation performance on KITTI-SF. The proposed supervised framework use key point loss only. In unsupervised setting, the results of OGC are collected from the first round training for a pair comparison.

but underscores its reliability in scenarios with clear spatial or motion distinctions.

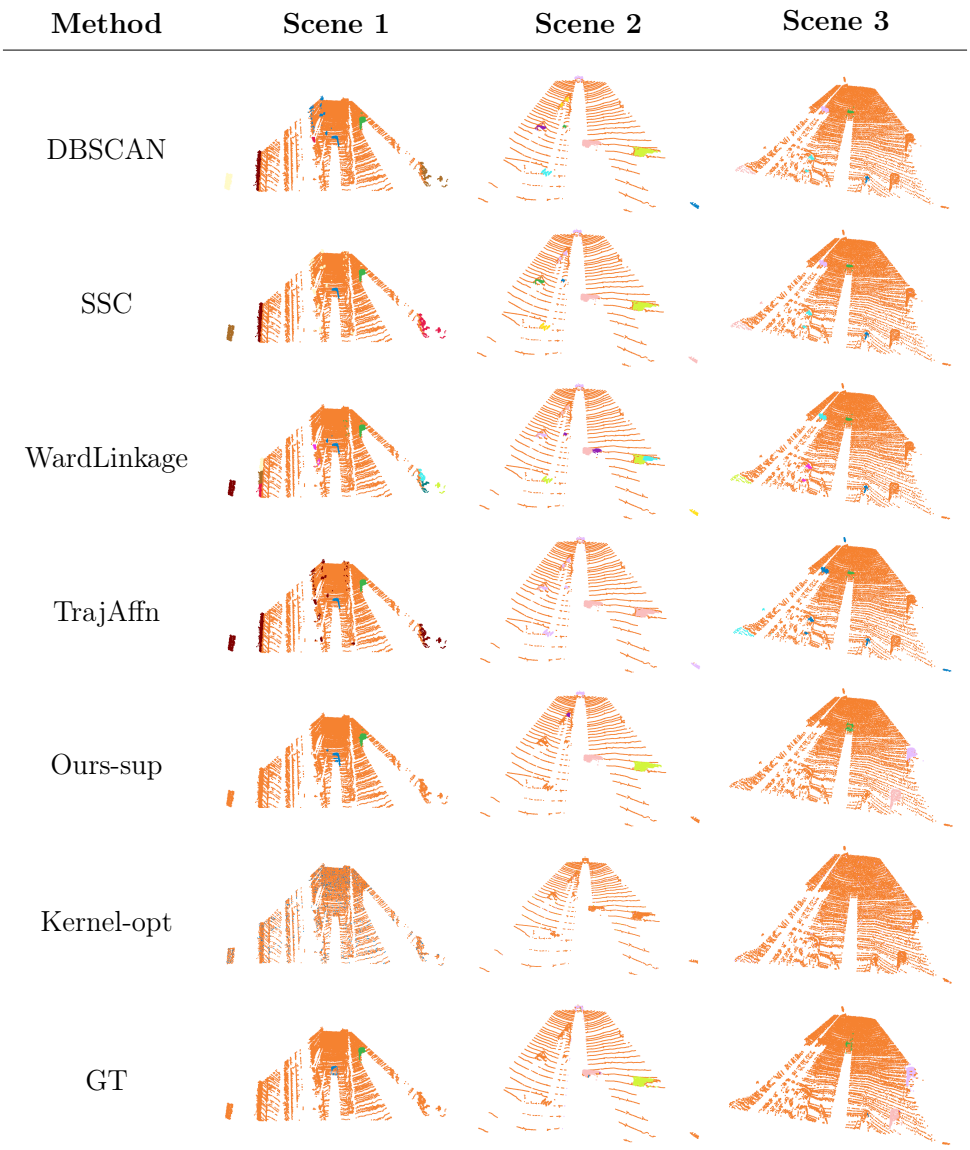


Figure 5.10: Visual Results on KITTI-SF. Three different scenes are selected to compare different methods. For the best clarity, view these images in zoomed mode. The results labeled Kernel-opt are generated using the algorithm proposed in Chapter 4.

5.3.3 Results on KITTI-SF

The proposed framework, shown in Fig. 5.5, is trained on the KITTI-SF dataset in a fully supervised manner. During experiments, it was observed that positional encoding did not enhance performance; in fact, disabling it yielded better results. Consequently, positional encoding is omitted in the fully supervised training experiments on the KITTI-SF

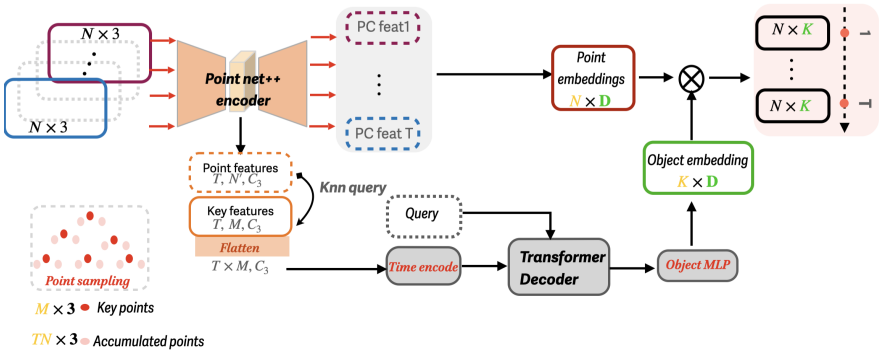


Figure 5.11: Plain model where the extrapolation of key point mask is omitted.

dataset. A visual comparison between the proposed method and other baselines is illustrated in Fig. 5.10. Quantitative results are presented in Table 5.5, where the proposed method outperforms baseline methods in the fully supervised setting. Due to limitations of the KITTI-SF dataset, multi-frame (more than two frames) joint learning segmentation could not be performed. However, the proposed unsupervised network can be effectively trained on the dataset. Even after the first round of training, the proposed method outperformed classical segmentation algorithms.

5.3.4 Pilot Studies

We conduct experiments on OGC-DRSV to verify the generalizability of the position encoding in MaskFormer Head and our mask extrapolation head.

1. Can the position encoding help improve the segmentation?

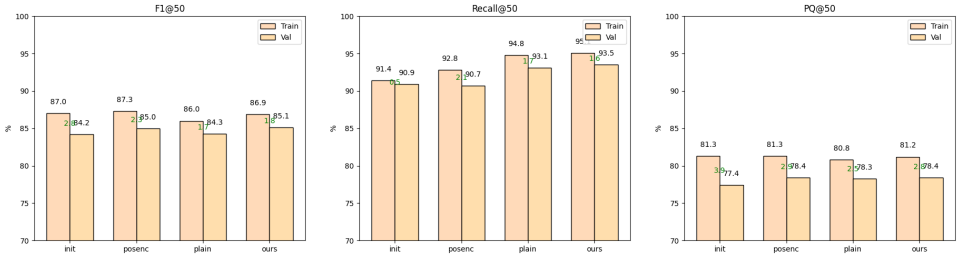
As demonstrated in Section 5.2.4, non-parametric queries outperform parametric queries in the proposed framework. To further validate design choices, this section presents ablation studies on positional encoding configurations. Two variants are compared: (1) **No Position Encoding**: The model operates solely on raw point features and key points, excluding explicit spatial cues. (2) **Position-Enhanced**: The model incorporates a positional encoding module that explicitly models spatial relationships both intra-frame and inter-frame.

Results and Discussion: Models with position encoding consistently achieved higher segmentation quality, improving Precision and Recall metrics by capturing positional consistency across frames.

2. Role of Key Mask Extrapolation for Per-Frame Mask Prediction

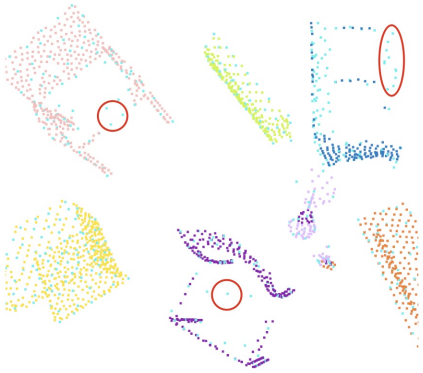
In the primary framework, key mask extrapolation is utilized to provide context for mask prediction in each frame by guiding segmentation based on key points extracted from the entire sequence. Here, we assess whether this extrapolation step is essential by creating a variant that computes per-frame masks independently using point embeddings without key mask guidance. A variant of the main framework is proposed in which point embeddings from a sequence of frames are used to compute per-point masks independently, without extrapolating from key masks. The workflow is illustrated in Fig. 5.11. We compare segmentation metrics under an unsupervised setting between this variant and the main framework in Fig. 5.5. Results in Table. 5.6 show a slight improvement in PQ and F1 scores, while the Pre score demonstrates less favorable outcomes compared to the baseline A1. This is expected, as the key mask, though aggregated from sequence features, may not capture the full range of details. Using per-frame features enables richer, more detailed information for each specific frame. The primary motivation is that key points can guide mask prediction for each frame. To accomplish this, the proposed method learns key features by leveraging point features extracted from each frame individually. Once the key point mask is predicted, a per-point mask can be extrapolated based on pointwise distances. A comparison on this group of pilot study is shown in Fig. 5.12 with a focus on Recall, PQ, and F1 score.

Theoretical Insight: Key mask extrapolation facilitates mask consistency across temporal sequences by using distance-based extrapolation from key points. This guides the model in balancing frame-specific de-

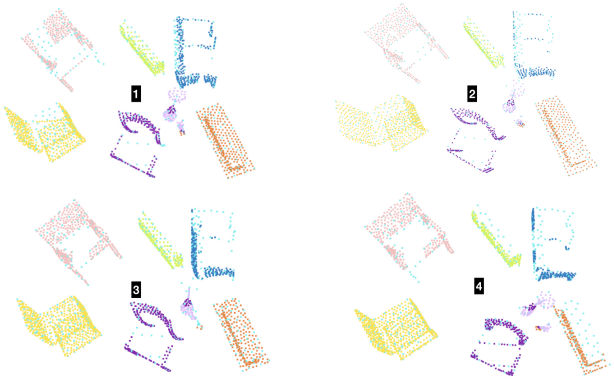


(a) Prec@50 (b) Recall@50 (c) F1@50

Figure 5.12: Comparison of models across different metrics discussed in Section 5.3.4. In each subfigure, model without posenc, model with posenc, plain model, plain model with posenc are compared. The performance drop from training set to validation set is also illustrated.



(a) Zoom-in view of frame two. Red markers indicate key points that are distant from each aligned point cloud.



(b) A sequence from OGC-DRSV.

Figure 5.13: Key points and their alignment across frames in the OGC-DRSV dataset.

tails with sequence-level coherence, crucial in scenes with occlusions or partially visible objects.

Settings	AP	PQ	F1	Pre	Rec	mIoU	RI
Model-A1	84.6	77.4	84.2	79.9	90.0	85.0	96.3
Model-A2	87.5	78.4	85.0	80.0	90.9	85.9	96.2
Model-A3	88.7	78.3	84.8	77.9	93.1	87.5	96.3

Table 5.6: Performance metrics for multi-frame segmentation on OGC-DRSV dataet; A1 is the results of using parametric queries (without position encoding). A2 is the results of using position encoding in the MaskFormer head. A3 is the results of using plain model to predict per-point mask.

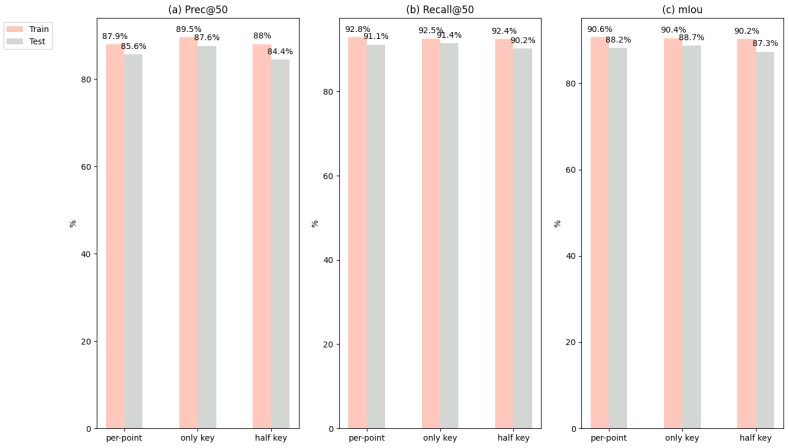


Figure 5.14: Comparison on usage of key point loss. In the accompanying subfigures, pink-colored bars represent training set performance metrics, while grey-colored bars correspond to testing set results. All models are trained in a fully-supervised manner, with the sole variation being the loss computation strategy.

3. Evaluation of Key Loss Alone in Model Training

The impact of using only key loss is also analyzed in this section, which penalizes discrepancies between predicted and true positions of key points rather than requiring alignment of every point. This approach is particularly relevant for single-view datasets, where each frame may lack full object information due to occlusions or limited viewpoints. Consequently, key points may be positioned far from aligned points in certain frame. As shown in Fig. 5.13, key points and aligned points are not fully overlapping because the input sequence is unevenly distributed. This sit-

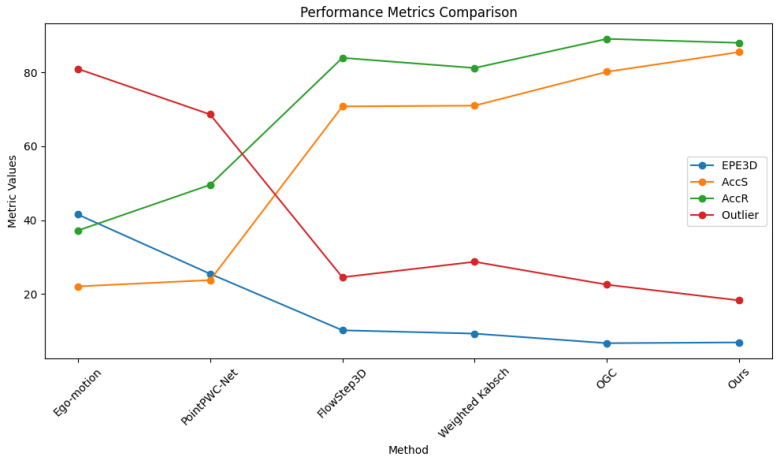


Figure 5.15: Scene flow estimation on the KITTI-SF dataset. The segmentation masks are used to enhance flow estimation through the object-aware ICP algorithm introduced in OGC. Compared to baseline methods, the proposed approach achieves the highest improvement in flow quality, demonstrating superior accuracy and reduced EPE3D.

uation reflects real-life scenarios, where certain objects may be missing from specific frames due to rapid movement.

Setup: Ablation study is performed with a fully-supervised model under three settings: key loss only, weighted combination of key loss (0.5) and per-point loss (0.5), and per-point loss only. All compared models incorporate both position encoding and time encoding. Notably, the model optimized solely with key loss is still able to enforce per-point mask alignment, despite being optimized exclusively with key points. As shown in Fig. 5.14, the model that only with key point loss achieved best results on Precision and Recall, as well as mIou.

5.3.5 Flow improvement

Once the object segmentation is obtained, the scene flow quality can be further improved using an iterative optimization algorithm. Given well-trained segmentation models on KITTI-SF dataset, the object-aware ICP algorithm (Song and Yang 2022) correct the inconsistency in flows and lead to a larger improvement than fine-tuning flowstep3d (Kittenplon et al. 2021) model. The comparison is shown in Fig. 5.15. According

to the metric value, the proposed method performs better compared to other baseline methods.

5.4 Concluding remarks

Contributions. This chapter presents an unsupervised learning framework that incorporates geometry consistency and motion pattern consistency to enable multi-frame object segmentation. This method utilizes shared key point mask across frames, which allows a more flexible input of dynamic sequence. The representational power of key point features is significantly enhanced by a simple time encoding head. The segmentation results can then be used to improve the quality of scene flow estimation.
Limitations. The main limitation of the proposed approach lies in its generalization ability to real-world datasets. To address this, one potential solution is to explore incorporating sparse correspondences as part of the input, a technique commonly used in computer graphics applications, which could facilitate automatic shape alignment. Additionally, the fusion of sequence feature is worth-exploring to further improve the capability of key points in the accumulated point set. This chapter discuss the usage of mean pooling to implement multi-frame feature fusion. However, mean pooling is relatively simple and could lead to information loss. This point feature fusion could be further improved by more advanced modules, such as transformer with cross attention, etc.

6 Conclusion and Future Work

6.1 Recapitulation of core contributions

The overall purpose of this thesis is to empower dynamic scene understanding by leveraging the compositional structure of scenes. Specifically, it explores two main sub problems: how to effectively estimate scene flow and how to use observed motion patterns to segment 3D objects.

In Chapter 2, a comprehensive comparison and in-depth analysis of recent deep learning methods for scene flow estimation from 2019 to 2024 is presented, covering supervised, weakly-supervised, and self-supervised approaches. This thesis systematically compares current approaches for estimating scene flow according to their learning strategies. Additionally, this thesis reviews methods for 3D object segmentation on both static input and dynamic input.

In Chapter 3, a point Transformer architecture is integrated with point-voxel correlation field to estimate scene flow effectively. Scene flow estimation is a high-dimensional and computationally intensive task. To address this challenge, this thesis employs a deep learning architecture that balances efficiency (fast feature aggregation) and effectiveness (sta-

ble flow estimation). An enhanced Point Transformer is utilized to extract point features efficiently while preserving a global understanding of the scene context. Additionally, global motion aggregation further enhances the effectiveness of the point-voxel correlation module. Evaluation on synthetic dataset: FlyingThings3D, and real dataset: KITTI, demonstrates its effectiveness and generalization ability.

In Chapter 4, a clustering-free framework for 3D object segmentation is presented. This chapter explores a compact algorithm which only needs to optimize key point masks rather than full point masks. Then the key point masks are propagated to full points via kernel function without any direct object-level labels. The proposed method can effectively address the under-segmentation and over segmentation problems without relying on clustering methods or object detectors. Finally, extensive experiments are conducted to demonstrates the competitive results on object segmentation benchmarks.

Chapter 5 delves into unsupervised object segmentation with a focus on multi-frame segmentation. The proposed method is flexible to input frame numbers and end-to-end trainable, which utilizes shared key point mask across frames. The representational power of key point features is significantly enhanced by a simple time encoding head. Experimental results on both indoor dataset and outdoor dataset show the effectiveness of the method. Furthermore, the segmentation results can be used to improve the quality of scene flow estimation.

6.2 Conclusion and Future Perspectives

The directions of research covered in this thesis are open-ended, and many additional experiments and extensions are worth exploring. Despite being limited in scope, this thesis opens up broader, long-term research directions for the problems covered.

Label-free Efficient Scene Flow Estimation Models. In the task of scene flow estimation, this thesis investigates an effective mechanism to address feature extraction and the integration of local and global information between points. Current methods still require annotation intervention. In future work, research can focus on weakly supervised and self-supervised methods, specifically designing a unified framework applicable to multiple datasets. In the area of scene flow estimation, while fast and highly accurate solutions exist, narrowing the gap between efficiency and state-of-the-art performance in real-time systems for mobile platforms remains a significant challenge. Pioneering works (Jund et al. 2021, Vedder et al. 2023, Li and Lucey 2024) demonstrate progress toward lightweight, mobile-compatible systems, though computational bottlenecks persist. For instance, the dual point-voxel architecture introduced here incurs inference latency, hindering edge-device applicability. Future research should prioritize end-to-end frameworks that unify weakly or self-supervised paradigms—reducing annotation dependence—while maintaining computational efficiency across diverse datasets. Additionally, enhancing scene flow datasets for long-term motion and 3D shape reconstruction could further alleviate supervision requirements. As hardware constraints ease, integrating these advances may accelerate adoption of high-accuracy, real-time label-free models for resource-limited platforms.

Memory Optimization in 3D Point Cloud Segmentation. The memory footprint of 3D point clouds in autonomous driving arises from the need to process millions of points per frame, compounded by LiDAR's high sampling rates. This not strains hardware but also limits the scalability of algorithms. For instance, the segmentation pipeline in this thesis struggles with real-time performance due to iterative optimization steps and redundant computations. To mitigate this, dataset-specific kernel functions could dynamically adapt to regional geometric characteristics

(e.g., road surfaces vs. pedestrians), optimizing resource allocation during inference. As mentioned in previous chapters, multiple observations is more effective in object segmentation than single observation. Thus, exploring the fusion of sequence features can further improve the capabilities of key points within the accumulated point set. This thesis employs mean pooling to fuse features and obtain key point embeddings. However, mean pooling is a relatively simple method and may result in information loss. Advanced fusion strategies, such as cross-attention mechanisms in transformer architectures, could selectively preserve discriminative features across frames. These mechanisms would allow the model to weight features based on their relevance (e.g., prioritizing points with high motion entropy), thereby improving segmentation precision without proportional computational cost increases.

Enhancing Feature Propagation via Ground-Aware Preprocessing. A critical limitation of nearest neighbor propagation in 3D point cloud processing is its susceptibility to dominance by pervasive ground points in autonomous driving datasets. While the method in Chapter 5 efficiently transfers features to key regions, its indiscriminate spatial aggregation often dilutes foreground object features (e.g., vehicles, pedestrians) due to the overwhelming density of ground-plane points. To address this, ground plane fitting can be integrated as a preprocessing step to isolate and suppress ground points before feature propagation. For instance, leveraging techniques like (Li et al. 2017), which robustly segment ground surfaces using iterative plane estimation, could enable foreground-focused feature aggregation. By masking or downweighting ground points early in the pipeline, the propagation mechanism would prioritize dynamic or semantically critical regions, improving segmentation accuracy for real-world dataset.

Toward Integrated Perception: Meta-Supervision via Scene Flow for Unified 3D Understand. In contemporary intelligent perception

systems, critical tasks such as motion estimation, object detection, segmentation, and tracking are often addressed through modular, isolated frameworks. While effective, this fragmented approach limits holistic scene understanding and computational efficiency. This thesis pioneers a meta-supervision paradigm, where scene flow—the 3D motion field of points between consecutive frames—serves as a supervisory signal for training 3D object segmentation models. By leveraging scene flow's inherent spatiotemporal consistency, the method circumvents reliance on large-scale manual annotations, making it particularly advantageous in domains where dense labels are impractical (e.g., long-range LiDAR sequences or rare object categories).

A unified network is expected to jointly predict scene flow, segmentation, and detection in a single forward pass, minimizing computational redundancy. To enhance segmentation under partial observations, deformable shape models should be incorporated, allowing the network to adapt to dynamic object interactions, such as the articulation of a turning truck's trailer. Additionally, the supervisory role of scene flow can be extended to other modalities, such as radar-camera fusion, or to tasks like trajectory prediction, leveraging motion as a universal prior for more robust and comprehensive perception.

A Appendix

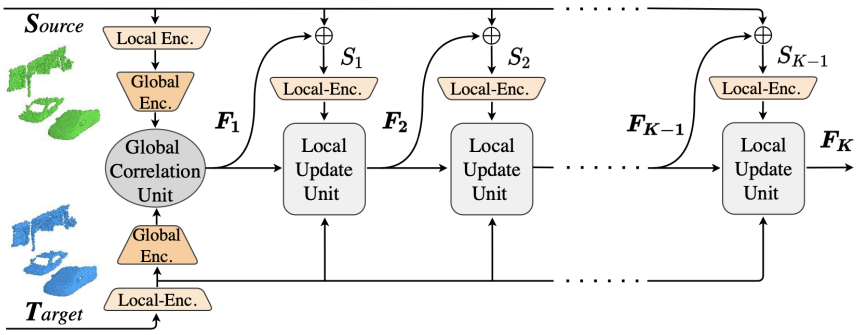


Figure A.1: The architecture of flow prediction in FlowStep3d (Kittenplon et al. 2021).

A.1 Self-Supervised Scene Flow Estimator for object segmentation

FlowStep3D (Kittenplon et al. 2021) is used in Chapter 4 and Chapter 5 to produce flow estimations. The method uses a PointNet++ backbone to extract per-point features from each of the two point cloud frames independently. It leverage a recurrent architecture, GRU to refine the scene flow predictions iteratively. The overall structure of flowstep3d is shown in Fig. A.1.

A.2 Object Segmentation

This section provides detailed algorithms used in object segmentation and introduce data augmentation used on KITTI-SF dataset.

A.2.1 DBSCAN algorithm

DBSCAN stands for "Density-Based Spatial Clustering of Applications with Noise". It is an unsupervised clustering-based segmentation method. Hence, it is a typical baseline method for unsupervised segmentation. Comparisons about this method and our method can be found in Sec. 5.3. The detail of DBSCAN is as the following:

Algorithm 2 DBSCAN Algorithm

Compute neighbors $N_\varepsilon(p) = \{q \mid d(p,q) \leq \varepsilon\}$ for each point p and identify core points; // *Identify core points*
Join neighboring core points into clusters $C_i = \{p \mid p$ is connected to core points in $C_i\}$; // *Assign core points*
for each non-core point p **do**
  if p has a neighboring core point **then**
    Assign p to a neighboring core point's cluster; // *Assign border points*
  else
    Mark p as noise; // *Assign noise points*
  end if
end for

There are two hyper parameters to finetune the results of DBSCAN: eps ε and minpoint.

A.2.2 The Weighted Kabsch Algorithm

The weighted Kabsch algorithm is a method for aligning two sets of points in $\mathbb{R}^3$ in a way that minimizes the weighted root mean square deviation (RMSD). It is commonly used in bioinformatics, computer vision, and other fields where 3D shape comparison is important.

Let $\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_N\}$ and $\mathbf{Q} = \{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_N\}$ be two sets of points in $\mathbb{R}^3$, where $\mathbf{p}_i = (x_i, y_i, z_i)$ and $\mathbf{q}_i = (u_i, v_i, w_i)$ for $i =$

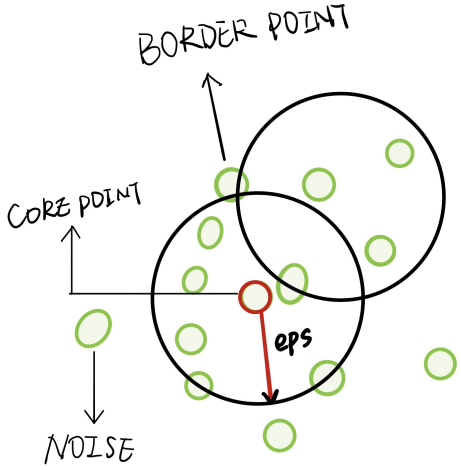


Figure A.2: DBSCAN algorithm.

$1, 2, \ldots, N$. Furthermore, let $\{w_1, w_2, \ldots, w_N\}$ be a set of corresponding weights associated with each point pair.

The weighted Kabsch algorithm proceeds as follows:

1. Compute the weighted centroids of $\mathbf{P}$ and $\mathbf{Q}$:

$$\bar{\mathbf{p}} = \frac{1}{\sum_{i=1}^{N} w_i} \sum_{i=1}^{N} w_i \mathbf{p}_i,$$
$$\bar{\mathbf{q}} = \frac{1}{\sum_{i=1}^{N} w_i} \sum_{i=1}^{N} w_i \mathbf{q}_i.$$

2. Translate the point sets so that their weighted centroids are at the origin:

$$\mathbf{p}'_i = \mathbf{p}_i - \bar{\mathbf{p}},$$
$$\mathbf{q}'_i = \mathbf{q}_i - \bar{\mathbf{q}}.$$

3. Compute the weighted covariance matrix:

$$\mathbf{H} = \sum_{i=1}^{N} w_i \mathbf{p}'_i \otimes \mathbf{q}'_i,$$

where $\otimes$ denotes the outer product.

4. Compute the optimal rotation matrix $\mathbf{R}$ by finding the best orthogonal matrix that minimizes the Frobenius norm of $\mathbf{R} - \mathbf{H}$. This is typically done using the singular value decomposition (SVD) of $\mathbf{H}$.

5. Apply the rotation matrix $\mathbf{R}$ to the translated point set $\mathbf{P}'$:

$$\mathbf{P}_{\text{aligned}} = \{\mathbf{R}\mathbf{p}'_1, \mathbf{R}\mathbf{p}'_2, \ldots, \mathbf{R}\mathbf{p}'_N\}.$$

The weighted Kabsch algorithm thus provides a transformation that aligns the point set $\mathbf{P}$ to $\mathbf{Q}$ in a weighted least-squares sense.

A.2.3 Data augmentation

In the fully-supervised object segmentation settings, OGC (Song and Yang 2022) introduce a geometry invariance loss to increase the generalization ability of segmentation network. In the training set, we apply data augmentation to increase the model's robustness to various transformations and improve its generalization to real-world scenarios. The augmentation pipeline includes translation and rotation designed to simulate realistic variations in scale, rotation, and position that a model might encounter in deployment.

- Scaling:

  We apply a scaling transformation where the scale factor is uniformly sampled from a range of 0.95 to 1.05. This means each point cloud can be slightly shrunk or enlarged by up to 5%, simulating variations in object sizes or distances from the sensor. This scaling helps the model to generalize better across different scales and prepares it for objects or scenes that may appear slightly larger or smaller than those seen during training.

- Rotation:

  The point cloud is randomly rotated around the vertical (y-axis), with the rotation angle sampled from -180° to 180°. This rotation mimics changes in orientation that the model might encounter in different viewpoints or scene layouts. By randomly rotating the point cloud around the y-axis, the model learns to recognize patterns in the data regardless of their orientation, which is especially

important in applications where the scene or objects can appear from various angles.

- Translation (specific to KITTI-SF dataset):

  For the KITTI-SF dataset, we add an additional translation step. In the x and z directions, translations are uniformly sampled from -1 to 1 units, allowing the entire point cloud to shift horizontally or depth-wise by up to 1 unit in either direction. This simulates slight shifts in the vehicle's or sensor's lateral or longitudinal position between frames. In the y (vertical) direction, translations are sampled from -0.1 to 0.1 units, adding a smaller, controlled vertical shift. This slight vertical translation accounts for minor variations in sensor height or terrain elevation changes. These translations make the model more robust to positional variations, preparing it for real-world scenes where the sensor might be in slightly different positions across frames.

Bibliography

Ahmed, S. M. and Chew, C. M., 2020. Density-based clustering for 3d object detection in point clouds. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10608–10617.

Altschuler, J., Niles-Weed, J. and Rigollet, P., 2017. Near-linear time approximation algorithms for optimal transport via sinkhorn iteration. *Advances in neural information processing systems*, 30.

Aygun, M., Osep, A., Weber, M., Maximov, M., Stachniss, C., Behley, J. and Leal-Taixé, L., 2021. 4d panoptic lidar segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5527–5537.

Battrawy, R., Schuster, R., Mahani, M.-A. and Stricker, D., 2022. Rms-flownet: Efficient and robust multi-scale scene flow estimation for large-scale point clouds. Institute of Electrical and Electronics Engineers (IEEE), 883–889.

Baur, S., Emmerichs, D., Moosmann, F., Pinggera, P., Ommer, B. and Geiger, A., 2021a. Slim: Self-supervised lidar scene flow and motion segmentation. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Los Alamitos, CA, USA: IEEE Computer Society, 13106–13116.

Baur, S. A., Emmerichs, D. J., Moosmann, F., Pinggera, P., Ommer, B. and Geiger, A., 2021b. Slim: Self-supervised lidar scene flow and mo-

tion segmentation. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 13126–13136.

Baur, S. A., Moosmann, F., Wirges, S. and Rist, C. B., 2019. Real-time 3d lidar flow for autonomous vehicles. *2019 IEEE Intelligent Vehicles Symposium (IV)*, France: IEEE, 1288–1295.

Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C. and Gall, J., 2019. Semantickitti: A dataset for semantic scene understanding of lidar sequences. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, IEEE.

Besl, P. J. and McKay, N. D., 1992. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14 (2), 239–256.

Brox, T. and Malik, J., 2010. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE transactions on pattern analysis and machine intelligence*, 33 (3), 500–513.

Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G. and Beijbom, O., 2020. nuscenes: A multimodal dataset for autonomous driving. *Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA*, IEEE, 11621–11631.

Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H. et al., 2015. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*.

Chang, M.-F., Lambert, J., Sangkloy, P., Singh, J., Bak, S., Hartnett, A., Wang, D., Carr, P., Lucey, S., Ramanan, D. et al., 2019. Argoverse: 3d tracking and forecasting with rich maps. *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition, CA, USA: IEEE, 8748–8757.

Chen, F., Tsaftaris, S. A. and Giuffrida, M. V., 2024. Gmt: Guided mask transformer for leaf instance segmentation. *arXiv preprint arXiv:2406.17109*.

Chen, K., Lopez, B. T., Agha-mohammadi, A.-a. and Mehta, A., 2022. Direct lidar odometry: Fast localization with dense point clouds. *IEEE Robotics and Automation Letters*, 7 (2), 2000–2007.

Chen, X., Li, S., Mersch, B., Wiesmann, L., Gall, J., Behley, J. and Stachniss, C., 2021a. Moving object segmentation in 3d lidar data: A learning-based approach exploiting sequential data. *IEEE Robotics and Automation Letters*, 6 (4), 6529–6536.

Chen, X., Li, S., Mersch, B., Wiesmann, L., Gall, J., Behley, J. and Stachniss, C., 2021b. Moving object segmentation in 3d lidar data: A learning-based approach exploiting sequential data. *IEEE Robotics and Automation Letters*, 6 (4), 6529–6536.

Chen, Y., Liu, J., Zhang, X., Qi, X. and Jia, J., 2023. Voxelnext: Fully sparse voxelnet for 3d object detection and tracking. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21674–21683.

Cheng, B., Misra, I., Schwing, A. G., Kirillov, A. and Girdhar, R., 2022. Masked-attention mask transformer for universal image segmentation. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1290–1299.

Cheng, W. and Ko, J. H., 2022. Bi-pointflownet: Bidirectional learning for point cloud based scene flow estimation. *ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVIII*, Berlin, Heidelberg: Springer-Verlag, 108–124.

Cortes, C., Haffner, P. and Mohri, M., 2002. Rational kernels. *Advances in neural information processing systems*, 15.

Ding, F., Palffy, A., Gavrila, D. M. and Lu, C. X., 2023. Hidden gems: 4d radar scene flow learning using cross-modal supervision. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9340–9349.

Ding, L., Dong, S., Xu, T., Xu, X., Wang, J. and Li, J., 2022. Fh-net: A fast hierarchical network for scene flow estimation on real-world point clouds. *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIX*, Springer, 213–229.

Dong, G., Zhang, Y., Li, H., Sun, X. and Xiong, Z., 2022. Exploiting rigidity constraints for lidar scene flow estimation. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA, USA: IEEE Computer Society, 12766–12775.

Dosovitskiy, A., Ros, G., Codevilla, F., López, A. M. and Koltun, V., 2017. Carla: An open urban driving simulator. *ArXiv*, abs/1711.03938.

Ester, M., Kriegel, H.-P., Sander, J., Xu, X. et al., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. *kdd*, volume 96, 226–231.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J. and Zisserman, A., 2010. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88, 303–338.

Falanga, D., Kim, S. and Scaramuzza, D., 2019. How fast is too fast? the role of perception latency in high-speed sense and avoid. *IEEE Robotics and Automation Letters*, 4 (2), 1884–1891.

Fan, H., Yang, Y. and Kankanhalli, M., 2021. Point 4d transformer networks for spatio-temporal modeling in point cloud videos. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 14204–14213.

Fan, H., Yu, X., Ding, Y., Yang, Y. and Kankanhalli, M., 2022. Pstnet: Point spatio-temporal convolution on point cloud sequences. *arXiv preprint arXiv:2205.13713*.

Fu, J., Xiang, Z., Qiao, C. and Bai, T., 2023. Pt-flownet: Scene flow estimation on point clouds with point transformer. *IEEE Robotics and Automation Letters*, 8 (5), 2566–2573.

Fussell, G., 2023. Gestalt theory: 6 essential principles for design.

Geiger, A., Lenz, P., Stiller, C. and Urtasun, R., 2013. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32 (11), 1231–1237.

Geiger, A., Lenz, P. and Urtasun, R., 2012a. Are we ready for autonomous driving? the kitti vision benchmark suite. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 3354–3361.

Geiger, A., Lenz, P. and Urtasun, R., 2012b. Are we ready for autonomous driving? the kitti vision benchmark suite. *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Glorot, X., Bordes, A. and Bengio, Y., 2011. Deep sparse rectifier neural networks. *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, JMLR Workshop and Conference Proceedings, 315–323.

Gojcic, Z., Litany, O., Wieser, A., Guibas, L. J. and Birdal, T., 2021. Weakly supervised learning of rigid 3d scene flow. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, Los Alamitos, CA, USA: IEEE Computer Society, 5688–5699.

Gu, X., Tang, C., Yuan, W., Dai, Z., Zhu, S. and Tan, P., 2022. Rcp: Recurrent closest point for scene flow estimation on 3d point clouds. *arXiv preprint arXiv:2205.11028*.

Gu, X., Wang, Y., Wu, C., Lee, Y. J. and Wang, P., 2019. Hplflownet: Hierarchical permutohedral lattice flownet for scene flow estimation on large-scale point clouds. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 3254–3263.

He, K., Zhang, X., Ren, S. and Sun, J., 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *Proceedings of the IEEE international conference on computer vision*, 1026–1034.

Hong, F., Zhou, H., Zhu, X., Li, H. and Liu, Z., 2021. Lidar-based panoptic segmentation via dynamic shifting network. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 13090–13099.

Houston, J., Zuidhof, G., Bergamini, L., Ye, Y., Jain, A., Omari, S., Iglovikov, V. and Ondruska, P., 2020. One thousand and one hours: Self-driving motion prediction dataset.

Huang, J., Wang, H., Birdal, T., Sung, M., Arrigoni, F., Hu, S.-M. and Guibas, L. J., 2021. Multibodysync: Multi-body segmentation and motion estimation via 3d scan synchronization. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7108–7118.

Huang, S., Gojcic, Z., Huang, J., Wieser, A. and Schindler, K., 2022a. Dynamic 3d scene analysis by point cloud accumulation. *European Conference on Computer Vision*, Springer, 674–690.

Huang, S., Gojcic, Z., Huang, J., Wieser, A. and Schindler, K., 2022b. Dynamic 3d scene analysis bynbsp;point cloud accumulation. *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVIII*, Berlin, Heidelberg: Springer-Verlag, 674–690. URL `https://doi.org/10.1007/978-3-031-19839-7_39`.

Hubel, D. H. and Wiesel, T. N., 1968. Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, 195 (1), 215–243.

Hui, L., Tang, L., Shen, Y., Xie, J. and Yang, J., 2022. Learning superpoint graph cut for 3d instance segmentation. *Advances in Neural Information Processing Systems*, 35, 36804–36817.

Hur, J. and Roth, S., 2020. Self-supervised monocular scene flow estimation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA, USA: IEEE Computer Society, 7394–7403.

Ilg, E., Saikia, T., Keuper, M. and Brox, T., 2018. Occlusions, motion and depth boundaries with a generic network for disparity, optical flow or scene flow estimation. *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany: Springer, 614–630.

Jiang, C., Wang, G., Miao, Y. and Wang, H., 2022. 3d scene flow estimation on pseudo-lidar: Bridging the gap on estimating point motion. *IEEE Transactions on Industrial Informatics*.

Jin, Z., Lei, Y., Akhtar, N., Li, H. and Hayat, M., 2022. Deformation and correspondence aware unsupervised synthetic-to-real scene flow estimation for point clouds, 7223–7233.

Jund, P., Sweeney, C., Abdo, N., Chen, Z. and Shlens, J., 2021. Scalable scene flow from point clouds in the real world. *IEEE Robotics and Automation Letters*, 7 (2), 1589–1596.

Kabsch, W., 1976. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A*, 32 (5), 922–923.

Kenney, J., Buckley, T. and Brock, O., 2009. Interactive segmentation for manipulation in unstructured environments. *2009 IEEE International Conference on Robotics and Automation*, IEEE, 1377–1382.

Kesten, R., Usmana, M., Houston, J., Pandya, T., Nadhamuni, K., Ferreira, A., Yuan, M., Low, B., Jain, A., Ondruska, P., Omari, S., Shah, S., Kulkarni, A., Kazakova, A., Tao, C., Platinsky, L., Jiang, W. and Shet, V., 2019. Lyft level 5 av dataset. URL `https://level5.lyft.com/dataset/`.

Khatri, I., Vedder, K., Peri, N., Ramanan, D. and Hays, J., 2024. I can't believe it's not scene flow! *arXiv preprint arXiv:2403.04739*.

Khatri, I., Vedder, K., Peri, N., Ramanan, D. and Hays, J., 2025. I can't believe it's not scene flow! *European Conference on Computer Vision*, Springer, 242–257.

Kittenplon, Y., Eldar, Y. C. and Raviv, D., 2021. Flowstep3d: Model unrolling for self-supervised scene flow estimation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4114–4123.

Krispel, G., Opitz, M., Waltner, G., Possegger, H. and Bischof, H., 2020. Fuseseg: Lidar point cloud segmentation fusing multi-modal data. *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 1874–1883.

Lang, I., Aiger, D., Cole, F., Avidan, S. and Rubinstein, M., 2022. Scoop: Self-supervised correspondence and optimization-based scene flow. *arXiv preprint arXiv:2211.14020*.

Li, B., Zheng, C., Giancola, S. and Ghanem, B., 2022a. Sctn: Sparse convolution-transformer network for scene flow estimation. 36 (2), 1254–1262.

Li, B., Zheng, C., Li, G. and Ghanem, B., 2022b. Learning scene flow in 3d point clouds with noisy pseudo labels. *arXiv preprint arXiv:2203.12655*.

Li, H., Dong, G., Zhang, Y., Sun, X. and Xiong, Z., 2022c. Rppformer-flow: Relative position guided point transformer for scene flow estimation. *Proceedings of the 30th ACM International Conference on Multimedia*, New York, NY, USaA: Association for Computing Machinery, MM '22, 4867–4876.

Li, L., Yang, F., Zhu, H., Li, D., Li, Y. and Tang, L., 2017. An improved ransac for 3d point cloud plane segmentation based on normal distribution transformation cells. *Remote Sensing*, 9 (5), 433.

Li, Q., Zhuang, Y., Chen, Y., Huai, J., Li, M., Ma, T., Tang, Y. and Liang, X., 2023. 3d-seqmos: A novel sequential 3d moving object segmentation in autonomous driving. *arXiv preprint arXiv:2307.09044*.

Li, R., Lin, G., He, T., Liu, F. and Shen, C., 2021a. Hcrf-flow: Scene flow from point clouds with continuous high-order crfs and position-aware flow embedding. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Los Alamitos, CA, USA: IEEE Computer Society, 364–373.

Li, R., Lin, G. and Xie, L., 2021b. Self-point-flow: Self-supervised scene flow estimation from point clouds with optimal transport and random walk, 15572–15581.

Li, R., Zhang, C., Lin, G., Wang, Z. and Shen, C., 2022d. Rigidflow: Self-supervised scene flow learning on point clouds by local rigidity prior. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA, USA: IEEE Computer Society, 16959–16968.

Li, X., Kaesemodel Pontes, J. and Lucey, S., 2021c. Neural scene flow prior. *Advances in Neural Information Processing Systems*, 34, 7838–7851.

Li, X. and Lucey, S., 2024. Fast kernel scene flow. *arXiv preprint arXiv:2403.05896*.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. and Zitnick, C. L., 2014. Microsoft coco: Common objects in context. *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, Springer, 740–755.

Lin, X., Casas, J. R. and Pardàs, M., 2018. Temporally coherent 3d point cloud video segmentation in generic scenes. *IEEE Transactions on Image Processing*, 27 (6), 3087–3099.

Liu, H., Lu, T., Xu, Y., Liu, J., Li, W. and Chen, L., 2022. Camliflow: bidirectional camera-lidar fusion for joint optical flow and scene flow estimation. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5791–5801.

Liu, S.-H., Yu, S.-Y., Wu, S.-C., Chen, H.-T. and Liu, T.-L., 2020. Learning gaussian instance segmentation in point clouds. *arXiv preprint arXiv:2007.09860*.

Liu, X., Qi, C. R. and Guibas, L. J., 2019a. Flownet3d: Learning scene flow in 3d point clouds. *Proceedings of the IEEE/CVF Conference on*

Computer Vision and Pattern Recognition, California, USA: IEEE, 529–537.

Liu, X., Yan, M. and Bohg, J., 2019b. Meteornet: Deep learning on dynamic 3d point cloud sequences. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9246–9255.

Lu, F., Chen, G., Li, Z., Zhang, L., Liu, Y., Qu, S. and Knoll, A., 2022. Monet: Motion-based point cloud prediction network. *IEEE Transactions on Intelligent Transportation Systems*, 23 (8), 13794–13804.

Lu, Y., Jiang, Q., Chen, R., Hou, Y., Zhu, X. and Ma, Y., 2023. See more and know more: Zero-shot point cloud segmentation via multi-modal visual data. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 21674–21684.

Luo, C., Yang, X. and Yuille, A., 2021. Self-supervised pillar motion learning for autonomous driving, 3182–3191.

Marcuzzi, R., Nunes, L., Wiesmann, L., Vizzo, I., Behley, J. and Stachniss, C., 2022. Contrastive instance association for 4d panoptic segmentation using sequences of 3d lidar scans. *IEEE Robotics and Automation Letters*, 7 (2), 1550–1557.

Maretic, H. P., Gheche, M. E., Chierchia, G. and Frossard, P., 2019. Got: An optimal transport framework for graph comparison. *Advances in Neural Information Processing Systems*.

Marichal, X. and Umeda, T., 2003. Real-time segmentation of video objects for mixed-reality interactive applications. *Visual Communications and Image Processing 2003*, SPIE, volume 5150, 41–50.

Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A. and Brox, T., 2016. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. *Proceedings of the*

IEEE conference on computer vision and pattern recognition, 4040–4048.

Mazur, K. and Lempitsky, V., 2021. Cloud transformers: A universal approach to point cloud processing tasks. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10715–10724.

Menze, M. and Geiger, A., 2015. Object scene flow for autonomous vehicles. *Proceedings of the IEEE conference on computer vision and pattern recognition*, Boston, MA, USA: IEEE, 3061–3070.

Menze, M., Heipke, C. and Geiger, A., 2015. Joint 3d estimation of vehicles and scene flow. *ISPRS annals of the photogrammetry, remote sensing and spatial information sciences*, 2, 427.

Menze, M., Heipke, C. and Geiger, A., 2018. Object scene flow. *ISPRS Journal of Photogrammetry and Remote Sensing*, 140, 60–76.

Misra, I., Girdhar, R. and Joulin, A., 2021. An end-to-end transformer model for 3d object detection. *Proceedings of the IEEE/CVF international conference on computer vision*, 2906–2917.

Mittal, H., Okorn, B. and Held, D., 2020. Just go with the flow: Self-supervised scene flow estimation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Washington, USA: IEEE, 11177–11185.

Muhammad, K., Hussain, T., Ullah, H., Del Ser, J., Rezaei, M., Kumar, N., Hijji, M., Bellavista, P. and de Albuquerque, V. H. C., 2022. Vision-based semantic segmentation in scene understanding for autonomous driving: Recent achievements, challenges, and outlooks. *IEEE Transactions on Intelligent Transportation Systems*, 23 (12), 22694–22715.

Nikolentzos, G., Meladianos, P. and Vazirgiannis, M., 2017. Matching node embeddings for graph similarity. *Proceedings of the AAAI Conference on Artificial Intelligence*, California, USA, volume 31.

Nunes, U. M. and Demiris, Y., 2018. 3d motion segmentation of articulated rigid bodies based on rgb-d data. *BMVC*, volume 5, 7.

Ochs, P., Malik, J. and Brox, T., 2013. Segmentation of moving objects by long term video analysis. *IEEE transactions on pattern analysis and machine intelligence*, 36 (6), 1187–1200.

Ouyang, B. and Raviv, D., 2021a. Occlusion guided scene flow estimation on 3d point clouds, 2799–2808.

Ouyang, B. and Raviv, D., 2021b. Occlusion guided self-supervised scene flow estimation on 3d point clouds. *2021 International Conference on 3D Vision (3DV)*, IEEE, 782–791.

Parra, G. and Tobar, F., 2017. Spectral mixture kernels for multi-output gaussian processes. *Advances in Neural Information Processing Systems*, 30.

Pontes, J. K., Hays, J. and Lucey, S., 2020. Scene flow from point clouds with or without learning, 261–270.

Powers, D. M., 2020. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*.

Puy, G., Boulch, A. and Marlet, R., 2020. *FLOT: Scene Flow on Point Clouds Guided by Optimal Transport*. 527–544.

Qi, C. R., Su, H., Mo, K. and Guibas, L. J., 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 652–660.

Qi, C. R., Yi, L., Su, H. and Guibas, L. J., 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Proceedings*

of the 31st International Conference on Neural Information Processing Systems, Red Hook, NY, USA: Curran Associates Inc., NIPS'17, 5105–5114.

Ren, S., Luzi, F., Lahrichi, S., Kassaw, K., Collins, L. M., Bradbury, K. and Malof, J. M., 2024. Segment anything, from space? *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 8355–8365.

Schölkopf, B. and Smola, A. J., 2002. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

Seeger, M., 2004. Gaussian processes for machine learning. *International journal of neural systems*, 14 (02), 69–106.

Shi, H., Lin, G., Wang, H., Hung, T.-Y. and Wang, Z., 2020. Spsequencenet: Semantic segmentation network on 4d point clouds. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4574–4583.

Shi, Y., Cao, X. and Zhou, B., 2021. Self-supervised learning of part mobility from point cloud sequence. *Computer Graphics Forum*, Wiley Online Library, volume 40, 104–116.

Shi, Y. and Ma, K., 2022. Safit: Segmentation-aware scene flow with improved transformer. *2022 International Conference on Robotics and Automation (ICRA)*, IEEE, 10648–10655.

Song, Z. and Yang, B., 2022. Ogc: Unsupervised 3d object segmentation from rigid dynamics of point clouds. *Advances in Neural Information Processing Systems*, 35, 30798–30812.

Song, Z. and Yang, B., 2024. Unsupervised 3d object segmentation of point clouds by geometry consistency. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–14.

Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B. et al., 2020. Scalability in perception for autonomous driving: Waymo open dataset. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA, USA: IEEE Computer Society, 2443–2451.

Teed, Z. and Deng, J., 2020. Raft: Recurrent all-pairs field transforms for optical flow. *European Conference on Computer Vision*, Springer, 402–419.

Teed, Z. and Deng, J., 2021. Raft-3d: Scene flow using rigid-motion embeddings. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8375–8384.

Tishchenko, I., Lombardi, S., Oswald, M. R. and Pollefeys, M., 2020. Self-supervised learning of non-rigid residual flow and ego-motion. *2020 International Conference on 3D Vision (3DV)*, Fukuoka, Japan: IEEE, 150–159.

Triess, L. T., Peter, D., Rist, C. B. and Zöllner, J. M., 2020. Scan-based semantic segmentation of lidar point clouds: An experimental study. *2020 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 1116–1121.

Umam, A., Yang, C.-K., Chuang, J.-H. and Lin, Y.-Y., 2024. Unsupervised point cloud co-part segmentation via co-attended superpoint generation and aggregation. *IEEE Transactions on Multimedia*.

Vedder, K., Peri, N., Chodosh, N., Khatri, I., Eaton, E., Jayaraman, D., Liu, Y., Ramanan, D. and Hays, J., 2023. Zeroflow: Scalable scene flow via distillation. *arXiv preprint arXiv:2305.10424*.

Vedula, S., Baker, S., Rander, P., Collins, R. and Kanade, T., 1999. Three-dimensional scene flow. *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, 722–729 vol.2.

Vogel, C., Roth, S. and Schindler, K., 2014. View-consistent 3d scene flow estimation over multiple frames. *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV 13*, Springer, 263–278.

Vu, T., Kim, K., Luu, T. M., Nguyen, T. and Yoo, C. D., 2022. Softgroup for 3d instance segmentation on point clouds. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2708–2717.

Wang, C., Li, X., Pontes, J. K. and Lucey, S., 2022a. Neural prior for trajectory estimation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6532–6542.

Wang, G., Hu, Y., Liu, Z., Zhou, Y., Tomizuka, M., Zhan, W. and Wang, H., 2022b. *What Matters for 3D Scene Flow Network*, Springer. 38–55.

Wang, G., Hu, Y., Wu, X. and Wang, H., 2021a. Residual 3d scene flow learning with context-aware feature extraction. *arXiv preprint arXiv:2109.04685*.

Wang, G., Jiang, C., Shen, Z., Miao, Y. and Wang, H., 2022c. Sfgan: Unsupervised generative adversarial learning of 3d scene flow from the 3d scene self. *Advanced Intelligent Systems*, 4 (4), 2100197.

Wang, G., Wu, X., Liu, Z. and Wang, H., 2021b. Hierarchical attention learning of scene flow in 3d point clouds. *IEEE Transactions on Image Processing*, PP, 1–1.

Wang, H., Pang, J., Lodhi, M. A., Tian, Y. and Tian, D., 2021c. Festa: Flow estimation via spatial-temporal attention for scene point clouds, 14168–14177.

Wang, K. and Shen, S., 2022. Estimation and propagation: Scene flow prediction on occluded point clouds. *IEEE Robotics and Automation Letters*, 7, 12201–12208.

Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., Lu, T., Luo, P. and Shao, L., 2022d. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8 (3), 415–424.

Wang, W., Yu, R., Huang, Q. and Neumann, U., 2018. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2569–2578.

Wang, Y., Chen, Y. and Zhang, Z.-X., 2022e. 4d unsupervised object discovery. *Advances in Neural Information Processing Systems*, 35, 35563–35575.

Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M. and Solomon, J. M., 2019. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)*, 38 (5), 1–12.

Wang, Z., Li, S., Howard-Jenkins, H., Prisacariu, V. and Chen, M., 2020. Flownet3d++: Geometric losses for deep scene flow estimation. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, Snowmass Village, CO, USA: IEEE, 91–98.

Wang, Z., Wei, Y., Rao, Y., Zhou, J. and Lu, J., 2023. 3d point-voxel correlation fields for scene flow estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45 (11), 13621–13635.

Ward Jr, J. H., 1963. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58 (301), 236–244.

Wei, Y., Wang, Z., Rao, Y., Lu, J. and Zhou, J., 2021. Pv-raft: Point-voxel correlation fields for scene flow estimation of point clouds. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA, USA: IEEE Computer Society, 6950–6959.

Wilson, A. and Nickisch, H., 2015. Kernel interpolation for scalable structured gaussian processes (kiss-gp). *International conference on machine learning*, PMLR, 1775–1784.

Wu, W., Shan, Q. and Fuxin, L., 2022a. Pointconvformer: Revenge of the point-based convolution. *arXiv preprint arXiv:2208.02879*.

Wu, W., Wang, Z. Y., Li, Z., Liu, W. and Fuxin, L., 2020. Pointpwc-net: Cost volume on point clouds for (self-) supervised scene flow estimation. *European Conference on Computer Vision*, Springer, 88–107.

Wu, X., Lao, Y., Jiang, L., Liu, X. and Zhao, H., 2022b. Point transformer v2: Grouped vector attention and partition-based pooling. *NeurIPS*.

Xu, H., Zhang, J., Cai, J., Rezatofighi, H. and Tao, D., 2022. Gmflow: Learning optical flow via global matching. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8121–8130.

Yang, X., Yuan, L., Wilber, K., Sharma, A., Gu, X., Qiao, S., Debats, S., Wang, H., Adam, H., Sirotenko, M. et al., 2024. Polymax: General dense prediction with mask transformer. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 1050–1061.

Yin, J., Shen, J., Gao, X., Crandall, D. J. and Yang, R., 2021. Graph neural network and spatiotemporal transformer attention for 3d video object detection from point clouds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45 (8), 9822–9835.

You, Y., Luo, K., Phoo, C. P., Chao, W.-L., Sun, W., Hariharan, B., Campbell, M. and Weinberger, K. Q., 2022. Learning to detect mobile objects from lidar scans without labels. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1130–1140.

Zhai, G., Kong, X., Cui, J., Liu, Y. and Yang, Z., 2020. Flowmot: 3d multi-object tracking by scene flow association. *arXiv preprint arXiv:2012.07541*.

Zhang, S., Cui, S. and Ding, Z., 2020. Hypergraph spectral clustering for point cloud segmentation. *IEEE Signal Processing Letters*, 27, 1655–1659.

Zhao, H., Jiang, L., Jia, J., Torr, P. H. and Koltun, V., 2021. Point transformer. *Proceedings of the IEEE/CVF international conference on computer vision*, 16259–16268.

Zhong, J.-X., Cheng, T.-Y., He, Y., Lu, K., Zhou, K., Markham, A. and Trigoni, N., 2023. Multi-body se (3) equivariance for unsupervised rigid segmentation and motion estimation. *Advances in Neural Information Processing Systems*, 36, 76085–76097.

Zhong, J.-X., Cheng, T.-Y., He, Y., Lu, K., Zhou, K., Markham, A. and Trigoni, N., 2024. Multi-body se (3) equivariance for unsupervised rigid segmentation and motion estimation. *Advances in Neural Information Processing Systems*, 36.

Zou, C., He, B., Zhu, M., Zhang, L. and Zhang, J., 2019. Learning motion field of lidar point cloud with convolutional networks. *Pattern Recognition Letters*, 125, 514–520.

Zuanazzi, V., van Vugt, J., Booij, O. and Mettes, P., 2020. Adversarial self-supervised scene flow estimation. *2020 International Conference on 3D Vision (3DV)*, Fukuoka, Japan: IEEE, 1049–1058.