Deep Learning for Scene Flow Estimation on Point Clouds: A Survey and Prospective Trends

Zhiqi Li,[1] Nan Xiang,[2] Honghua Chen,[3] Jianjun Zhang[1] and Xiaosong Yang[1]

[1]Bournemouth University, Poole, UK
xyang@bournemouth.ac.uk
[2]Department of Computing School of Advanced Technology, Xi'an Jiaotong-Liverpool University, Suzhou, China
[3]School of Computer Science and Engineering, Nanyang Technological University, Singapore, Singapore

Abstract
Aiming at obtaining structural information and 3D motion of dynamic scenes, scene flow estimation has been an interest of research in computer vision and computer graphics for a long time. It is also a fundamental task for various applications such as autonomous driving. Compared to previous methods that utilize image representations, many recent researches build upon the power of deep analysis and focus on point clouds representation to conduct 3D flow estimation. This paper comprehensively reviews the pioneering literature in scene flow estimation based on point clouds. Meanwhile, it delves into detail in learning paradigms and presents insightful comparisons between the state-of-the-art methods using deep learning for scene flow estimation. Furthermore, this paper investigates various higher-level scene understanding tasks, including object tracking, motion segmentation, etc. and concludes with an overview of foreseeable research trends for scene flow estimation.

Keywords: 3D scene flow, literature survey

CCS Concepts: • Computing methodologies → Scene understanding; Tracking; Learning paradigms; Motion capture

1. Introduction

A bunch of research works have emerged from autonomous driving (AD) to support advanced transportation sector. In this context, understanding the complex environment is vital for automated vehicles to drive safely. A scene flow estimator could intuitively discriminates different motion patterns of moving agents, for example pedestrians, cyclists, cars, and so on from on-board sensor data. As shown in Figure 1, scene flow represents the motion field of individual objects in a 3D scene [VBR*99]. 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 [HR20], which yield limited accuracy and suffers from long runtime. The advent of affordable 3D sensors, for example 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. Deep learning (DL) is a branch of machine learning techniques, which usually utilizes deep neural networks to solve machine learn-

ing problems. It extracts features automatically and emphasizes on learning a high-level abstract representations of data [GHH*21]. Learning process can be fully-supervised, weakly-supervised, and self-supervised. A plethora of deep learning techniques on point clouds have emerged to solve different classical computer vision tasks such as 3D shape classification [MWB21, GLMH55], object detection [ZCL20, QCLG20], object tracking [SHHX18], semantic scene segmentation [ZZtZX20, HYX*20], and instance segmentation [JYC*20], to name a few. With the rise of deep learning techniques for scene understanding tasks, deploying deep neural networks for scene flow estimation has attracted increasing research attention.

Thanks to the introduction of large-scale synthetic dataset FlyingThings3D [MIH*16] 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, for example ICP [BM92] and shows great potential to be applied in real scenarios. To this end, datasets such as KITTI [MG15], NuScenes [CBL*], and Argoverse [CLS*19] are created,
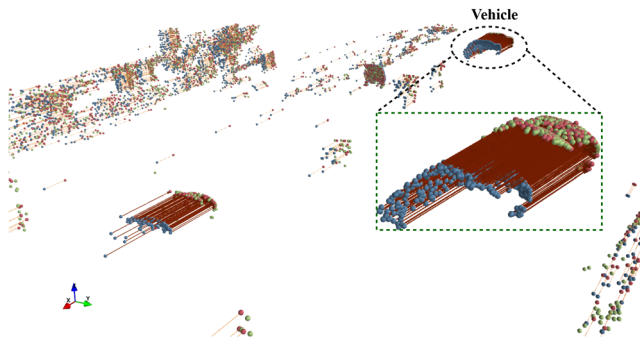
Figure 1: *Visualization of scene flow for a KITTI example scene. The source point cloud is shown in blue and the target point cloud is shown in green. The deformed points are obtained by adding scene flow vectors (shown in red arrows) to source points, which are shown in PaleVioletRed.*

which contain real scenes scanned from various actual environment. However, datasets collected by LiDAR do not provide reliable correspondences between consecutive scans. Therefore, a lot of DL 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 DL models, unleashing the full power of deep neural networks on 3D point cloud understanding is still in its infancy.

We summarize and categorize current challenges in scene flow estimation into data challenges and DL 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 neighbourhood.
- **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.
- **LiDAR challenge**. Environment interference is challenging to data collection using LiDAR. Although LiDAR is not sensitive to the variations of lighting, it is still struggling with reflective surfaces and bad weather (e.g. heavy fog, rain, and snow). The consequence of these imperfections is the loss of object motion and structure information.

Challenges from DL 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.
- **Transformation challenge**. There exist multiple transformations (e.g. rotations, translations) in real dynamic scenes, which is challenging for DL models to handle effectively. Some objects will be distorted in the consecutive frames if their transformations are not strictly aligned.
- **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 DL algorithms.
- **Efficiency challenge**. Real-time processing ability is imperative for AD entities. However, the computing power and memory space allocated for processing massive 3D data constructed on vehicles are limited. Currently, efficient DL model that can produce real-time large scene perception is still under-explored.

There are a few surveys [YX16, XAZX17] which have thorough analyses of methods for traditional optical flow estimation and depth estimation. Xiang et al. [XAZX17] reviewed scene flow applications, including image segmentation, image matching, and feature extraction. However, they do not provide sufficient quantitative comparisons between different methods and lack the review of learning based methods. Recently, Liu et al. [LLW*20] and Zhai et al. [ZXLK21] have presented some learning-based scene flow estimation literature and compared their performance on various datasets. Unlike Liu et al. [LLW*20] that only outlined image-based scene flow estimation methods, Zhai et al. [ZXLK21] cover both the optical flow (2D) and scene flow (3D) estimation literature and categorize them into knowledge-driven, data-driven and hybrid-driven methods. Zhai et al. [ZXLK21] introduce scene flow estimation approaches according to the dimension of data representation: 2.5D (image-based) and 3D (point-based). This survey aims to narrow the gap in this topic. Therefore, we comprehensively review up-to-date compelling DL models applied in point cloud-based scene flow estimation approaches. The main contributions of this paper are summarized as follows:

- **Comprehensive review**. For the first time, we investigate DL methods for point cloud-based scene flow estimation. We provide a comprehensive comparison and insightful analysis on recent deep learning methods (2019–2023), including supervised, weakly-supervised, and self-supervised scene flow estimation methods.

- **Review of open challenges**. We provide an overview of existing challenges in scene flow estimation, which is categorized into data challenge and DL challenge.
- **Applications and research directions**. We present how the estimated scene flow can benefit higher-level scene understanding tasks. Several promising research directions in 3D scene flow estimation are discussed.

2. Problem Statement and Taxonomy

Let $P \in R^{N_1 \times 3}$ denotes point cloud at time t with N_1 points and $Q \in R^{N_2 \times 3}$ represents the point cloud at time $t + 1$ with N_2 points. Scene flow estimation aims at recovering the 3D motion from point cloud P captured at the first frame to point cloud Q at the next frame. Therefore, the target for scene flow estimation is that each point $p_i \in P$ should be as near as possible to the corresponding point $q_i \in Q$ after scene flow recovery. It is noteworthy that due to the sparsity and unstructured nature of point clouds, the source point cloud and the target point cloud do not necessarily have the same number of points or have hard correspondences. Many methods estimate scene flow vectors for the points in the first point cloud. With this prior setting, the per-object transformation parameters can be predicted [GLW*21]. The prominent methods only use point coordinates to estimate the motion vector. There is also an attempt [LLX21] that makes use of colour and surface normal as additional clues to find point correspondences.

Evaluation metrics.. There are four main metrics to evaluate the predicted scene flow. More detailed equations of the following terms can be found in [WHWW21].

- 3D End Point Error (**EPE3D**): it is the average absolute distance (L_2 distance) between the predicted flow vector and ground truth flow vector in meters.
- **Acc3DS**: it is the percentage of flow vectors whose **EPE3D** $< 0.05m$ or the relative error $< 5\%$.
- **Acc3DR**: it is the percentage of flow vectors whose **EPE3D** $< 0.1m$ or the relative error $< 10\%$.
- **Outliers**: if the EPE3D of a point $> 0.3m$, it is considered as an outlier. So this metric depicts the percentage of points whose EPE3D $> 0.3m$ or relative error $> 10\%$.

3. Building Blocks in Scene Flow Estimation

This section summarizes some basic building blocks for scene flow estimation that comprise the DL pipeline. Learning-based frameworks for scene flow estimation from point clouds usually consist of three stages: (1) feature extraction; (2) feature fusion and matching; and (3) flow generation and refinement.

3.1. Feature extraction paradigms

Traditional convolutions are not suitable to irregular point sets. To enable effective usage of the geometry domain knowledge on point clouds, point feature learning is an essential step. We introduce the dominant feature extraction blocks leveraged by scene flow estimation methods in this section.

3.1.1. *Set conv layer*

Set conv layer is first proposed in PointNet++ for point cloud classification and segmentation [QYSG17]. Point feature is independently calculated via an MLP (multi layer perceptron) and then accumulated by max pooling. A set conv layer takes N points $p_i = \{x_i, f_i\}$ with its XYZ coordinates $x_i \in \mathbb{R}^3$ and its feature $f_i \in \mathbb{R}^c$ ($i = 1, \ldots, N$) as input. The outputs include a sub-sampled point cloud with N' points and the point-wise feature, where $p_j = \{x'_j, f'_j\}$. For each sub-sampled region (centred at point x'_j) defined by a ball neighbourhood specified by radius r, the updated local feature is computed based on a symmetric function defined as

$$f'_j = \underset{\{i| \ ||x_i - x'_j|| \leq r\}}{\mathrm{MAX}} \left\{ h(f_i, x_i - x'_j) \right\}, \tag{1}$$

where $h(\cdot)$ is a non-linear function (an MLP layer) with concatenated f_i and point difference $x_i - x'_j$ as inputs [QYSG17], and MAX is the element-wise max pooling operator.

3.2. PointConv feature pyramid

PointConv layer is proposed to learn point features hierarchically. The PointConv method involves inputting the positions of point clouds and training an MLP to estimate a weight function. The method also involves applying an inverse density scale to the learned weights to adjust for non-uniform sampling. It has been leveraged by many scene flow estimation works. The PointPWC-Net generates multiple levels of feature representations, with each level computed through convolution on the previous level. The PointConv operation is defined as follows:

$$\mathrm{PointConv}(S, W, F)_{xyz} = \sum_{(\delta_x, \delta_y, \delta_z) \in G} S(\delta_x, \delta_y, \delta_z)$$
$$W(\delta_x, \delta_y, \delta_z) F(x + \delta_x, y + \delta_y, z + \delta_z) \tag{2}$$

where $S(\delta_x, \delta_y, \delta_z)$ denotes the inverse density at a point $(\delta_x, \delta_y, \delta_z)$. The weight function $W(\delta_x, \delta_y, \delta_z)$ is approximated by MLPs from 3D coordinates $(\delta_x, \delta_y, \delta_z)$ and the inverse density $S(\delta_x, \delta_y, \delta_z)$. $F(x + \delta_x, y + \delta_y, z + \delta_z)$ represents the feature of a point in the local region G centred around point $p = (x, y, z)$. After point convolution, the feature in a local region is updated.

3.3. Point information fusion and matching

3.3.1. *Flow embedding layer*

This layer learns to aggregate both feature similarities and spatial relationships of points to yield embedded features for point motions [LQG19]. As illustrated in Figure 2, this layer fuses the feature from source point cloud P: $\{p_i = (x_i, f_i)\}_{i=1}^{N_1}$ and target point cloud Q: $\{q_j = (y_j, g_j)\}_{j=1}^{N_2}$. The flow embedding is computed by

$$e_i = \underset{\{j| \ ||y_j - x_i|| \leq r\}}{\mathrm{MAX}} \left\{ h(f_i, g_j, y_j - x_i) \right\}. \tag{3}$$

An improved version of this embedding layer is proposed by Wang et al. [WWLW21], a weighted embedding strategy that samples neighbouring points in the second frame for the source point. Motion embedded based on a patch-to-patch manner involves the larger receptive field of each point.
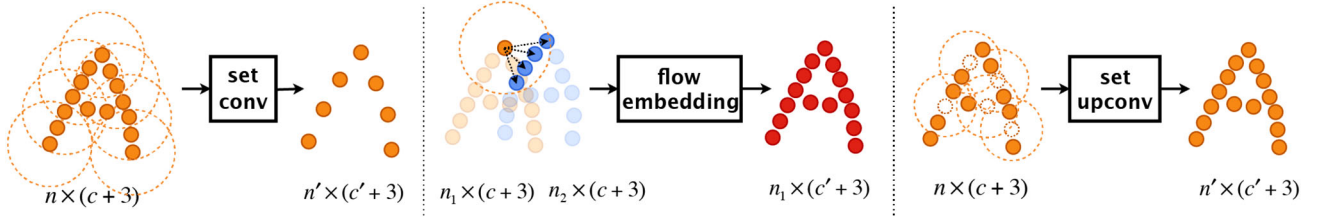
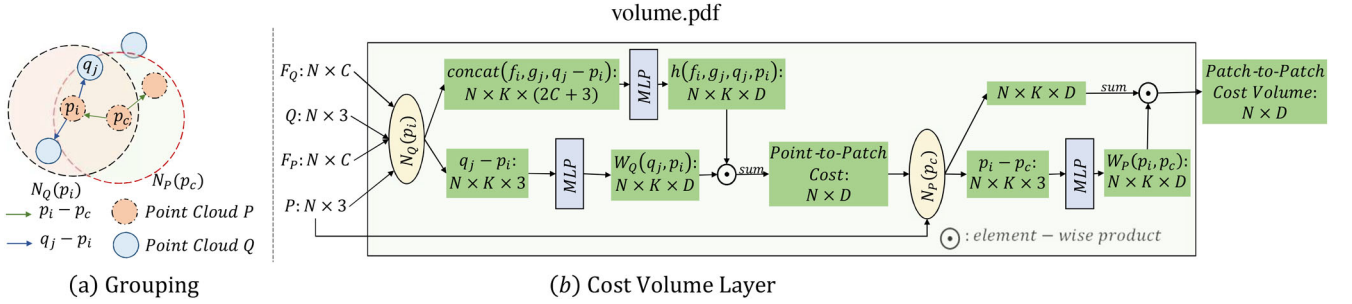**Figure 2:** *The network architecture of FlowNet3D [LQG19].*

(a) Grouping (b) Cost Volume Layer

Figure 3: *The cost volume layer in PointPWC-Net [WWL*20].*

3.3.2. *Correlation matrix*

Another stream of works attempts to find soft correspondences between source point cloud P and target point cloud Q. Inspired by optimal transport theory [Vil09], building an optimal transport could help address one-to-one matching between P and Q [PBM20, LLX21]. Recently, SCTN [LZGG22] and FlowStep3D [KER21] adopt a correlation matrix to estimate point correspondences. The correlation matrix is defined as

$$M(i, j) = 1 - \frac{F_\theta(p_i^t)^T \cdot F_\theta(q_j^{t+1})}{||F_\theta(p_i^t)||_2 ||F_\theta(q_j^{t+1})||_2}, \quad (4)$$

where p_i^t and q_j^{t+1} represent points from the source and target point clouds separately. $F_\theta(\cdot)$ represents the point feature extraction function. After obtaining this correlation matrix, scene flow can be predicted by the Sinkhorn algorithm [PBM20].

3.3.3. *Cost volume*

Cost volume is widely used in stereo matching, which encodes the relation between two consecutive frames. In 2D image field, cost volume is often represented by a 3D tensor. Constructing cost volume in 3D point clouds is more difficult than in the 2D domain since point clouds are unordered and possess different sampling densities. To reduce the computational complexity, Wu et al. [WWL*20] introduce a discretization operation on the cost volume. The matching cost between point p_i and point q_j is defined as

$$
\begin{aligned}
\text{Cost}(p_i, q_j) &= h(f_i, g_j, q_j, p_i) \\
&= \text{MLP}(concat(f_i, g_j, q_j - p_i)),
\end{aligned} \quad (5)
$$

where $concat(\cdot)$ is the abbreviation of concatenation and f_i, g_j are features correspond to point p_i, q_j. PointPWC-Net [WWL*20] uses

multi-layer perceptron (MLP) to obtain nonlinear relationship between two points with an additional direction vector represented by $(q_j - p_i)$. Based on Equation (5), the cost volume for an individual point p_c is formulated as

$$CV(p_c) = \sum_{p_i \in N_p(p_c)} W_p(p_i, p_c) \sum_{q_j \in N_Q(p_i)} W_Q(q_j, p_i) Cost(q_j, p_i), \quad (6)$$

where W_P and W_Q are convolutional weights to compute the costs from patches in point cloud P to that in point cloud Q. $N_p(p_c)$ represents the neighbourhood of point p_c and $N_Q(p_i)$ represents the neighbourhood of point p_i in point cloud Q. So the cost volume is aggregated in a patch-to-patch matching manner. The pipeline of constructing a cost volume is depicted in Figure 3.

3.4. Flow generation and refinement

3.4.1. *Set upconv layer*

In the upsampling step for flow refinement, the set upconv layer propagates the input set of points into a set of target point coordinates by aggregating the neighbouring point features of the input points. It shares the same structure with set conv layers and it is flexible and trainable to propagate/summarize features from one point cloud to another. We refer readers to FlowNet3D [LQG19] for more details on this layer.

3.4.2. *Gated recurrent unit*

Recurrent updating mechanism is widely used in scene flow estimation methods [GTY*22, KER21, DZL*22]. The updated scene flow vector is produced by Gated Recurrent Unit (GRU) with a few set upconv layers.
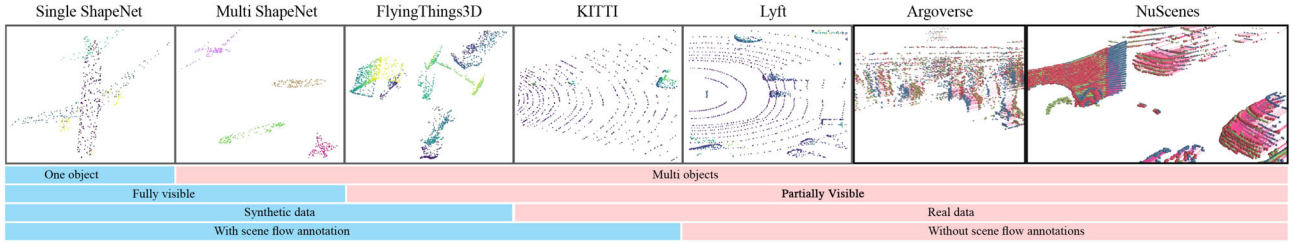
Figure 4: *Illustration and summarization of the differences between datasets: Single ShapeNet, Multi ShapeNet [CFG*15], FlyingThings3D [MIH*16], KITTI Object [MG15], Lyft [HZB*20], Argoverse [CLS*19], and NuScenes [CBL*]. For clarify here, we added two datasets (NuScenes [CBL*] and Argoverse [CLS*19]) based on the version of [ZvVBM20].*

As presented in FlowStep3D [KER21], the hidden state h_k is calculated as

$$z_k = \sigma\big(set\_conv_z([h_{k-1}, x_k])\big),$$
$$r_k = \sigma\big(set\_conv_r([h_{k-1}, x_k])\big),$$
$$\tilde{h}_k = \tanh\big(set\_conv_h([r_k \odot h_{k-1}, x_k])\big),$$
$$h_k = (1 - z_k) \odot h_{k-1} + z_k \odot \tilde{h}_k,$$

where $\odot$ represents Hadamard product and $\sigma(\cdot)$ is the sigmoid activation function. The initial state h_0 is calculated by two set conv layers based on the feature of the source point cloud.

4. Datasets

In this section, we concentrate on point cloud datasets employed in scene flow estimation. A taxonomic study is presented in terms of the source of the data, as elaborated in Figure 4.

4.1. 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 [CFG*15] where the objects are represented by point cloud. Each 3D object in the second frame is yielded through a transformation matrix.
- **Flyingthings3D** 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 3824 examples. FlyingThings3D [MIH*16] fills the gap of datasets lacking ground truth scene flow.
- **GTA-SF** is proposed by DCA-SRSFE[JLA*22] 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.

4.2. Real datasets

As shown in Table 1, we 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** [GLU12] was originally proposed in 2012 for stereo matching and optical flow estimation. It also provides 3D object benchmarks and 3D visual odometry dataset. In the context of scene flow, there are 150 scenes in total with ground truth.
- **KITTI Object** [MG15] is a real-world dataset consisting of 200 annotated scenes of LIDAR data collected using a Velodyne 64 LIDAR.
- **StereoKITTI** [MHG15, MHG18] removes 58 scenes from original data (200 train samples and 200 test 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 [GWW*19].
- **SemanticKITTI** [BGM*19] is based on the odometry dataset of the KITTI Vision Benchmark [GLU12] collected in both urban and rural areas. It provides 21 LiDAR sequences which are split into 11 (00-10) LiDAR sequences for training and 11 (11-21) for testing.
- **Lyft** [KUH*19] contains 22,680 real-scanned scenes with multi-objects. However, it does not provide any point correspondence and is a partially visible dataset. So it can only be used in weakly-supervised methods and used for training.
- **Argoverse** [CLS*19] 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. [PHL20]. The point clouds are collected from two Velodyne VLP-32 sensors. It is noteworthy that the vehicle poses and the 3D object tracks in the original Argoverse 3D Tracking set are utilized to generate pseudo scene flow annotations[PHL20]. The whole dataset contains 2691 training samples and 212 test samples.
- **NuScenes** [CBL*] 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. This difference leads to a

Table 1: *Open real scene datasets. Avg points per frame is the number of points from all LiDAR returns computed on the released data. Trains and tests represent the number of training and testing samples in the dataset. Scenes represents the number of scenes captured in the dataset. Resolution is the corresponding image size of each captured scene. Day&night means the dataset covers data collected day and night.*

Name	Avg points per frame	Trains	Tests	Scenes	Resolution	Day&Night	Traffic Conditions	Annotation
KITTI2015 [MHG15, MG15]	-	150	50	22	(375,1242)	✗	urban, rural	150 frames
LiDAR KITTI [GLU12]	120K	-	-	-	-	✗	urban	Occlusion labels, 3D labels
StereoKITTI [MHG18, MHG15]	-	-	142	-	-	✗	urban	142 frames
NuScenes [CBL*]	34K	1,513	310	1,000	-	✓	urban	40K frames
Waymo [SKD*20]	117K	-	-	1,150	(1920, 1280/1040)	✓	urban	230K frames
Argoverse [CLS*19]	107K	2,691	212	113	(2056,2464)	✗	urban	22K frames
Lyft [KUH*19]	-	18,900	3,780	22,680	-	✗	urban	46K frames

discrepancy in data sparsity that yields a distribution shift between KITTI and NuScenes. NuScenes has recorded diverse data from Boston and Singapore. 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 [SKD*20] 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. It contains 158,081 training and 39,987 validation frames of point clouds with LiDAR labels [JLA*22], such as vehicles, pedestrians, signs and cyclists. However, scene flow labels are not included.

4.3. Data preprocessing

LiDAR raw data are usually scanned in large-scale and unevenly distributed with many irregularly shaped contents. As mentioned before in Section 1, noise and outliers are inevitably imposed when collecting LiDAR data. Therefore, the pre-processing step is necessary for dealing with noise, error, as well as outliers. Pre-processing on point clouds (e.g. ground point removal, down-sampling) is also a significant step before estimating scene flow. Removing ground points with inconspicuous features will enable more efficient learning on point clouds. The most simple method is via thresholding on the height axis, like in HPLFlowNet [GWW*19]. However, this approach is a little aggressive and will lead to important information loss on some objects. In practice, ground points usually constitute a flat plane with less significant visual cues. There are two ground segmentation algorithms: RANSAC and GroundSegNet are proposed to improve the effectiveness of ground points removal. RANSAC is the abbreviation of Random Sampling and Consensus. It fits a plane in a set of points and classifies points close to the plane as ground points [LQG19]. GroundSegNet is originated from the segmentation branch in PointNet [QSMG17], which is trained to classify points to the ground or non-ground part [LQG19]. Both algorithms generate accurate segmentation results on KITTI2015 [MHG15, MG15].

5. Methodology

This section reviews the existing methods from the perspective of supervision and analyses how the state-of-the-art methods deal with challenges that exist in scene flow estimation. We roughly categorized them into the following types: supervised, weakly supervised,

and self-supervised methods. And we refer reader to Figure 5 for a summary of the state-of-the-art learning wisdom of scene flow estimation in the recent few years.

5.1. Supervised methods

Early methods [BMWR19, ZHZ*19] 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, we can identify a core set of the most innovative work on supervised learning approaches 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. Liu et al. [LQG19] proposed FlowNet3D by extracting point features from point clouds directly. It has three main layers for point cloud processing and uses PointNet++ as its backbone for feature learning. As shown in Figure 2, the flow embedding layer aims to aggregate point similarities for scene flow encoding. FlowNet3D finds soft correspondences between point clouds in two consecutive frames. The set upconv layer (Section 3.4.1) 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. [WWLW21] 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 layer by layer. HALFNet [WWLW21] 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 [ISKB18] 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 performance on this kind of dataset.
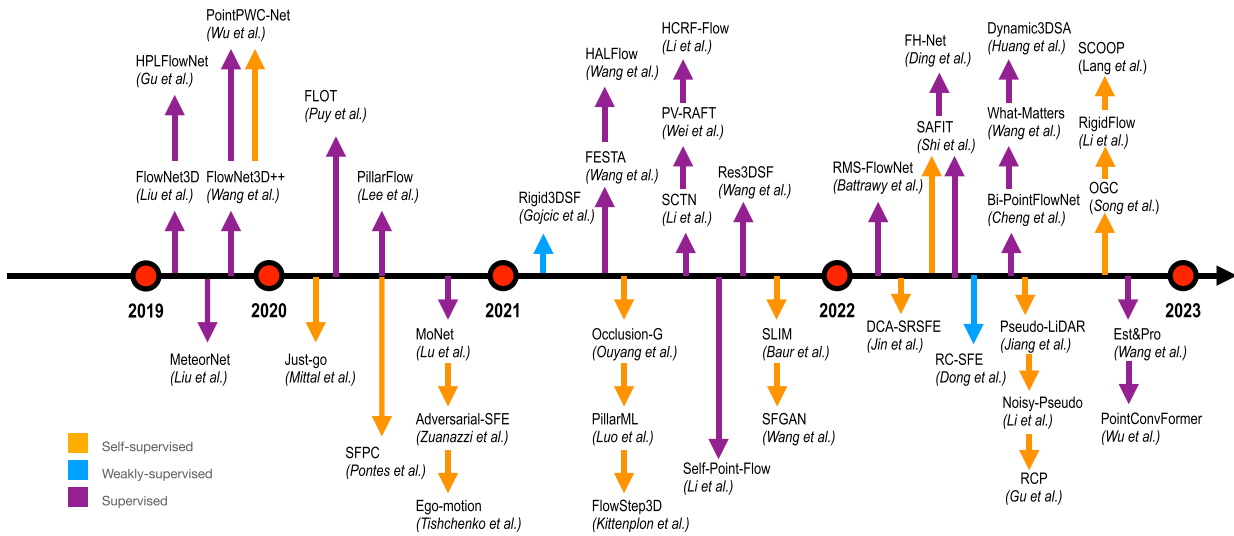
Figure 5: *Chronological overview of the most relevant work on deep learning-based scene flow estimation on 3D Point Clouds.*

FESTA. Previous methods, for example FlowNet3D [LQG19] and MeteorNet [LYB19] 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 [WPL*21]. Hence, it is intractable to estimate accurate scene flow with the unstable features extracted by FPS. FESTA [WPL*21] address this issue via the spatial abstraction with attention (SA^2) 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. Wu et al. proposed PointPWC-Net [WWL*20] that predicts scene flow via constructing the cost volume at each feature pyramid level. To capture large motions, PointPWC-Net proposes 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 [MG15]. Firstly, it failed to perform well when the object is a straight line or a plane. In addition, 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 [WSF22] 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 [WHWW21] includes a context-aware point feature pyramid mod-

ule 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 4 indicate the effectiveness of the framework proposed by Res3DSF [WHWW21]. Res3DSF well addresses the diversity of motion fields, so that it can estimate long-distance motion.

FLOT. Several studies in graph matching, such as [MGCF19, NMV17], utilize optimal transport to find correspondences between two different graphs. Inspired by these works, FLOT [PBM20] 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 [PBM20] 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. The Sinkhorn algorithm is iteratively applied to update the cost matrix, resulting in enhanced scene flow estimation.

SCTN. Different from FLOT [PBM20] which only focuses on sparse 3D coordinates and applies point-based convolutions [QYSG17] to learn features, SCTN [LZGG22] 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 point-wise 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 [LLH*21] leverages a traditional graphical model: high-order conditional random fields (CRFs) where DNNs and CRFs work collaboratively to achieve point-wise 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 [LQG19], which includes set conv layer (Section 3.1.1), flow embedding layer (Section 3.3.1), and set upconv layer (Section 3.4.1). Armed with a position encoding unit and a pseudo pairing unit, HCRF-Flow [LLH*21] can dynamically aggregate matching cost. 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 [WWL*20] utilizes a coarse-to-fine strategy to find point correspondences. However, it suffers from the error accumulation [WWR*21]. PV-RAFT [WWR*21] 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 [TD20]. With voxel correlation features that encodes long-range point clouds, and point-based features that aggregates fine-grained local information, 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 [GWW*19] 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 [WHL*22] follows common practices to compute point features through the set conv layer (Section 3.1.1). To capture reliable match candidates from point clouds even in a long distance, WhatMatters proposes a novel all-to-all point 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 [DDX*22] 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 [SM22] 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 [GWW*19]. Besides, SAFIT has 10.90% and 21.82% accuracy improvement over FLOT on FlyingThings3D and KITTI where occluded points are not removed [LQG19].

Dynamic3DSA. To facilitate the analysis of point cloud sequences, four different tasks are integrated into a complete multi-frame 4D scene analysis approach. Huang et al. [HGH*22] 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 [CK22] 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 -> concatenation -> MLP -> max-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 [WWL*20], Bi-PointFlowNet reduces the total operation by 44% and accelerates the inference by 33%.

Est&Pro. Est&Pro [WS22] 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 non-occluded points to those occluded points. Intuitively, the flow estimator is responsible to the non-occluded points, while the flow propagation network focuses on motion flows of the occluded points.

RMS-FlowNet. RMS-FlowNet [BSMS22] 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 [WWLW21], RMS-FlowNet proposes a Patch-to-Dilated-Patch flow embedding strategy, which re-computes features generated from previous steps with new attention scores. This desgin could speed up the model without sacrificing the accuracy. RMS-FlowNet use a fully supervised loss function similar to PointPWC-Net. This work bears great improvements to the recent efforts on quicker predictions handling large, consecutive point clouds containing over 250K points.

To facilitate an inductive summarization on the above methods, we divide these scene flow models by their building blocks, as listed in Table 2. We also systematically investigate the advantages and disadvantages of different methods.

5.2. Weakly/Self-supervised methods

There are a lot of supervised methods trained on a synthetic dataset and fine-tuned on a small set of real data. However, this training scheme leads to domain gap between the synthetic dataset and the real-scanned dataset, which makes the trained models perform poorly in real-world scenes. A handful of works [MOH20, KER21, JLA*22] have been proposed to handle performance gap between different datasets by devising self-supervised architectures. According to the backbone used by these self-supervised methods, we divide them into flow embedding based, correspondences based, and correspondences free methods. Table 3 summarizes the advantages, deficiencies, and training datasets of these methods.

Just-Go. Mittal et al. [MOH20] utilize nearest neighbour loss and cycle consistency loss based on the framework of FlowNet3D [LQG19]. Nearest neighbour loss is formulated as the average Euclidean distance of the transformed point to its nearest neighbour 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 unlabelled autonomous driving datasets that contain sequential point cloud data. However, it ignores the local geometrical properties of point clouds.

Adversarial-SFE. Victor et al. [ZvVBM20] 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 near-

est neighbours, Adversarial-SFE. [ZvVBM20] 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. [WJS*22] utilize generative adversarial networks (GANs) to learn scene flow. SFGAN [WJS*22] 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. The adversarial training on the generator and discriminator enables SFGAN to specify the consistency of the scene during a period of time.

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, colour, or material reflection. Self-Point-Flow [LLX21] uses global mass constraints with multiple descriptors to formulate one-to-one matching with 3D point coordinate, colour, 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 (labelled node set) and directed subgraph (unlabelled node set).

FlowStep3D. Inspired by RAFT [TD20], FlowStep3D [KER21] 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 gated recurrent unit. FlowStep3D adopts several basic layers, for example set conv layer (Section 3.1.1), flow embedding layer (Section 3.3.1) in Flownet3D [LQG19]. Two regularization loss weights are proposed to adjust the regularization. It contributes to the updating of scene flow during iterations.

Table 2: *Summarization of fully supervised DL architectures for scene flow estimation. FLY3D is the abbreviation of FlyingThings3D. ⋆ denotes methods with open-sourced code.*

	Methods	Highlights	Datasets used
Feature embedding based Methods	Flownet3D⋆ [LQG19]	**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++ [WLHJ*20]	**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⋆ [WPL*21]	**Pros**: Point clouds with RGB information as input; Temporal-Spatial attention mechanism; Occlusion aware. **Cons**: Poor generalization ability.	LiDAR KITTI, FLY3D
	HALFlow [WWLW21]	**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 [LLH*21]	**Pros**: Point-level and region-level constraints; Good generalization ability. **Cons**: Time-consuming.	StereoKITTI, FLY3D
	Bi-PointFlowNet⋆ [CK22]	**Pros**: High accuracy on both occluded version and non-occluded version of FLY3D and KITTI.	KITTI2015, StereoKITTI, FLY3D
	RMS-Flownet[BSMS22]	**Pros**: Hierarchical learning method; Efficient.	StereoKITTI, FLY3D
	WhatMatters⋆ [WHL*22]	**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⋆[DDX*22]	**Pros**: New data-augmentation strategy; Cross-frame feature enhancement; High inference speed.	KITTI2015, FLY3D, Waymo
Correspondences based Methods	FLOT⋆ [PBM20]	**Pros**: Simple and efficient; Addressed transformation challenge. **Cons**: Annotation-hungry; Poor performance on occluded points.	StereoKITTI, FLY3D
	SCTN⋆ [LZGG22]	**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⋆ [WWR*21]	**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⋆ [SM22]	**Pros**: Supervised and self-supervised training fashion; Small model size. **Cons**: Annotation-hungry.	KITTI2015, FLY3D, StereoKITTI
Cost volume based Methods	PointPWC-Net⋆ [WWL*20]	**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 [WHWW21]	**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⋆ [WSF22]	**Pros**: Feature-based attention module; Improved re-weighting mechanism in calculating convolutional weights. **Cons**: Poor performance on occlusions.	StereoKITTI, FLY3D
	Est&Pro⋆ [WS22]	**Pros**: Occlusion-aware; Uncertainty guided network. **Cons**: The overall performance relies on ground-truth occlusion masks.	KITTI2015, FLY3D
Other Methods	HPLFlowNet⋆ [GWW*19]	**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 [LCL*22]	**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

SFPC. SFPC [PHL20] 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 [WWL*20, MOH20]

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.

Table 3: *Summarization of self-supervised/weakly supervised DL architectures for scene flow estimation based on Point Clouds. FLY3D is the abbreviation of FlyingThings3D. ⋆ denotes methods with open-sourced code.*

	Methods	Highlights	Datasets used
Flow embedding based	Just-Go⋆ [MOH20]	**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, LiDAR KITTI, KITTI2018
	SFPC [PHL20]	**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 [ZvBM20]	**Pros**: Addressed deep model generalization challenge; Local structures aware. **Cons**: Suffer from occlusions.	KITTI Object, FLY3D, Lyft
	SFGAN [WJS*22]	**Pros**: Adversarial learning between the scene flow generator and the point cloud discriminator. **Cons**: Suffer from occlusion challenge and LiDAR challenge.	FLY3D, LiDAR KITTI
	OGC⋆ [SY22]	**Pros**: Simultaneous 3D objeccts segmentation and scene flow estimation.	FLY3D, KITTI2015
	Self-Point-Flow⋆ [LLX21]	**Pros**: Combined multiple clues (i.e. colours, surface normal); Addressed annotation challenge; Good generalization ability. **Cons**: Suffer from occlusion challenge.	KITTI2015, FLY3D
	Noisy-Pseudo [LZLG22]	**Pros**: Monocular RGB images and point clouds as data source; Addressed annotation challenge and generalization challenge. **Cons**: Suffer from efficiency challenge.	FLY3D, StereoKITTI, LiDAR KITTI
	Pseudo-LiDAR⋆ [JWMW22]	**Pros**: Solved the LiDAR challenge; Adapted 2D stereo images to 3D scene flow estimation. **Cons**: Suffer from data noise and accuracy challenge.	FLY3D, StereoKITTI, NuScenes, Argoverse
Correspondences based	SCOOP⋆ [LAC*22]	**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 [DZL*22]	**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.	SemanticKITTI, StereoKITTI, Waymo
	RigidFlow [LZL*22]	**Pros**: Enhanced local rigidity in scene flow estimation; Good generalization ability. **Cons**: Failed on non-rigid motion; Suffer from occlusions.	StereoKITTI, FLY3D
	FlowStep3D⋆ [KER21]	**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.	StereoKITTI, FLY3D
	RCP [GTY*22]	**Pros**: Addressed the difference in sampling data challenge; Simultaneous scene flow estimation and point registration. **Cons**: Suffer from efficiency challenge and occlusion challenge.	FLY3D, StereoKITTI, ModelNet40 [WSK*15]
	Rigid3DSF⋆ [GLW*21]	**Pros**: Weakly supervised; Addressed big data challenge and LiDAR challenge; Good performance on different motion fields and occluded points. **Cons**: The estimation of ego-motion relies on soft correspondence; Lack of the similarity measurement of point spatial features.	StereoKITTI, SemanticKITTI, FLY3D
	DCA-SRSFE⋆ [JLA*22]	**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, StereoKITTI
Correspondences free	SLIM ⋆[BEM*21]	**Pros**: Motion-aware; Good generalization to unseen data. **Cons**: The aggregated transform matrix is only suitable for stationary points; Suffer from occlusions.	FLY3D, NuScenes,CARLA, KITTI2018
	Ego-Motion ⋆ [TLOP20]	**Pros**: Hybrid training scheme. **Cons**: Suffer from accuracy and efficiency challenge.	KITTI2015, FLY3D
	Occlusion-G⋆ [OR21b]	**Pros**: Occlusion-weighted cost volume structure; Detection on large motion and occlusions; Addressed LiDAR challenge. **Cons**: Poor generalization ability.	KITTI2015, FLY3D
	PillarML⋆ [LYY21]	**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

PillarML. Stemmed from the merits of motion representation in bird's eye view (BEV), PillarML [LYY21] organizes points into different pillars in vertical order and estimate pillar motion by the velocity residing on each pillar. PillarML [LYY21] consists of LiDAR-based structural consistency, probabilistic motion masking, and a cross-sensor motion regularization module. The pillar motion is estimated from unlabelled 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 [BEM*21] 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 [MOH20, WWL*20], SLIM leverages arbitrary point densities and does not rely on one-to-one correspondences. SLIM is upgraded based on RAFT [TD20] and evaluated on several real datasets: KITTI2018 [MHG18], Nuscenes [CBL*], CARLA [DRC*17], and KITTI-RL [GLSU13].

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 [OR21b] aims to estimate 3D scene flow with occlusions in a self-supervised way. It uses a cost volume structure same as PointPWC-Net [WWL*20], 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 [OR21a] 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 [LZLG22] 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 neighbouring 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. [JLA*22] proposed a mean-teacher framework for unsupervised domain adaptation from synthetic data to real data. DCA-SRSFE [JLA*22] 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 [Kab76]. 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 [GTY*22] decomposes scene flow estimation into two interlaced steps. The first step optimizes 3D flow point-wisely, followed by a recurrent network to optimize 3D flow globally. In the point-wise 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 [GWW*19], Ego-motion [TLOP20] 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 [MIH*16] and KITTI2015 [MHG15, MG15].

RigidFlow. RigidFlow [LZL*22] 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 [BEM*21, PHL20] 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 [LLX21], 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. [JWMW22]. This work can accurately perceives 3D dynamics in 2D images by utilizing a pseudo-LiDAR point cloud as a bridge to compensate for the limitations 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 [SY22] 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.

SCOOP. SCOOP [LAC*22] consists of a self-supervised neural network and an optimization module that work hybridly 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 [LKPL21] 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. [GLW*21] proposed a data-driven method that integrates flow into a higher-level scene abstraction represented by multi rigid-body motion. Rigid3DSF [GLW*21] connects point-wise 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 reasoned 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 [BGM*19] without dense scene flow annotations.

RC-SFE. RC-SFE [DZL*22] 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 [GLW*21] as input. To convert the initial point correspondences status and pre-warped scene flow, RC-SFE applies Kabsch algorithm [Kab76] 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 [GLW*21], RC-SFE relies on the segmentation of background to generate accurate esti-

mation. Dealing with non-rigid motions and occlusions is worthy of further exploration in the future.

5.3. Quantitative analysis

Results of recent deep learning based methods on different datasets along with non-learning methods are tabulated in Table 4 and Table 5. It is hard to declare which approach is the winner compared to others as it depends on the datasets and specific data training scheme they used. We focus on the results generated from the same training dataset and make the following observations.

5.3.1. *Performance on synthetic dataset*

1) PointPWC-Net [WWL*20] is optimized during run time with self-supervised loss through gradient descent. PointPWC-Net and SFPC [PHL20]) have considerable improvements among purely non-learning approaches such as non-rigid iterative closest point (NICP) algorithm [ARV07].
2) On synthetic dataset FlyingThings3D [MIH*16], Bi-PointFlowNet [CK22] and WhatMatters [WHL*22] achieve the best result compared to other fully-supervised learning methods.
3) Its worth noting that self-supervised learning approaches have on-par performance with some fully supervised methods on synthetic dataset, such as recurrent neural network based methods: RCP [GTY*22] and FlowStep3D [KER21]. Since Rigid3DSF [GLW*21] and Ego-motion [TLOP20] contain weak supervision, they have better results than other self-supervised methods [LZL*22, GTY*22, LZLG22].
4) SCTN [LZGG22] introduces a transformer which learns contextual relations between points. As shown in Table 4, the sparse convolution-transformer network proposed in SCTN [LZGG22] is competitive with other methods for improving the accuracy of scene flow estimation.

5.3.2. *Performance on real datasets*

1) Res3DSF [WHWW21] outperforms PV-RAFT [WWR*21] on KITTI2015 dataset. It shows the best performance on the predominant 3D metrics. Compared to the models that are trained with full supervision, WhatMatters [WHL*22] and Bi-PointFlowNet [CK22] achieve better performance than Rigid3DSF [GLW*21], HCRF-Flow [LLH*21], FlowStep3D [KER21], and FLOT [PBM20] on StereoKITTI.
2) According to the evaluation results, FlowStep3D [KER21] and RCP [GTY*22] generalizes well on Stereo KITTI [MHG15, MHG18]. Besides, Self-Point-Flow [LLX21] improves 70% accuracy over Ego-motion [TLOP20] that only uses geometry information (point coordinates). DCA-SRFE [JLA*22] successfully reaches the best performance on StereoKITTI among other self-supervised methods. It even achieves on-par performance with the state-of-the-art methods [GLW*21, GTY*22] trained under full supervision.
3) The accuracy gap remains between the synthetic dataset and the real scene dataset. It is also challenging for both supervised and self-supervised methods.

Table 4: *The quantitative evaluation results on Flyingthings3D [MIH*16]. Self/full indicates the training strategy on FlyingThings3D. Lower values are better for the error metrics including EPE3D and Outliers. Higher values are better for the accuracy metrics including Acc3DS and Acc3DR. All results are compared based on the quantitative results provided by original papers.*

Dataset	Method	Sup.	EPE3D ↓	Acc3DS ↑	Acc3DR ↑	Outliers ↓
Flyingthings3D	PointPWCNet [WWL*20]	non-learning	0.433	0.062	0.195	-
	ICP [BM92]	non-learning	0.412	0.169	0.346	-
	SFPC [PHL20]	non-learning	0.259	0.163	0.416	-
	NICP [ARV07]	non-learning	0.339	0.141	0.357	-
	FlowNet3D [LQG19]	full	0.114	0.413	0.770	0.602
	PointPWC-Net [WWL*20]	full	0.059	0.738	0.928	0.342
	FLOT [PBM20]	full	0.052	0.732	0.927	0.357
	HPLFlowNet [GWW*19]	full	0.080	0.614	0.856	0.429
	RMS-FlowNet [BSMS22]	full	0.056	0.792	0.955	0.324
	FlowStep3D [KER21]	full	0.046	0.816	0.961	0.217
	PointConvFormer [WSF22]	full	0.042	0.865	0.966	0.226
	SCTN [LZGG22]	full	0.038	0.847	0.968	0.268
	RCP [GTY*22]	full	0.040	0.857	0.964	0.198
	HCRF-Flow [LLH*21]	full	0.049	0.834	0.951	0.261
	PV-RAFT [WWR*21]	full	0.046	0.817	0.957	0.292
	SAFIT [SM22]	full	0.050	0.743	0.932	0.346
	HALFlow [WWLW21]	full	0.049	0.785	0.947	0.308
	Res3DSF [WHWW21]	full	0.031	0.914	0.977	0.155
	Bi-PointFlowNet [CK22]	full	0.028	0.918	0.978	0.143
	WhatMatters [WHL*22]	full	**0.028**	**0.929**	**0.981**	**0.146**
	Ego-motion [TLOP20]	hybrid	0.068	0.670	0.879	0.404
	Rigid3DSF [GLW*21]	weakly	**0.052**	**0.746**	**0.936**	**0.361**
	PointPWC-Net [WWL*20]	self	0.125	0.307	0.655	0.703
	SAFIT [SM22]	self	0.171	0.213	0.476	0.756
	FlowStep3D [KER21]	self	0.085	0.536	0.826	0.420
	Self-Point-Flow [LLX21]	self	0.101	0.423	0.775	0.607
	Noisy-pseudo [LZLG22]	self	0.068	0.628	0.881	0.438
	RCP [GTY*22]	self	0.077	0.586	0.860	0.414
	RigidFlow [LZL*22]	self	0.069	0.596	0.871	0.464

4) Up to now, there are only a few methods that conduct experiments on Waymo [SKD*20] dataset and NuScenes dataset [CBL*]. We hope this survey would trigger more attempts in using real datasets to train scene flow estimation network in the future. We refer readers to the specific papers that provide results on Waymo [JLA*22, DDX*22], NuScenes [BEM*21, MOH20], Lyft [ZvBM20, JLA*22] for more details.

6. Applications

Scene flow is one of the most fundamental visual cues in the hierarchy of dynamic scene perception. It provides applicable information for higher-level tasks. The progress in scene flow estimation will refurbish the performance of other scene understanding tasks [GLW*21].

6.1. Point cloud densification

Point clouds are unevenly distributed and sparse. Therefore, some small objects cannot be well-represented by limited number of points. To increase the quality of point clouds, a number of point cloud upsampling methods [YLF*18, QAL*21] have been proposed to generate dense and complete point clouds from the original sparse point cloud. Recent work (e.g., SFPC [PHL20]) discovered that the predicted scene flow can be used to densify the point clouds. SFPC [PHL20] uses five adjacent frames from an Argoverse scene in each direction to densify the current frame. The visual comparison of non-rigid densification proposed by SFPC [PHL20] against the original sparse point cloud and ICP is shown in Figure 6. This visual comparison indicates that SFPC recovers more detailed geometry of objects than ICP.

6.2. Motion segmentation

From pedestrians walking at a constant speed to high-speed vehicles, the issue of detecting objects of interest can be addressed by segmenting the underlying motions. Intuitively, the segmentation of different motion fields is conducted through classifying the point cloud into moving bodies and stationary backgrounds [BEM*21]. Discontinuities in the scene flow are key cues for segmenting a point cloud into several individual objects with different motion fields. Recently, SLIM [BEM*21] proposes a self-supervised learning approach and presents motion segmentation results that illustrate the effectiveness of jointly estimating scene flow and segmenting motion fields. The performance of SLIM is improved with self-supervised motion segmentation signal as it achieves a mIoU score of 59.5% with a sensitivity of 73.1% on KITTI dataset.

Table 5: *The quantitative evaluation results on three versions of KITTI scene flow datasets. Self/full means self-supervised and fully-supervised learning approach.*

Dataset	Method	Sup.	EPE3D ↓	Acc3DS ↑	Acc3DR ↑	Outliers ↓
KITTI2015	PointPWCNet [WWL*20]	non-learning	0.272	0.170	0.357	
[MHG15,	ICP [BM92]	non-learning	0.409	0.052	0.281	
MG15]	SFPC [PHL20]	non-learning	0.093	0.648	0.821	
	NICP [ARV07]	non-learning	0.338	0.221	0.430	
	FlowNet3D [LQG19]	full	0.177	0.374	0.668	0.527
	PV-RAFT [WWR*21]	full	0.056	0.823	0.937	0.216
	Ego-motion [TLOP20]	full	0.103	0.488	0.822	0.394
	SAFIT [SM22]	full	0.048	0.802	0.935	0.218
	Res3DSF [WHWW21]	full	**0.035**	**0.893**	**0.962**	**0.165**
	Ego-motion [TLOP20]	self	0.415	0.221	0.372	0.810
	Self-Point-Flow [LLX21]	self	0.112	0.538	0.794	0.409
	SAFIT [SM22]	self	0.132	0.469	0.708	0.437
StereoKITTI	HPLFlowNet [GWW*19]	full	0.117	0.478	0.778	0.410
[MHG15,	PointPWCNet [WWL*20]	full	0.069	0.728	0.888	0.265
MHG18]	FLOT [PBM20]	full	0.056	0.755	0.908	0.242
	HALFlow [WWLW21]	full	0.062	0.765	0.903	0.249
	HCRF-Flow [LLH*21]	full	0.053	0.863	0.944	0.178
	FlowStep3D [KER21]	full	0.055	0.805	0.925	0.149
	Rigid3DSF [GLW*21]	full	0.042	0.849	0.959	0.208
	RMS-FlowNet [BSMS22]	full	0.053	0.818	0.938	0.203
	RCP [GTY*22]	full	0.048	0.849	0.945	0.123
	PointConvFormer [WSF22]	full	0.048	0.866	0.933	0.173
	WhatMatters [WHL*22]	full	0.031	0.905	0.958	0.161
	Bi-PointFlowNet [CK22]	full	**0.030**	**0.920**	**0.960**	**0.141**
	PointPWCNet [WWL*20]	self	0.255	0.238	0.496	0.686
	FlowStep3D [KER21]	self	0.102	0.708	0.839	0.246
	OGC [SY22]	self	0.067	0.802	0.891	0.226
	RCP [GTY*22]	self	0.076	0.786	0.892	0.185
	RigidFlow [LZL*22]	self	0.062	0.724	0.892	0.262
	Noisy-Pseudo [LZLG22]	self	0.058	0.744	0.898	0.246
	DCA-SRSFE [JLA*22]	self	**0.052**	**0.794**	**0.968**	**0.180**
KITTI2018	SCTN [LZGG22]	full	**0.037**	**0.873**	0.959	**0.179**
[MHG18]	Just-Go [MOH20]	self	0.126	0.320	0.736	-
	SFGAN [WJS*22]	self	**0.098**	**0.302**	0.682	0.558
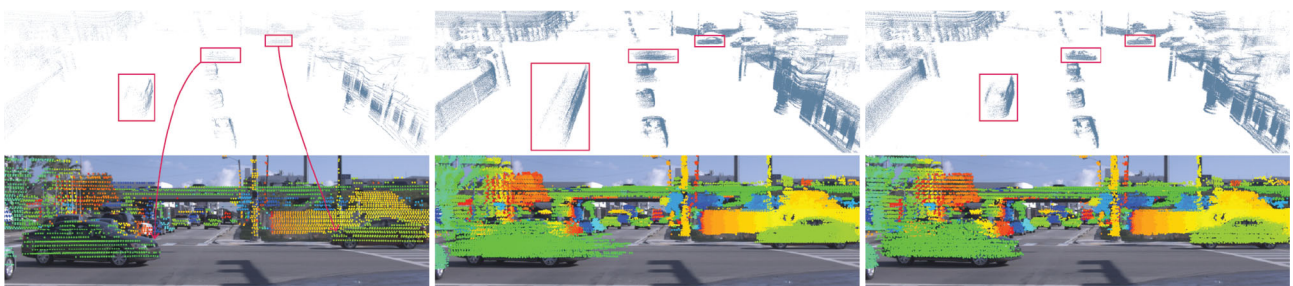
Figure 6: *Application of scene flow approach for point cloud densification, from [PHL20]. The left image is projected from original sparse point cloud collected in Argoverse scene. Middle image represents the densified frame via Iterative Closest Point (ICP) algorithm. The right image is the densification from SFPC [PHL20].*

6.3. LiDAR odometry task

Finding alignments of point clouds to gather motion information between two consecutive LiDAR point clouds is called LiDAR odometry. Approaches that employ ICP [BM92, SHT09] involve three steps: association, transformation, and error evaluation. Yet, it is time consuming and error-prone. Recently, many deep learning-based methods [WSZ*19, LW20] make use of feature correspondences and spatial relationships to LiDAR odometry task. Wang et al. [WWLW21] apply the scene flow to the first point cloud and

solve the 6-DoF pose transformation matrix in a closed-form. With accurate estimation of the scene flow from the point cloud in the first frame to the second frame, HALFNet [WWLW21] obtains better results than the ICP-based method on the LiDAR odometry task. This improvement validates the capability of scene flow to boost the performance of LiDAR odometry task.

6.4. Object tracking

Self-driving can be divided into four separate parts: detection, object tracking, motion forecasting, and motion planning [LYU18]. The objective of object tracking is to identify and locate multiple objects of interest and keep track of their trajectories simultaneously. To enhance the robustness of motion prediction, Flow-Mot [ZKC*20] suggests using the estimated scene flow to compute object-level movement. While most tracking methods adopt a "track-by-detection" approach and utilize the Kalman Filter to avoid having to adjust hyperparameters, FlowMot uses scene flow estimation to obtain 3D motion information that is consistent. Recently, Yang et al. [YJY*22] proposed a novel scene flow based point cloud feature fusion module that leverages temporal information in dynamic 3D point cloud sequences to improve 3D object tracking. These works demonstrate the potential of scene flow to address the challenges faced by current object tracking methods that lack generalization across different datasets.

6.5. 4D vision task

As autonomous vehicles and robotics work in dynamic environment, which indicates they need to interact with the surrounding environment in a period of time. Hence continuous movements in dynamic scenes can be utilized to extract spatial and temporal information, which can be further applied to 4D vision tasks such as 4D semantic segmentation [WLX*22] and 4D acquisition of large scenes. Scene flow estimation provides such spatio-temporal context information of points. It also contributes to 4D point clouds (with additional temporal dimension), where the spatio-temporal neighbourhoods are constructed via the motion of points among several continuous frames. Mustafa et al. [MH20] apply semantic coherence between multiple frames to improve 4D scene flow estimation, co-segmentation and reconstruction [GLW*21]. The application of scene flow in 4D semantic segmentation enables robotic systems to enhance their robustness by leveraging the temporal information from previous frames [shi20]. Recently, the automotive industry is giving increasing attention to 4-D millimeter-wave (mmWave) radar as a rising sensor due to its complementary advantages over LiDAR. RaFlow [DPD*22] provides a new avenue of estimating scene flow, which is specifically designed for radar and calculates the scene flow between two radar point clouds.

7. Potential Research Directions

To address the issue caused by diversified motion fields, there have been quite a few attempts, for example Rigid3DSF [GLW*21] and SLIM [BEM*21] that learn background and foreground motions separately. For the occlusion challenge, Occlusion-G [OR21b], FESTA [WPL*21], and Est&Pro [WS22] explored different mask-

ing operations to reduce the interference of the occluded points. In terms of accuracy, the state-of-the-art supervised method (What-Matters [WHL*22]) improves the accuracy from 41.3% to 92.9% on the FlyingThings3D dataset. Also, several architectures, such as SLIM [BEM*21] and SCTN [LZGG22] still cannot afford the burden of processing a large amount of points. The training time drastically increases as the size of point cloud increased.

Here we provide an overview of promising directions for further research.

7.1. Multi-source and multi-modality data fusion

2D images contain fine-grained information while 3D point clouds provides more geometric details. LiDARs and cameras (e.g. RGB-D camera, monocular camera) are the most common sensors for multi-modal perception in the literature [FHSR*20]. Despite the interest in scene understanding via multi-modality data fusion is growing, only a few papers [LYY21, JWMW22] utilize multi-modality data in scene flow. The effectiveness of data fusion algorithms is restricted by the representation of spatial-temporal information and the ability of CNNs to learn. This is a complex issue that deserves further exploration.

7.2. Multitask learning

An important avenue for future work is to deploy end-to-end multi-task learning (MTL) pipelines. In the field of visual computing, labels are very limited among all kinds of real datasets, and there is still a long way to go to train a robust and accurate learner. MTL which learns task relations from data automatically helps reduce the manual labelling cost for each learning task. A popular example is shown in semantic segmentation and depth estimation [ZY22]. From this perspective, extracting the commonalities from several related tasks for joint learning across tasks is a promising direction to boost performance. As scene flow is inherently a low-level visual cue, it can be integrated with other visual components such as object locations for higher-level scene understanding tasks. Such joint learning enables the model to better cope with complex scene data and improves its self-evolution and self-adaption with multi-task knowledge. Besides, a multi-task learning strategy can even outperform separate models trained independently on each task [CGK18a] and further improve the robustness [MTL22]. As shown in recent works [TWZ*18, CGK18b, HH19, JKBC20], there are considerable attempts to integrate multiple tasks in a unified architecture.

7.3. Domain adaptation

Most current deep learning networks are data-driven. Many state-of-the-art DL models have achieved impressive results. However, those DL models are fine-tuned on a fixed task set. It is still in the beginning to adapt current DL models to different domains. Since 3D annotations usually depend on the annotations obtained from the image domain, it is hard to achieve equal accuracy on a larger dataset. One enlightenment is to use transfer learning. Transfer the knowledge gained from solving one problem (e.g. depth estimation), and apply it to different but related problem such as scene flow estimation. From a broader perspective of self driving and robotic
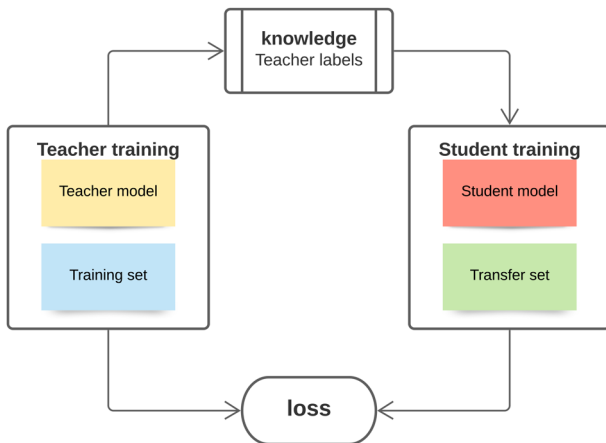
Figure 7: *The architecture of Knowledge Distillation.*

navigation, continuous learning, contrastive learning, and lifelong learning deserve further investigation.

7.4. Semi-supervised learning scheme

Semi-supervised learning is a novel branch of machine learning that leverages unlabelled data to reduce the usage of manual annotations. Jiang et al. [JSJ*19] introduced a compact network (SENSE) that shares common encoder features among optical flow, disparity, occlusion, and semantic segmentation. SENSE [JSJ*19] handles partially labelled data from images very well. To ameliorate the issue of sparse ground-truth annotations of scene flow, SENSE applies a distillation loss and a self-supervised loss to the supervised losses, which forms their semi-supervised loss. The success of semi-supervised learning in the field of optical flow estimation shows that it has the potential to fill the gap between unsupervised learning and supervised learning.

7.5. Knowledge distillation

Collecting large-scale dynamic scene data requires complex calibration. In addition, the cost of transforming the original data into a trainable format is expensive. As a consequence, a labelled dataset for scene flow estimation is very rare. Therefore, applying the knowledge distillation model for training small scene flow estimation networks would be a possible solution for data-hungry networks. In machine learning, knowledge distillation (KD) is the process of compressing the knowledge in a large model into a smaller one. As shown in Figure 7, the traditional knowledge distillation model consists of a teacher model and a student model. In many proposed deep learning models, there are often heavy parameters. Although it's commonly accepted that integrating multiple models and introducing more parameters improves the accuracy of a model, we have to bear high computational costs in the meanwhile [MFL*20]. KD allows training smaller models with minimal loss in performance. The main innovation of KD is that the student network is trained not only via the information provided by true labels but also by observing how the teacher network works with the data. To

our best knowledge, DCA-SRSFE [JLA*22] is the only method that applies the KD model to point-based scene flow estimation so far.

7.6. Efficiency

Deep learning requires expensive GPUs and lots of machines. However, memory and computation resources on board are limited. When it comes to processing large-scale point clouds captured from outdoor scenes, this limitation makes the accurate estimation of scene flow more difficult. The design of convolutional kernels and feature descriptors is the key to balance the efficiency and accuracy of processing 3D data. In spite of the significant improvements of DL models in 3D point cloud learning [QYSG17, WSL*19, ZFF*21], DL models that achieve real-time perception of surrounding dynamics for AVs are still under-explored.

8. Conclusions

This paper reviews the state-of-the-art approaches for scene flow estimation on point clouds within the scope of deep learning paradigms. A comprehensive overview of the challenges in this field is listed. Extensive analyses on supervised, weakly-supervised, and self-supervised scene flow estimation methods are presented. Merits and demerits of these methods are also covered. Moreover, this paper introduces several higher-level scene understanding tasks from the perspective of scene flow estimation and discusses promising research directions. We hope this survey will inspire more research in this field.

References

[ARV07] Amberg B., Romdhani S., Vetter T.: Optimal step non-rigid icp algorithms for surface registration. *2007 IEEE Conference on Computer Vision and Pattern Recognition* (2007), 1–8. doi: https://doi.org/10.1109/CVPR.2007.383165.

[BEM*21] Baur S., Emmerichs D., Moosmann F., Pinggera P., Ommer B., Geiger A.: Slim: Self-supervised lidar scene flow and motion segmentation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE Computer Society, Los Alamitos, CA, USA (Oct 2021), pp. 13106–13116. doi: https://doi.org/10.1109/ICCV48922.2021.01288.

[BGM*19] Behley J., Garbade M., Milioto A., Quenzel J., Behnke S., Stachniss C., Gall J.: Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, (Oct 2019). doi: https://doi.org/10.1109/ICCV.2019.00939.

[BM92] Besl P. J., McKay N. D.: A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence 14*, 2 (1992), 239–256. doi: https://doi.org/10.1109/34.121791.

[BMWR19] BAUR S. A., MOOSMANN F., WIRGES S., RIST C. B.: Real-time 3d lidar flow for autonomous vehicles. In *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, France (2019), pp. 1288–1295. doi: https://doi.org/10.1109/IVS.2019.8814094.

[BSMS22] BATTRAWY R., SCHUSTER R., MAHANI M.-A., STRICKER D.: *Rms-flownet: Efficient and robust multi-scale scene flow estimation for large-scale point clouds*. Institute of Electrical and Electronics Engineers (IEEE), (2022), pp. 883–889. doi: https://doi.org/10.1109/ICRA46639.2022.9811981.

[CBL*] CAESAR H., BANKITI V., LANG A. H., VORA S., LIONG V. E., XU Q., KRISHNAN A., PAN Y., BALDAN G., BEIJBOM O.: nuscenes: A multimodal dataset for autonomous driving. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*. IEEE, Seattle, WA, USA (2020), pp. 11621–11631. doi: https://doi.org/10.1109/CVPR42600.2020.01164.

[CGK18a] CIPOLLA R., GAL Y., KENDALL A.: Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Los Alamitos, CA, USA (Jun 2018), pp. 7482–7491. doi: https://doi.org/10.1109/CVPR.2018.00781.

[CGK18b] CIPOLLA R., GAL Y., KENDALL A.: Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, Los Alamitos, CA, USA (Jun 2018), pp. 7482–7491. doi: https://doi.org/10.1109/CVPR.2018.00781.

[CK22] CHENG W., KO J. H.: Bi-pointflownet: Bidirectional learning for point cloud based scene flow estimation. In *ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVIII*. Springer-Verlag, Berlin, Heidelberg (2022), pp. 108–124. doi: https://doi.org/10.1007/978-3-031-19815-1_7.

[CFG*15] CHANG A. X., FUNKHOUSER T., GUIBAS L., HANRAHAN P., HUANG Q., LI Z., SAVARESE S., SAVVA M., SONG S., SU H., ET AL.: Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012 (2015).

[CLS*19] CHANG M.-F., LAMBERT J., SANGKLOY P., SINGH J., BAK S., HARTNETT A., WANG D., CARR P., LUCEY S., RAMANAN D., et al.: Argoverse: 3D tracking and forecasting with rich maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, CA, USA (2019), pp. 8748–8757. doi: https://doi.org/10.1109/CVPR.2019.00895.

[DDX*22] DING L., DONG S., XU T., XU X., WANG J., LI J.: Fh-net: A fast hierarchical network for scene flow estimation on real-world point clouds. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIX*. Springer, (2022), pp. 213–229.

[DPD*22] DING F., PAN Z., DENG Y., DENG J., LU C. X.: Self-supervised scene flow estimation with 4-D automotive radar.

[DRC*17] DOSOVITSKIY A., ROS G., CODEVILLA F., LÓPEZ A. M., KOLTUN V.: Carla: An open urban driving simulator. *ArXiv abs/1711.03938* (2017).

[DZL*22] DONG G., ZHANG Y., LI H., SUN X., XIONG Z.: Exploiting rigidity constraints for lidar scene flow estimation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Los Alamitos, CA, USA (Jun 2022), pp. 12766–12775. doi: https://doi.org/10.1109/CVPR52688.2022.01244.

[FHSR*20] FENG D., HAASE-SCHUTZ C., ROSENBAUM L., HERTLEIN H., GLÄSER C., TIMM F., WIESBECK W., DIETMAYER K.: Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems PP*. (Feb 2020), 1–20. doi: https://doi.org/10.1109/TITS.2020.2972974.

[GHH*21] GUO Z., HUANG Y., HU X., WEI H., ZHAO B.: A survey on deep learning based approaches for scene understanding in autonomous driving. *Electronics 10*, 4 (2021). doi: https://doi.org/10.3390/electronics10040471.

[GLMH55] GUO M., LIU Z., MU T., HU S.: Beyond self-attention: External attention using two linear layers for visual tasks. *IEEE Transactions on Pattern Analysis & Machine Intelligence, 01* (Oct 5555), 1–13. doi: https://doi.org/10.1109/TPAMI.2022.3211006.

[GLSU13] GEIGER A., LENZ P., STILLER C., URTASUN R.: Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research 32*, 11 (2013), 1231–1237. doi: https://doi.org/10.1177/0278364913491297.

[GLU12] GEIGER A., LENZ P., URTASUN R.: Are we ready for autonomous driving? The kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*. (2012), pp. 3354–3361. doi: https://doi.org/10.1109/CVPR.2012.6248074.

[GLW*21] GOJCIC Z., LITANY O., WIESER A., GUIBAS L. J., BIRDAL T.: Weakly supervised learning of rigid 3D scene flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, Los Alamitos, CA, USA (Jun 2021), pp. 5688–5699. doi: https://doi.org/10.1109/CVPR46437.2021.00564.

[GTY*22] GU X., TANG C., YUAN W., DAI Z., ZHU S., TAN P.: Rcp: Recurrent closest point for scene flow estimation on 3D point clouds. *arXiv preprint arXiv:2205.11028* (2022).

[GWW*19] GU X., WANG Y., WU C., LEE Y., WANG P.: Hplflownet: Hierarchical permutohedral lattice flownet for scene flow estimation on large-scale point clouds. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Los Alamitos, CA, USA (Jun

IEEE Robotics and Automation Letters 7, 3 (2022), 8233–8240. doi: https://doi.org/10.1109/LRA.2022.3187248.

2019), pp. 3249–3258. doi: https://doi.org/10.1109/CVPR.2019.00337.

[HGH*22] HUANG S., GOJCIC Z., HUANG J., WIESER A., SCHINDLER K.: Dynamic 3D scene analysis by point cloud accumulation. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVIII*. Springer-Verlag, Berlin, Heidelberg (2022), pp. 674–690. doi: https://doi.org/10.1007/978-3-031-19839-7_39.

[HH19] HASSANI K., HALEY M.: Unsupervised multi-task feature learning on point clouds. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE Computer Society, Los Alamitos, CA, USA (Nov 2019), pp. 8159–8170. doi: https://doi.org/10.1109/ICCV.2019.00825.

[HR20] HUR J., ROTH S.: Self-supervised monocular scene flow estimation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Los Alamitos, CA, USA (Jun 2020), pp. 7394–7403. doi: https://doi.org/10.1109/CVPR42600.2020.00742.

[HYX*20] HU Q., YANG B., XIE L., ROSA S., GUO Y., WANG Z., TRIGONI N., MARKHAM A.: Randla-net: Efficient semantic segmentation of large-scale point clouds. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Los Alamitos, CA, USA (Jun 2020), pp. 11105–11114. doi: https://doi.org/10.1109/CVPR42600.2020.01112.

[HZB*20] HOUSTON J., ZUIDHOF G., BERGAMINI L., YE Y., JAIN A., OMARI S., IGLOVIKOV V., ONDRUSKA P.: One thousand and one hours: Self-driving motion prediction dataset, (2020). arXiv:2006.14480.

[ISKB18] ILG E., SAIKIA T., KEUPER M., BROX T.: Occlusions, motion and depth boundaries with a generic network for disparity, optical flow or scene flow estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, Munich, Germany (2018), pp. 614–630. doi: https://doi.org/10.1007/978-3-030-01258-8_38.

[JKBC20] JHA A., KUMAR A., BANERJEE B., CHAUDHURI S.: Adamt-net: An adaptive weight learning based multi-task learning model for scene understanding. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. (2020), pp. 3027–3035. doi: https://doi.org/10.1109/CVPRW50498.2020.00361.

[JLA*22] JIN Z., LEI Y., AKHTAR N., LI H., HAYAT M.: Deformation and correspondence aware unsupervised synthetic-to-real scene flow estimation for point clouds. (2022), pp. 7223–7233. doi: https://doi.org/10.1109/CVPR52688.2022.00709.

[JSJ*19] JIANG H., SUN D., JAMPANI V., LV Z., LEARNED-MILLER E., KAUTZ J.: Sense: A shared encoder network for scene-flow estimation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE Computer Society, Los Alamitos, CA, USA (Nov 2019), pp. 3194–3203. doi: https://doi.org/10.1109/ICCV.2019.00329.

[JWMW22] JIANG C., WANG G., MIAO Y., WANG H.: 3D scene flow estimation on pseudo-lidar: Bridging the gap on estimating point motion. IEEE Transactions on Industrial Informatics. (2022).

[JYC*20] JIANG H., YAN F., CAI J., ZHENG J., XIAO J.: End-to-end 3D point cloud instance segmentation without detection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Los Alamitos, CA, USA (Jun 2020), pp. 12793–12802. doi: https://doi.org/10.1109/CVPR42600.2020.01281.

[Kab76] KABSCH W.: A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A 32*, 5 (Sep 1976), 922–923. doi: https://doi.org/10.1107/S0567739476001873.

[KER21] KITTENPLON Y., ELDAR Y. C., RAVIV D.: Flowstep3d: Model unrolling for self-supervised scene flow estimation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Los Alamitos, CA, USA (Jun 2021), pp. 4112–4121. doi: https://doi.org/10.1109/CVPR46437.2021.00410.

[KUH*19] 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., SHET V.: Lyft level 5 av dataset, (2019). URL: https://level5.lyft.com/dataset/.

[LAC*22] LANG I., AIGER D., COLE F., AVIDAN S., RUBINSTEIN M.: Scoop: Self-supervised correspondence and optimization-based scene flow. *arXiv preprint arXiv:2211.14020* (2022). doi: https://doi.org/10.48550/arXiv.2211.14020.

[LCL*22] LU F., CHEN G., LI Z., ZHANG L., LIU Y., QU S., KNOLL A.: Monet: Motion-based point cloud prediction network. *IEEE Transactions on Intelligent Transportation Systems 23*, 8 (2022), 13794–13804. doi: https://doi.org/10.1109/TITS.2021.3128424.

[LKPL21] LI X., KAESEMODEL PONTES J., LUCEY S.: Neural scene flow prior. *Advances in Neural Information Processing Systems 34*, (2021), 7838–7851.

[LLH*21] LI R., LIN G., HE T., LIU F., SHEN C.: Hcrf-flow: Scene flow from point clouds with continuous high-order crfs and position-aware flow embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, Los Alamitos, CA, USA (Jun 2021), pp. 364–373. doi: https://doi.org/10.1109/CVPR46437.2021.00043.

[LLW*20] LIU J., LI H., WU R., ZHAO Q., GUO Y., CHEN L.: A survey on deep learning methods for scene flow estimation. *Pattern Recognition 106* (2020), 107378. doi: https://doi.org/10.1016/j.patcog.2020.107378.

[LLX21] LI R., LIN G., XIE L.: Self-point-flow: Self-supervised scene flow estimation from point clouds with optimal transport and random walk. (2021), pp. 15572–15581. doi: https://doi.org/10.1109/CVPR46437.2021.01532.

[LQG19] Liu X., Qi C. R., Guibas L. J.: Flownet3d: Learning scene flow in 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, California, USA (2019), pp. 529–537. doi: https://doi.org/10.1109/CVPR.2019.00062.

[LW20] Li Z., Wang N.: Dmlo: Deep matching lidar odometry. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE Press, Las Vegas, NV, USA (2020), pp. 6010–6017. doi: https://doi.org/10.1109/IROS45743.2020.9341206.

[LYB19] Liu X., Yan M., Bohg J.: Meteornet: Deep learning on dynamic 3d point cloud sequences. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE Computer Society, Los Alamitos, CA, USA (2019), pp. 9245–9254. doi: https://doi.org/10.1109/ICCV.2019.00934.

[LYU18] Luo W., Yang B., Urtasun R.: Fast and furious: Real time end-to-end 3D detection, tracking and motion forecasting with a single convolutional net. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. IEEE, Utah, USA (2018), pp. 3569–3577. doi: https://doi.org/10.1109/CVPR.2018.00376.

[LYY21] Luo C., Yang X., Yuille A.: Self-supervised pillar motion learning for autonomous driving. (2021), pp. 3182–3191. doi: https://doi.org/10.1109/CVPR46437.2021.00320.

[LZGG22] Li B., Zheng C., Giancola S., Ghanem B.: Sctn: Sparse convolution-transformer network for scene flow estimation. (2022), pp. 1254–1262.

[LZL*22] Li R., Zhang C., Lin G., Wang Z., Shen C.: Rigidflow: Self-supervised scene flow learning on point clouds by local rigidity prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, Los Alamitos, CA, USA (June 2022), pp. 16959–16968.

[LZLG22] Li B., Zheng C., Li G., Ghanem B.: Learning scene flow in 3d point clouds with noisy pseudo labels. *arXiv preprint arXiv:2203.12655* (2022).

[MFL*20] Mirzadeh S. I., Farajtabar M., Li A., Levine N., Matsukawa A., Ghasemzadeh H.: Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI Conference on Artificial Intelligence*. New York, USA (2020), vol. *34*, pp. 5191–5198. doi: https://doi.org/10.1609/aaai.v34i04.5963.

[MG15] Menze M., Geiger A.: Object scene flow for autonomous vehicles. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. IEEE, Boston, MA, USA (2015), pp. 3061–3070. doi: https://doi.org/10.1109/CVPR.2015.7298925.

[MGCF19] Maretic H. P., Gheche M. E., Chierchia G., Frossard P.: Got: An optimal transport framework for graph comparison. *Advances in Neural Information Processing Systems* (2019).

[MH20] Mustafa A., Hilton A.: Semantically coherent 4d scene flow of dynamic scenes. *International Journal of Computer Vision 128*, 2 (2020), 319–335. doi: https://doi.org/10.1007/s11263-019-01241-w.

[MHG15] Menze M., Heipke C., Geiger A.: Joint 3d estimation of vehicles and scene flow. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* II-3/W5, (Aug 2015), 427–434. doi: https://doi.org/10.5194/isprsannals-II-3-W5-427-2015.

[MHG18] Menze M., Heipke C., Geiger A.: Object scene flow. *ISPRS Journal of Photogrammetry and Remote Sensing 140*, (2018), 60–76. doi: https://doi.org/10.1016/j.isprsjprs.2017.09.013.

[MIH*16] Mayer N., Ilg E., Hausser P., Fischer P., Cremers D., Dosovitskiy A., Brox T.: A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. IEEE, Las Vegas, NV, USA (2016), pp. 4040–4048. doi: https://doi.org/10.1109/CVPR.2016.438.

[MOH20] Mittal H., Okorn B., Held D.: Just go with the flow: Self-supervised scene flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, Washington, USA (2020), pp. 11177–11185. doi: https://doi.org/10.1109/CVPR42600.2020.01119.

[MTL22] Multi-task learning for dense prediction tasks: A survey. *IEEE Transactions on Pattern Analysis & Machine Intelligence 44*, 07 (Jul 2022), 3614–3633. doi: https://doi.org/10.1109/TPAMI.2021.3054719.

[MWB21] Mohammadi S. S., Wang Y., Bue A. D.: Pointviewgcn: 3d shape classification with multi-view point clouds. In *2021 IEEE International Conference on Image Processing (ICIP)*. IEEE, (2021), pp. 3103–3107. doi: https://doi.org/10.1109/ICIP42928.2021.9506426.

[NMV17] Nikolentzos G., Meladianos P., Vazirgiannis M.: Matching node embeddings for graph similarity. In *Proceedings of the AAAI Conference on Artificial Intelligence*. California, USA (2017), vol. *31*.

[OR21a] Ouyang B., Raviv D.: Occlusion guided scene flow estimation on 3d point clouds. (2021), pp. 2799–2808. doi: https://doi.org/10.1109/CVPRW53098.2021.00315.

[OR21b] Ouyang B., Raviv D.: Occlusion guided self-supervised scene flow estimation on 3D point clouds. In *2021 International Conference on 3D Vision (3DV)*. IEEE, (2021), pp. 782–791.

[PBM20] Puy G., Boulch A., Marlet R.: *FLOT: Scene Flow on Point Clouds Guided by Optimal Transport*. 2020, pp. 527–544. doi: https://doi.org/10.1007/978-3-030-58604-1_32.

[PHL20] PONTES J. K., HAYS J., LUCEY S.: Scene flow from point clouds with or without learning. (2020), pp. 261–270. doi: https://doi.org/10.1109/3DV50981.2020.00036.

[QAL*21] QIAN G., ABUALSHOUR A., LI G., THABET A., GHANEM B.: Pu-gcn: Point cloud upsampling using graph convolutional networks. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Los Alamitos, CA, USA (Jun 2021), pp. 11678–11687. doi: https://doi.org/10.1109/CVPR46437.2021.01151.

[QCLG20] QI C. R., CHEN X., LITANY O., GUIBAS L. J.: Imvotenet: Boosting 3d object detection in point clouds with image votes. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Los Alamitos, CA, USA (Jun 2020), pp. 4403–4412. doi: https://doi.org/10.1109/CVPR42600.2020.00446.

[QSMG17] QI C. R., SU H., MO K., GUIBAS L. J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Hawaii, USA (2017), pp. 652–660. doi: https://doi.org/10.1109/CVPR.2017.16.

[QYSG17] QI C. R., YI L., SU H., GUIBAS L. J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17, Curran Associates Inc., Red Hook, NY, USA (2017), pp. 5105–5114. doi: https://doi.org/10.5555/3295222.3295263.

[SHHX18] SHEN H., HUANG L., HUANG C., XU W.: Tracklet association tracker: An end-to-end learning-based association approach for multi-object tracking. *arXiv preprint arXiv:1808.01562* (2018).

[shi20] Spsequencenet: Semantic segmentation network on 4d point clouds. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Los Alamitos, CA, USA (Jun 2020), pp. 4573–4582. doi: https://doi.org/10.1109/CVPR42600.2020.00463.

[SHT09] SEGAL A., HÄHNEL D., THRUN S.: Generalized-icp. doi: https://doi.org/10.15607/RSS.2009.V.021.

[SKD*20] SUN P., KRETZSCHMAR H., DOTIWALLA X., CHOUARD A., PATNAIK V., TSUI P., GUO J., ZHOU Y., CHAI Y., CAINE B., et al.: Scalability in perception for autonomous driving: Waymo open dataset. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Los Alamitos, CA, USA (Jun 2020), pp. 2443–2451. doi: https://doi.org/10.1109/CVPR42600.2020.00252.

[SM22] SHI Y., MA K.: Safit: Segmentation-aware scene flow with improved transformer. In *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, (2022), pp. 10648–10655. doi: https://doi.org/10.1109/ICRA46639.2022.9811747.

[SY22] SONG Z., YANG B.: Ogc: Unsupervised 3D object segmentation from rigid dynamics of point clouds. *arXiv preprint arXiv:2210.04458* (2022).

[TD20] TEED Z., DENG J.: Raft: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision*. Springer, (2020), pp. 402–419. doi: https://doi.org/10.1007/978-3-030-58536-5_24.

[TLOP20] TISHCHENKO I., LOMBARDI S., OSWALD M. R., POLLEFEYS M.: Self-supervised learning of non-rigid residual flow and ego-motion. In *2020 International Conference on 3D Vision (3DV)*, IEEE, Fukuoka, Japan (2020), pp. 150–159. doi: https://doi.org/10.1109/3DV50981.2020.00025.

[TWZ*18] TEICHMANN M., WEBER M., ZÖLLNER M., CIPOLLA R., URTASUN R.: Multinet: Real-time joint semantic reasoning for autonomous driving. In *2018 IEEE Intelligent Vehicles Symposium (IV)*. (2018), pp. 1013–1020. doi: https://doi.org/10.1109/IVS.2018.8500504.

[VBR*99] VEDULA S., BAKER S., RANDER P., COLLINS R., KANADE T.: Three-dimensional scene flow. In *Proceedings of the Seventh IEEE International Conference on Computer Vision* (1999), vol. 2, pp. 722–729. doi: https://doi.org/10.1109/ICCV.1999.790293.

[Vil09] VILLANI C.: *Optimal transport: old and new*. Springer, (2009), vol. 338.

[WHL*22] WANG G., HU Y., LIU Z., ZHOU Y., TOMIZUKA M., ZHAN W., WANG H.: *What Matters for 3D Scene Flow Network*. (2022), pp. 38–55.

[WHWW21] WANG G., HU Y., WU X., WANG H.: Residual 3d scene flow learning with context-aware feature extraction. *arXiv preprint arXiv:2109.04685* (2021).

[WJS*22] WANG G., JIANG C., SHEN Z., MIAO Y., WANG H.: Sfgan: Unsupervised generative adversarial learning of 3d scene flow from the 3d scene self. *Advanced Intelligent Systems 4*, 4 (2022), 2100197. doi: https://doi.org/10.1002/aisy.202100197.

[WLHJ*20] WANG Z., LI S., HOWARD-JENKINS H., PRISACARIU V., CHEN M.: Flownet3d++: Geometric losses for deep scene flow estimation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. IEEE, Snowmass Village, CO, USA (2020), pp. 91–98. doi: https://doi.org/10.1109/WACV45572.2020.9093302.

[WLX*22] WEI Y., LIU H., XIE T., KE Q., GUO Y.: Spatial-temporal transformer for 3D point cloud sequences. (2022), pp. 657–666. doi: https://doi.org/10.1109/WACV51458.2022.00073.

[WPL*21] WANG H., PANG J., LODHI M. A., TIAN Y., TIAN D.: Festa: Flow estimation via spatial-temporal attention for scene point clouds. (2021), pp. 14168–14177. doi: https://doi.org/10.1109/CVPR46437.2021.01395.

[WS22] WANG K., SHEN S.: Estimation and propagation: Scene flow prediction on occluded point clouds. *IEEE Robotics and Automation Letters 7*, (Oct 2022), 12201–12208. doi: https://doi.org/10.1109/LRA.2022.3215019.

[WSF22] WU W., SHAN Q., FUXIN L.: Pointconvformer: Revenge of the point-based convolution. *arXiv preprint arXiv:2208.02879* (2022).

[WSK*15] Wu Z., Song S., Khosla A., Yu F., Zhang L., Tang X., Xiao J.: 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* IEEE Computer Society, Boston, MA, USA (2015), pp. 1912–1920. doi: https://doi.org/10.1109/CVPR.2015.7298801.

[WSL*19] Wang Y., Sun Y., Liu Z., Sarma S. E., Bronstein M. M., Solomon J. M.: Dynamic graph CNN for learning on point clouds. *ACM Transactions on Graphics (tog) 38*, 5 (2019), 1–12. doi: https://doi.org/10.1145/3326362.

[WSZ*19] Wang W., Saputra M. R. U., Zhao P., Gusmao P., Yang B., Chen C., Markham A., Trigoni N.: Deeppco: End-to-end point cloud odometry through deep parallel neural network. (2019), pp. 3248–3254. doi: https://doi.org/10.1109/IROS40897.2019.8967756.

[WWL*20] Wu W., Wang Z. Y., Li Z., Liu W., Fuxin L.: Pointpwc-net: Cost volume on point clouds for (self-) supervised scene flow estimation. In *European Conference on Computer Vision.* Springer, (2020), pp. 88–107. doi: https://doi.org/10.1007/978-3-030-58558-7_6.

[WWLW21] Wang G., Wu X., Liu Z., Wang H.: Hierarchical attention learning of scene flow in 3d point clouds. *IEEE Transactions on Image Processing PP*, (May 2021), 1–1. doi: https://doi.org/10.1109/TIP.2021.3079796.

[WWR*21] Wei Y., Wang Z., Rao Y., Lu J., Zhou J.: Pv-raft: Point-voxel correlation fields for scene flow estimation of point clouds. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* IEEE Computer Society, Los Alamitos, CA, USA (Jun 2021), pp. 6950–6959. doi: https://doi.org/10.1109/CVPR46437.2021.00688.

[XAZX17] Xuezhi X., Ali S. M., Zhai M., Xiao D.: Scene flow estimation methodologies and applications—A review. In *2017 29th Chinese Control And Decision Conference (CCDC).* IEEE, (2017), pp. 5424–5429. doi: https://doi.org/10.1109/CCDC.2017.7979461.

[YJY*22] Yang Y., Jiang K., Yang D., Jiang Y., Lu X.: Temporal point cloud fusion with scene flow for robust 3D object tracking. *IEEE Signal Processing Letters 29*, (2022), 1579–1583. doi: https://doi.org/10.1109/LSP.2022.3185948.

[YLF*18] Yu L., Li X., Fu C., Cohen-Or D., Heng P.: Pu-net: Point cloud upsampling network. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* IEEE Computer Society, Los Alamitos, CA, USA (Jun 2018), pp. 2790–2799. doi: https://doi.org/10.1109/CVPR.2018.00295.

[YX16] Yan Z., Xiang X.: Scene flow estimation: A survey. *ArXiv abs/1612.02590* (2016).

[ZCL20] Zhao N., Chua T., Lee G.: Sess: Self-ensembling semi-supervised 3d object detection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* IEEE Computer Society, Los Alamitos, CA, USA (Jun 2020), pp. 11076–11084. doi: https://doi.org/10.1109/CVPR42600.2020.01109.

[ZFF*21] Zhou H., Feng Y., Fang M., Wei M., Qin J., Lu T.: Adaptive graph convolution for point cloud analysis. In *2021 IEEE/CVF International Conference for Computer Vision (ICCV).* IEEE Computer Society (Oct 2021), pp. 4945–4954. https://doi.org/10.1109/ICCV48922.2021.00492.

[ZHZ*19] Zou C., He B., Zhu M., Zhang L., Zhang J.: Learning motion field of lidar point cloud with convolutional networks. *Pattern Recognition Letters 125*, (2019), 514–520. doi: https://doi.org/10.1016/j.patrec.2019.06.009.

[ZKC*20] Zhai G., Kong X., Cui J., Liu Y., Yang Z.: Flowmot: 3D multi-object tracking by scene flow association. *arXiv preprint arXiv:2012.07541* (2020).

[ZvVBM20] Zuanazzi V., van Vugt J., Booij O., Mettes P.: Adversarial self-supervised scene flow estimation. In *2020 International Conference on 3D Vision (3DV).* IEEE, Fukuoka, Japan (2020), pp. 1049–1058. doi: https://doi.org/10.1109/3DV50981.2020.00115.

[ZXLK21] Zhai M., Xiang X., Lv N., Kong X.: Optical flow and scene flow estimation: A survey. *Pattern Recognition.* (2021), pp. 107861. doi: https://doi.org/10.1016/j.patcog.2021.107861.

[ZY22] Zhang Y., Yang Q.: A survey on multi-task learning. *IEEE Transactions on Knowledge & Data Engineering 34*, 12 (Dec 2022), 5586–5609. doi: https://doi.org/10.1109/TKDE.2021.3070203.

[ZZtZX20] Zhang J., Zhu C., Tao Zheng L., Xu K.: Fusion-aware point convolution for online semantic 3d scene segmentation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* (Jun 2020), 4533–4542. doi: https://doi.org/10.1109/CVPR42600.2020.00459.