3DSFLabelling: Boosting 3D Scene Flow Estimation by Pseudo Auto-labelling

Chaokang Jiang[1], Guangming Wang[2], Jiuming Liu[3], Hesheng Wang[3], Zhuang Ma[1],
Zhenqiang Liu[1], Zhujin Liang[1], Yi Shan[1], Dalong Du[1†]
[1]PhiGent Robotics, [2]University of Cambridge, [3]Shanghai Jiaotong University
ts20060079a31@cumt.edu.cn, gw462@cam.ac.uk, {liujiuming, wanghesheng}@sjtu.edu.cn,
mazhuang097@outlook.com, {zhenqiang.liu, zhujin.liang, yi.shan, dalong.du}@phigent.ai
jiangchaokang.github.io/3DSFLabelling-Page

Abstract

Learning 3D scene flow from LiDAR point clouds presents significant difficulties, including poor generalization from synthetic datasets to real scenes, scarcity of real-world 3D labels, and poor performance on real sparse LiDAR point clouds. We present a novel approach from the perspective of auto-labelling, aiming to generate a large number of 3D scene flow pseudo labels for real-world LiDAR point clouds. Specifically, we employ the assumption of rigid body motion to simulate potential object-level rigid movements in autonomous driving scenarios. By updating different motion attributes for multiple anchor boxes, the rigid motion decomposition is obtained for the whole scene. Furthermore, we developed a novel 3D scene flow data augmentation method for global and local motion. By perfectly synthesizing target point clouds based on augmented motion parameters, we easily obtain lots of 3D scene flow labels in point clouds highly consistent with real scenarios. On multiple real-world datasets including LiDAR KITTI, nuScenes, and Argoverse, our method outperforms all previous supervised and unsupervised methods without requiring manual labelling. Impressively, our method achieves a tenfold reduction in EPE3D metric on the LiDAR KITTI dataset, reducing it from $0.190m$ *to a mere* $0.008m$ *error.*

1. Introduction

3D scene flow estimation through deducing per-point motion filed from consecutive frames of point clouds, serves a critical role across various applications, encompassing motion prediction [33, 48], anomaly motion detection [15], 3D object detection [8, 16, 50], and dynamic point cloud accumulation [14]. With the advancing of deep learning on point clouds [37, 38], many works [4, 9, 18, 27, 36, 39, 51] have developed the learning-based methods to estimate per-point motion from 3D point clouds. Some state-of-the-art methods [4, 39, 51] have reduced the average 3D EndPoint Error (EPE3D) to a few centimetres on the KITTI Scene Flow dataset (stereoKITTI) [30, 31]. However, due to the scarcity of scene flow labels, these methods rely heavily on synthetic datasets such as FlyingThings3D (FT3D) [29] for network training.

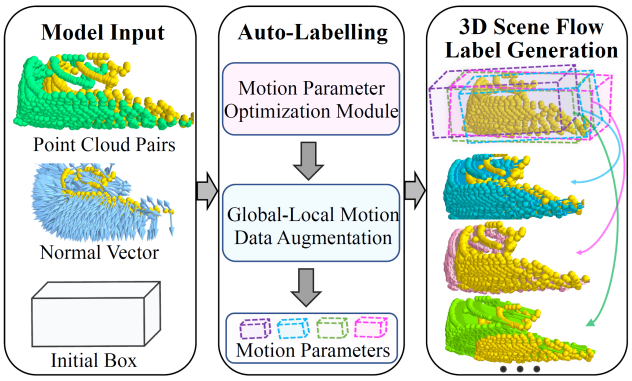

Figure 1. The proposed 3D scene flow pseudo-auto-labelling framework. Given point clouds and initial bounding boxes, both global and local motion parameters are iteratively optimized. Diverse motion patterns are augmented by randomly adjusting these motion parameters, thereby creating a diverse and realistic set of motion labels for the training of 3D scene flow estimation models.

When evaluated on the stereoKITTI dataset [30, 31], PV-RAFT [47] demonstrates an average EPE3D of just $0.056m$. However, when evaluated on the Argoverse dataset [3], the EPE3D metric astonishingly exceeds $10m$ [24]. Therefore, learning 3D scene flow on synthetic dataset [29] has a large gap with real-world application. Jin et al. [18] recently introduce a new synthetic dataset, GTA-SF, simulating LiDAR scans for autonomous driving. They propose a teacher-student domain adaptation framework to reduce the gap between synthetic and real datasets and improve some performance of 3D scene flow estimation. However, their performance is still poor in

[1†]Corresponding author.

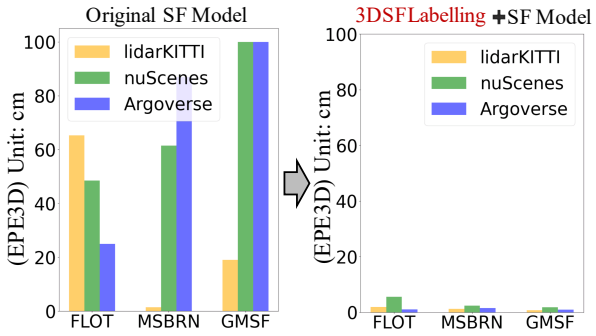

Figure 2. The accuracy improvement after integrating our proposed pseudo-auto-labelling method. Models trained on synthetic data performance poorly in 3D scene flow estimation for LiDAR-based autonomous driving. Our proposed 3D pseudo-auto-labelling method improves accuracy, reaching an EPE3D below $2cm$ across datasets [2, 3, 31].

real-world LiDAR data because of ideal sensor models and lack of scene variety. Ideally, models should learn from real sensor data in the autonomous driving field. However, labelling each point's 3D motion vector for the 3D scene flow task is extremely costly. This has driven many works [6, 22, 28, 32, 39, 44] towards unsupervised or self-supervised learning of 3D scene flow. Although these methods have achieved reasonable accuracy, they still fall behind supervised methods, highlighting the importance of real sensor data and corresponding 3D scene flow labels.

In this work, we address three key challenges in the field of autonomous driving: the reliance on synthetic datasets that still have a poor generalization with real-world scenarios, the scarcity of scene flow labels in actual driving scenes, and the poor performance of existing 3D scene flow estimation networks on real LiDAR data. Inspired by the rigid motion assumptions in RigidFlow [22] and RSF [5], we propose a novel scene flow auto-labelling approach that leverages the characteristics of rigid motion prevalent in autonomous driving scenarios (Fig. 1). Specifically, we utilize 3D anchor boxes to segment 3D objects in point clouds. The attributes of each object-level box are not only position and size but also rotation, translation, motion status, and normal vector attributes. By leveraging the constrained loss functions for the box parameters and inter-frame association, we optimize the attributes of the boxes, subsequently combining these parameters with the source point cloud to produce a realistic target point cloud. Importantly, the generated target point cloud maintains a one-to-one correspondence with the source point cloud, enabling the efficient generation of pseudo 3D scene flow labels.

To capture a more diverse range of motion patterns, we introduce a novel data augmentation strategy for 3D scene flow auto-labelling. Utilizing the attributes of each box, we simulate the rotations, translations, and motion status of both the ego vehicle and surrounding environment by adding Gaussian noise to these attributes. Consequently, we obtain numerous 3D scene flow labels with diverse motions that closely resemble real-world scenarios, furnishing the neural network with rich real training data and significantly improving the generalization capabilities of learning-based methods. Experimental results validate that our pseudo-label generation strategy consistently achieves state-of-the-art scene flow estimation results across various models [4, 36, 51] and datasets [2, 3, 30] (Fig. 2).

In summary, our contributions are as follows:

- We propose a new framework for the automatic labelling of 3D scene flow pseudo-labels, significantly enhancing the accuracy of current scene flow estimation models, and effectively addressing the scarcity of 3D flow labels in autonomous driving.
- We propose a universal 3D box optimization method with multiple motion attributes. Building upon this, we further introduce a plug-and-play 3D scene flow augmentation module with global-local motions and motion status. This allows for flexible motion adjustment of ego-motion and dynamic environments, setting a new benchmark for scene flow data augmentation.
- Our method achieves state-of-the-art performance on KITTI, nuScenes, and Argoverse LiDAR datasets. Impressively, our approach surpasses all supervised and unsupervised methods without requiring any synthesising data and manual scene flow labels.

2. Related Work

2.1. Supervised 3D Scene Flow Learning

In recent years, the performance of methods [28, 34, 42] for 3D scene flow based on point cloud deep learning has surpassed traditional methods. FlowNet3D [28] pioneers an end-to-end approach to learning 3D scene flow from point clouds. Some works, such as HALFlow [13], 3DFlow [43], PointPWC [49], and WSAFlowNet [45], utilize PWC structures to learn 3D scene flow in a coarse-to-fine manner. Other methods address the disorderliness of points by voxelizing point clouds and using sparse convolution or voxel correlation fields to learn 3D scene flow, such as PV-RAFT [47], DPV-RAFT [46], and SCTN [21]. Additional work refines the estimated scene flow through iterative procedures. MSBRN [4] proposes bidirectional gated recurrent units for iteratively estimating scene flow. GMSF [51] and PT-FlowNet [9] introduce point cloud transformers into 3D scene flow estimation networks. These supervised learning methods for 3D scene flow heavily rely on ground truth and are all trained on the FT3D dataset [29] and evaluated on stereoKITTI [30, 31] for network generalization test.

2.2. Unsupervised 3D Scene Flow Learning

JGwF [32] and PointPWC [49] initially propose several self-supervised learning losses such as cycle consistency loss and chamfer loss. EgoFlow [40] distinguishes 3D scene flow into ego-motion flow and remaining non-rigid flow, achieving self-supervised learning based on temporal consistency. SFGAN [44] introduces generative adversarial concepts into self-supervised learning for 3D scene flow. Recently, works like R3DSF [12], RigidFlow [22], and LiDARSceneFlow [7] greatly improve the accuracy of 3D scene flow estimation by introducing local or object-level rigidity constraints. RigidFlow [22] explicitly enforces rigid alignment within super-voxel regions by decomposing the source point cloud into multiple super-voxels. R3DSF [12] separately considers background and foreground object-level 3D scene flow, relying on segmentation and odometry tasks [25, 26].

2.3. 3D Scene Flow Optimization

3D scene flow optimization techniques have demonstrated remarkable generalization capabilities, attracting a significant amount of academic research recently. Graph prior [35] optimizes scene flow to be as smooth as possible by using the Laplacian of point clouds. Some techniques introduce neural networks to optimize 3D scene flow. NSFP [23] introduces a novel implicit regularizer, the Neural Scene Flow Prior, which primarily depends on runtime optimization and robust regularization. RSF [5] combines global ego-motion with object-specific rigid movements to optimize 3D bounding box parameters and compute scene flow. FastNSF [24] also adopts neural scene flow prior, and it shows more advantages in dealing with dense LiDAR points compared to learning methods. SCOOP [20], in the runtime phase, directly optimizes the flow refinement module using self-supervised objectives. Although optimization-based approaches for 3D scene flow estimation have demonstrated impressive accuracy, they typically involve high computational costs.

3. 3DSFLabelling

3D scene flow estimation infers the 3D flow, $SF_{pred} \in \mathbb{R}^{3\times N_1}$ from the source point cloud $PC_S \in \mathbb{R}^{3\times N_1}$ and the target point cloud $PC_T \in \mathbb{R}^{3\times N_2}$ for each point in the source point. Previous self-supervised learning methods [32, 49] typically use the estimated 3D motion vector SF_{pred} to warp the source point cloud PC_S to the target point cloud PC_{Sw}. By comparing the difference between PC_{Sw} and PC_T, a supervisory signal is generated.

In contrast with previous self-supervised learning methods, we propose bounding box element optimization to obtain the boxes and the box motion parameters from raw unlabelled point cloud data. Then, we use object-box-level motion parameters and global motion parameters to warp each box's points and the whole point cloud to the target point cloud, generating corresponding pseudo 3D scene flow labels. During the warping process of each object box, we propose augmenting the motion attributes of each object and the whole scene. This diversity assists the network in capturing a broader range of motion behaviours.

3.1. Prerequisites

Apart from the two input point clouds, we do not require any extra labels, such as object-level tracking and semantic information, or vehicle ego-motion labels. To reinforce the geometric constraints in the pseudo label generation module, we employ Open3d [52] to generate coarse per-point normals. Despite these normals not being perfectly accurate, they are readily obtainable and can provide useful geometric constraints. Finally, we establish initial 3D anchor boxes with specific centers (x, y, z), width w, length l, height h, and rotation angle θ, in accordance with the range of input points. As depicted in Fig. 3, the inputs of our model consist of the initial anchor box set, PC_S, PC_T, and point cloud normals N_S.

3.2. Motion Parameter Optimization Module

As shown in Fig. 3, we present the process of simulating the motion of point clouds in actual autonomous driving by updating four sets of parameters: differentiable bounding boxes $\Phi = [c, s, \theta]$, global motion parameters $\Theta = [R_{ego}, t_{ego}]$, motion parameters for each box $[R_{perbox}, t_{perbox}]$, and motion probability P_M for each box. The variables c, s, and θ represent the center coordinates, size, and orientation of the 3D box, respectively.

Inspired by RSF [5], we use the motion of object-level bounding boxes to present the point-wise 3D motion and make the step-like boxes differentiable through sigmoid approximation. By transforming the individual points to the bounding boxes, we introduce an object-level perception of the scene, enabling a more natural capture of rigid motion. This method proves advantageous in autonomous driving scenarios, where most objects predominantly exhibit rigid behaviour [12]. Additionally, in the context of autonomous driving, most scene motion is typically caused by the ego motion of the vehicle. Hence, setting global motion parameters is necessary to simulate the global consistent rigid motion of the whole scene. To discern whether the motion of each box is caused by ego-motion, we also set up a motion probability for each bounding box.

With the initial set of four motion parameters, the source point cloud is warped to the target frame, as follows:

$$PC_T^{\Theta}, PC_T^{\Phi} = \Omega_1(\Theta, PC_S), \Omega_2(\Upsilon(\Phi, PC_S)), \quad (1)$$

where Θ represents global motion parameters. Φ represents motion parameters of each bounding box, and Ω_1 and Ω_2

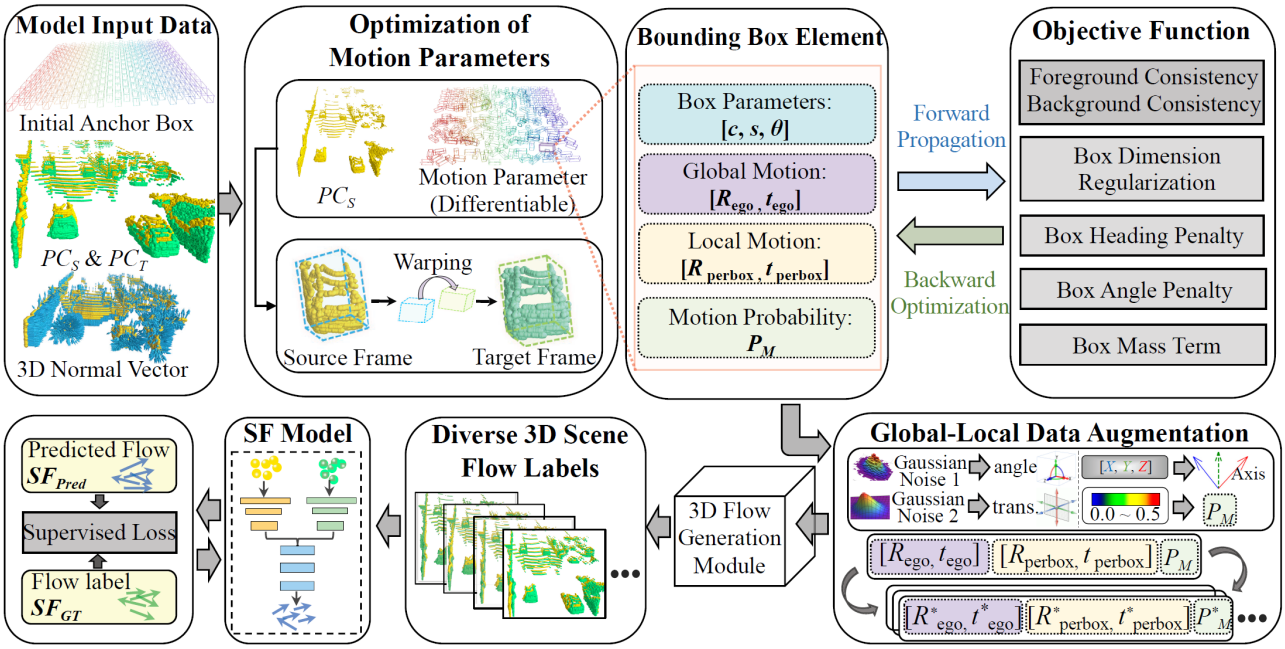

Figure 3. The proposed learning framework of pseudo 3D scene flow automatic labelling. The input comprises 3D anchor boxes, a pair of point clouds, and their corresponding coarse normal vectors. The optimization of motion parameters primarily updates the bounding box parameters, global motion parameters, local motion parameters, and the motion probability of the boxes. The attribute parameters for boxes are updated through backward optimization from six objective functions. Once optimized, the motion parameters simulate various types of motion using a global-local data augmentation module. A single source frame point cloud, along with the augmented motion parameters, produces diverse 3D scene flow labels. These labels serve to guide the supervised neural network to learn point-wise motion.

are background and foreground warping functions, respectively, generating the warped point clouds PC_T^{Θ} and PC_T^{Φ}. Υ signifies the removal of boxes with too few points.

Based on the real target frame of point cloud and the generated target point clouds PC_T^{Θ} and PC_T^{Φ}, we define loss functions to update and optimize the box attributes. We separately calculate the background and foreground losses:

$$L_{BG} = \kappa(N_T^{\Theta} \oplus PC_T^{\Theta}, N_T \oplus PC_T) + \delta(PC_T^{\Theta}, PC_T), \quad (2)$$

$$L_{FG} = \frac{1}{K_{box}} \sum P_M \times (\kappa(N_T^{\Phi} \oplus PC_T^{\Phi}, N_T \oplus PC_T) + \delta(PC_T^{\Phi}, PC_T)), \quad (3)$$

where κ is a function calculating nearest neighbour matches between the transformed point cloud and the target point cloud. δ is a pairwise distance function with location encoding. K_{box} is the number of boxes, P_M is the motion probability of each box, and the term $N_T \oplus PC_T$ represents the concatenation of the target point cloud's normal and positions. As for the motion probability P_M of each box:

$$P_M = \sigma(\alpha \times (\Omega_3(\Phi, \gamma_i) + \beta_i)) - \alpha \times (\Omega_3(\Phi, \gamma_i) - \beta_i)), \quad (4)$$

where $\sigma(x)$ represents the sigmoid function, α is a hyperparameter 'slope' in the sigmoid, β represents the half size of the vector of 3D dimensions w, l, and h of the bounding box. Coordinate values γ in the source point cloud are warped to the target point cloud via motion box parameters Φ. For each dynamic box, each point's relative position to the box's centre is calculated. Higher motion probability P_M is assigned to the points closer to the centre. A fixed hyperparameter α, controlling motion probability, may not effectively respond to diverse and complex autonomous driving scenarios. Therefore, we adopt an adaptive computation of α based on the variance of the point nearest-neighbour consistency loss from the previous generation. The variance in the nearest-neighbour consistency loss for different points in the background implies the distribution of dynamic objects in the scene. With fewer moving objects indicated by a lower variance, α should be adaptively reduced, tending to produce lower motion probability P_M for points.

In addition to L_{BG} and L_{FG}, we introduce box dimension regularization, heading term, and angle term to constrain the dimensions, heading, and rotation angles of the bounding boxes within a reasonable range [5]. We also introduce a mass term to ensure that there are as many points as possible within the box, making the estimated motion parameters of the box more robust [5].

3.3. Data Augmentation for 3D Flow Auto-labelling

Existing data augmentation practices [49] often add consistent random rotations and noise offsets to the input points, which indeed yields certain benefits. However, in

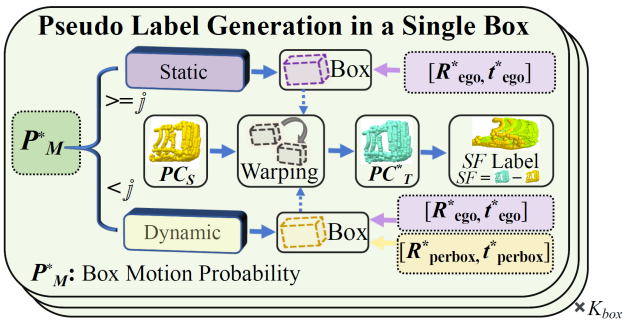

Figure 4. The proposed pseudo label generation module. With the augmented motion probability P_M^*, bounding boxes are categorized into dynamic and static types. Using global and local motion parameters, the PC_S is warped to the target point cloud PC_T^*. Finally, pseudo 3D scene flow labels SF are derived from the correspondence between PC_T^* and PC_S. K_{box} represents the number of boxes.

autonomous driving scenarios, there are frequently various complex motion patterns for multiple objects. To make models learn complex scene motion rules, we propose a novel data augmentation method for scene flow labelling in both global and object-level motions. Our method simulates a broad spectrum of 3D scene flow data variations, originating from ego-motion and dynamic object movement, thereby providing a promising solution to the challenge of securing abundant 3D scene flow labels.

As in Fig. 3, random noise is applied to either global or local motion parameters respectively. We generate a random rotation angle α and a random unit vector $\boldsymbol{u}$ for the rotation direction using random noise. They are used to create the Lie algebra $\boldsymbol{\xi}$. Subsequently, the Lie algebra $\boldsymbol{\xi}$ is converted into a rotation matrix $\mathbf{M}$ using the Rodrigues' rotation formula and applied to the original rotation matrix $\mathbf{R}$ to obtain a new rotation matrix $\mathbf{R}^*$, as follows:

$$\mathbf{M} = \mathbf{I} + \sin(|\boldsymbol{\xi}|)\frac{\boldsymbol{\xi}}{|\boldsymbol{\xi}|}_{\times} + (1 - \cos(|\boldsymbol{\xi}|))\left(\frac{\boldsymbol{\xi}}{|\boldsymbol{\xi}|}_{\times}\right)^2, \quad (5)$$

$$\boldsymbol{\xi} = \alpha \boldsymbol{u}, \mathbf{R}^* = \mathbf{R}\mathbf{M}. \quad (6)$$

The Lie algebra element $\boldsymbol{\xi}$, the product of scalar α and unit vector $\boldsymbol{u}$, signifies rotation magnitude and direction, with α and $\boldsymbol{u}$ representing rotation angle and axis, respectively. $\mathbf{I}$ is identity matrix, and $\boldsymbol{\xi}\times$ is the antisymmetric matrix of $\boldsymbol{\xi}$. Lie algebra intuitively and conveniently represents minor $SO(3)$ group variations. Rodrigues' rotation formula, mapping from the Lie algebra to the Lie group, facilitates the transformation of angle-based noise into a form directly applicable to the rotation matrix. This transformation brings mathematical convenience, making the update of the rotation matrix concise and efficient.

Importantly, our data augmentation targets dynamically moving objects, because persistently adding varied motion noise to bounding boxes perceived as static objects may disrupt original data distribution. Moreover, the translation and motion probability are also augmented. As depicted in Fig. 3, we generate noise within an appropriate range and directly add it to the translation matrix or motion probability, resulting in augmented translation and motion probability.

3.4. Pseudo Label Generation for 3D Scene Flow

The motion parameters are fed into the pseudo label generation module to obtain point-wise 3D scene flow labels. The specific process of the label generation module is shown in Fig. 4. We determine the motion state of the 3D bounding box through the motion probability P_M:

$$PC_T^* = \begin{cases} PC_S \times R_{ego}^* + t_{ego}^* & \text{if } P_M^* < \mathbb{J}, \\ PC_S^{ego} \times R_{perbox}^* + t_{perbox}^* & \text{if } P_M^* \geq \mathbb{J}. \end{cases} \quad (7)$$

PC_S^{ego} is the points in the dynamic box from the source point cloud, transformed through global rotation and translation. When P_M is less than threshold $\mathbb{J}$, the current bounding box is deemed static. Conversely, if P_M exceeds a predefined threshold $\mathbb{J}$, the current bounding box is considered dynamic. For static boxes, based on the existing global motion, we apply a uniform noise to all static boxes to simulate various ego-motion patterns. By adding minute noise to the motion probability P_M for each box, we can construct various motion states and show a greater variety of scene motions. Before transforming the dynamic boxes, a prior global transformation of all points is required. For dynamic bounding boxes, we add various noises to their existing motion, generating new rotations and translations, thereby creating various motion patterns. We warp the source point cloud within each box to the target frame using the box's motion parameters, obtaining the pseudo target point cloud PC_T^*.

The generated pseudo target point cloud PC_T^* and the real source frame point cloud PC_S have a perfect correspondence. Therefore, the 3D scene flow labels can be easily obtained by directly subtracting PC_S from PC_T^*:

$$SF = PC_T^* - PC_S. \quad (8)$$

The generated scene flow labels capture various motion patterns from real autonomous driving scenes. They help the model understand and adjust to complex driving conditions. This improves the model's ability to generalize in unfamiliar real-world scenarios.

4. Experiments

4.1. Datasets

Test Datasets: Graph prior [35] introduces two autonomous driving datasets, Argoverse scene flow [3] and nuScenes scene flow [2] datasets. Scene flow labels in the

Table 1. Comparison of our method with the best-performing methods on multiple datasets [2, 3, 10] and metrics. 'None', 'Weak', 'Self', and 'Full' represent non-learning, weakly supervised, self-supervised, and supervised methods, respectively. "↑" means higher is better, and "↓" means lower is better. Our method uses GMSF [51] as a baseline and combines it with our proposed pseudo-auto-labelling framework, 3DSFlabelling. Despite the use of a supervised learning structure, no ground truth is utilized in training.

Method	Sup.	LiDAR KITTI Scene Flow [10]				Argoverse Scene Flow [3]				nuScenes Scene Flow [2]			
		EPE3D↓	Acc3DS↑	Acc3DR↑	Outliers↓	EPE3D↓	Acc3DS↑	Acc3DR↑	Outliers↓	EPE3D↓	Acc3DS↑	Acc3DR↑	Outliers↓
Graph prior [35]	None	–	–	–	–	0.2570	0.2524	0.4760	–	0.2890	0.2012	0.4354	–
RSF [5]	None	0.0850	0.8830	0.9290	0.2390	–	–	–	–	0.1070	0.7170	0.8620	0.3210
NSFP [23]	None	0.1420	0.6880	0.8260	0.3850	0.1590	0.3843	0.6308	–	0.1751	0.3518	0.6345	0.5270
R3DSF [12]	Weak	0.0940	0.7840	0.8850	0.3140	0.4160	0.3452	0.4310	0.5580	–	–	–	–
FlowNet3D [28]	Full	0.7220	0.0300	0.1220	0.9650	0.4550	0.0134	0.0612	0.7360	0.5050	0.2120	0.1081	0.6200
PointPWC [49]	Full	0.3900	0.3870	0.5500	0.6530	0.4288	0.0462	0.2164	0.9199	0.7883	0.0287	0.1333	0.9410
DCA-SRSFE [18]	Full	0.5900	0.1505	0.3331	0.8485	0.7957	0.0712	0.1468	0.9799	0.7042	0.0538	0.1183	0.9766
FLOT [36]	Full	0.6532	0.1554	0.3130	0.8371	0.2491	0.0946	0.3126	0.8657	0.4858	0.0821	0.2669	0.8547
MSBRN [4]	Full	0.0139	0.9752	0.9847	0.1433	0.8691	0.2432	0.2854	0.7597	0.6137	0.2354	0.2924	0.7638
GMSF [51]	Full	0.1900	0.2962	0.5502	0.6171	7.2776	0.0036	0.0144	0.9930	9.4231	0.0034	0.0086	0.9943
Mittal et al. [32]	Self	0.9773	0.0096	0.0524	0.9936	0.6520	0.0319	0.1159	0.9621	0.8422	0.0289	0.1041	0.9615
Jiang et al. [17]	Self	0.4908	0.2052	0.4238	0.7286	0.2517	0.1236	0.3666	0.8114	0.4709	0.1034	0.3175	0.8191
Ours	Self	**0.0078**	**0.9924**	**0.9947**	**0.1328**	**0.0093**	**0.9780**	**0.9880**	**0.1302**	**0.0185**	**0.9534**	**0.9713**	**0.1670**

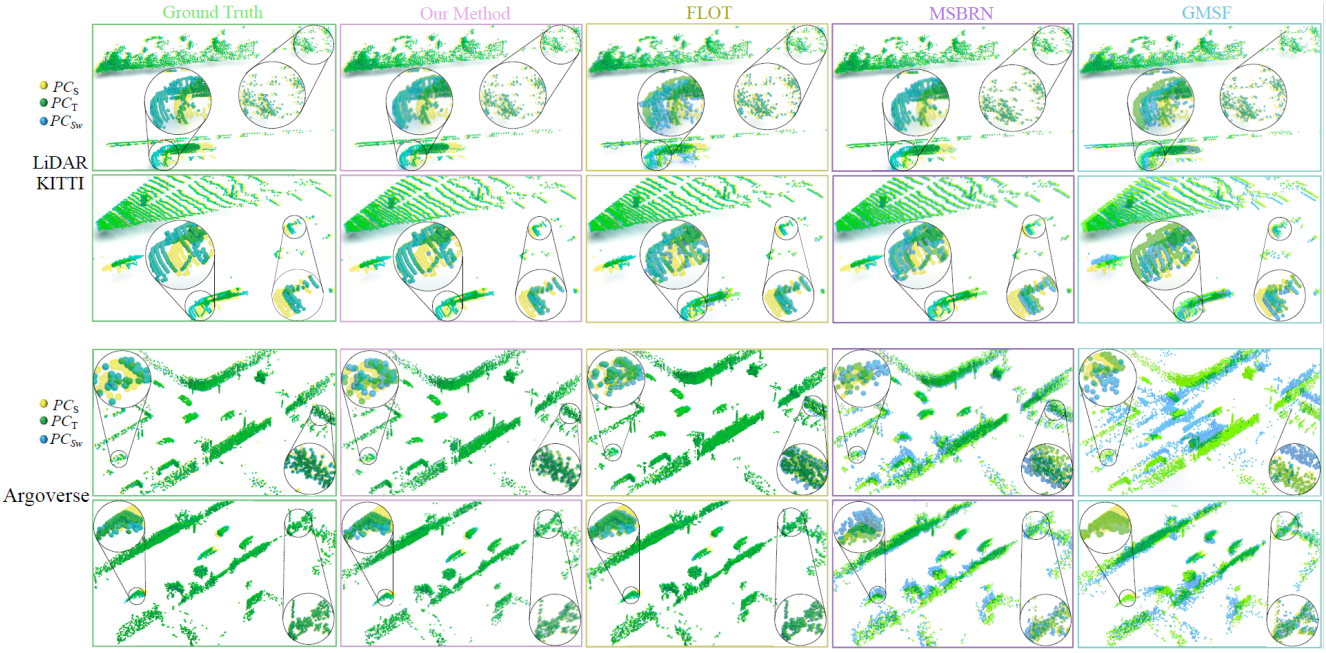

Figure 5. Registration visualization results of our method (GMSF [51]+3DSFlabelling) and baselines on the LiDAR KITTI and Argoverse datasets [3, 10]. The estimated target point cloud PC_{sw} is derived from warping the source point cloud PC_S to the target point cloud via 3D scene flow. The larger the overlap between PC_{sw} (blue) and the target point cloud PC_T (green), the higher the predicted accuracy of the scene flow. Local areas are zoomed in for better visibility. Our 3D scene flow estimation notably improves performance.

datasets are derived from LiDAR point clouds, object trajectories, map data, and vehicle pose. The datasets contain 212 and 310 test samples, respectively. R3DSF [12] introduces the lidarKITTI [10], which shares 142 scenes with stereoKITTI, collected via Velodyne's 64-beam LiDAR. Unlike FT3D [29] and stereoKITTI [30, 31], the point clouds from lidarKITTI are sparsely distributed. Note that LiDAR scene flow ground truths contain errors. We mitigate this by fusing the ground truth with the first point cloud to create a corrected second frame for network input, thus avoiding evaluation errors.

Training Datasets used in previous methods: FT3D [29] and stereoKITTI [30, 31] are the frequently used datasets for training previous 3D scene flow models [4, 28, 36, 49, 51]. FT3D consists of 19,640 training pairs, while stereoKITTI [30, 31] contains 142 dense point clouds, with the first 100 frames used for model fine-tuning in some works [23, 32]. Some works [23, 28, 32, 35, 49] train their models on 2,691 pairs of Argoverse [3] data and 1,513 pairs of nuScenes [2] data, with 3D scene flow annotations fol-

Table 2. The comparative results between our method and baseline. "↑" signifies accuracy enhancement. In real-world LiDAR scenarios, our method markedly improves the 3D flow estimation accuracy across three datasets [2, 3, 30] on the three baselines. This demonstrates that the proposed pseudo-auto-labelling framework can substantially boost the accuracy of existing methods, even without the need for ground truth.

Dataset	Method	EPE3D↓	Acc3DS↑	Acc3DR↑
LiDAR KITTI	FLOT [36]	0.6532	0.1554	0.3130
	FLOT+3DSFlabelling	**0.0189** ↑ **97.1%**	**0.9666**	**0.9792**
	MSBRN [4]	0.0139	0.9752	0.9847
	MSBRN+3DSFlabelling	**0.0123** ↑ **11.5%**	**0.9797**	**0.9868**
	GMSF [51]	0.1900	0.2962	0.5502
	GMSF+3DSFlabelling	**0.0078** ↑ **95.8%**	**0.9924**	**0.9947**
Argoverse	FLOT [36]	0.2491	0.0946	0.3126
	FLOT+3DSFlabelling	**0.0107** ↑ **95.7%**	**0.9711**	**0.9862**
	MSBRN [4]	0.8691	0.2432	0.2854
	MSBRN+3DSFlabelling	**0.0150** ↑ **98.3%**	**0.9482**	**0.9601**
	GMSF [51]	7.2776	0.0036	0.0144
	GMSF+3DSFlabelling	**0.0093** ↑ **99.9%**	**0.9780**	**0.9880**
nuScenes	FLOT [36]	0.4858	0.0821	0.2669
	FLOT+3DSFlabelling	**0.0554** ↑ **88.6%**	**0.7601**	**0.8909**
	MSBRN [4]	0.6137	0.2354	0.2924
	MSBRN+3DSFlabelling	**0.0235** ↑ **96.2%**	**0.9413**	**0.9604**
	GMSF [51]	9.4231	0.0034	0.0086
	GMSF+3DSFlabelling	**0.0185** ↑ **99.8%**	**0.9534**	**0.9713**

lowing the settings of the Graph prior [35]. The R3DSF [12] training set utilizes FT3D and semanticKITTI datasets [1], relying on ego-motion labels and semantic segmentation labels from semanticKITTI.

Training Datasets used in our methods: Because we do not need any labels for training data, we use raw LiDAR point clouds sampled from raw data. For testing on the lidarKITTI [31], we use LiDAR point clouds from sequences 00 to 09 of the KITTI Odometry dataset [11] for auto-labelling and training. For testing on the nuScenes scene flow dataset [2], we randomly sample 50,000 pairs of LiDAR point clouds from the 350,000 LiDAR point clouds in the nuScenes sweeps dataset [2]. For testing on the Argoverse scene flow Dataset [3], we use the LiDAR point clouds from sequences 01 to 05 of the Argoverse 2 Sensor Dataset [3] for auto-labelling and training. In the selection of training data, we exclude the test scenes.

4.2. Implementation Details

The effectiveness of the proposed auto-labelling framework is demonstrated using three prominent deep learning models: FLOT [36], MSBRN [4], and GMSF [51]. These models use optimal transport, coarse-to-fine strategies, and transformer architectures respectively. Hyperparameters consistent with the original networks are employed during the training process. The input point clouds, from which ground points have been filtered, are randomly sampled to incorporate 8192 points. The LiDAR point cloud data from KITTI [10] is confined to the front view perspective, maintaining consistency with previous studies [12].

Table 3. Model comparison on the Argoverse dataset [3]. 'M' represents millions of parameters, and time is in milliseconds.

Method	Sup.	EPE3D↓	Acc3DS↑	Acc3DR↑	Time↓	Params.↓
PointPWC [49]	Full	0.4288	0.0462	0.2164	147 ms	7.7 M
PV-RAFT [47]	Full	10.745	0.0200	0.0100	169 ms	–
R3DSF [12]	Weak	0.4160	0.3452	0.4310	113 ms	8.0 M
FlowStep3D [19]	Self	0.8450	0.0100	0.0800	729 ms	–
NSFP [23]	None	0.1590	0.3843	0.6308	2864 ms	–
Fast-NSF [24]	None	0.1180	0.6993	0.8355	124 ms	–
MBNSF [41]	None	0.0510	0.7936	0.9237	5000+ ms	–
MSBRN+3DSFlabelling	Self	0.0150	0.9482	0.9601	341 ms	3.5 M
GMSF+3DSFlabelling	Self	**0.0093**	**0.9780**	**0.9880**	251 ms	6.0 M
FLOT+3DSFlabelling	Self	0.0107	0.9711	0.9862	**78 ms**	**0.1 M**

Furthermore, we utilize four scene flow evaluation metrics [28, 36, 49, 51]: Average Endpoint Error (EPE3D), ACC3DS, ACC3DR, and Outliers.

4.3. Quantitative Results

The experimental results are presented in Table 1. We list the best-performing optimized [5, 12, 23, 35], self-supervised [17, 32], and supervised [18, 28, 49] models in the table. Our method achieves excellent performance on all datasets [2, 3, 10] and metrics. Particularly, compared to the baselines [51], there is an order of magnitude reduction in EPE3D on most datasets. The proposed auto-labelling method generates effective scene flow labels, perfectly simulating the rigid motion of various objects in the real world. The designed global-local data augmentation further expands the 3D scene flow labels. As a result, our method significantly outperforms other methods. We have also applied this plug-and-play auto-labelling framework for 3D scene flow (3DSFlabelling) to three existing models, as demonstrated in Table 2. The proposed method significantly enhances the accuracy of 3D scene flow estimation in these models [4, 36, 51].

Moreover, many existing works utilize a large number of model parameters [12, 47, 49] or adopt optimization methods [23, 24, 41] during testing for a more accurate estimation of 3D scene flow. These methods are highly time-consuming, and cannot ensure accuracy when reducing model parameters. Our proposed 3DSFlabelling effectively addresses this challenge. In Table 3, by using the small-parameter model FLOT (iter=1) [36] combined with our auto-labelling framework, we surpass all current supervised, unsupervised, weakly supervised, and optimized methods. This strongly validates the effectiveness of generating real-world labels in solving the challenges.

4.4. Visualization

Fig. 5 visualizes the precision of our method and others on two datasets [3, 31]. FLOT [36], with its mathematically optimal transport approach to matching point clouds, exhibits superior generalization. MSBRN [4], leveraging a multi-scale bidirectional recurrent network, robustly esti-

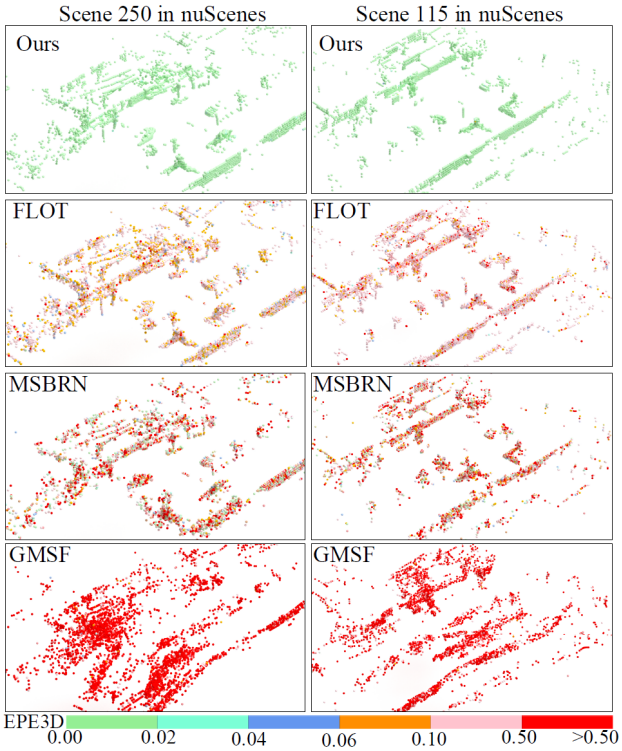

Figure 6. Error visualizing of our method (GMSF+3DSFlabelling) and baselines on the nuScenes dataset [2]. Using 3D EndPoint Error (EPE3D) as the metric, we categorize the error into six levels. Combining GMSF [51] with our proposed 3DSFlabelling, we manage to keep the EPE3D for most points within 0.02 meters, clearly outperforming other methods largely.

Table 4. Generalization comparison experiment. "A", "N", and "K" represent the Argoverse [3], nuScenes [2], and KITTI [10] datasets. '⇝' representing a model trained on the dataset on the left and directly evaluated on another new dataset on the right.

Method	Sup.	**A⇝N**		**N⇝A**		**A⇝K**		**N⇝K**	
		EPE3D	Acc3DS	EPE3D	Acc3DS	EPE3D	Acc3DS	EPE3D	Acc3DS
PointPWC [49]	Self	0.5911	0.0844	0.7043	0.0281	0.8632	0.0119	0.9307	0.0027
RigidFlow [12]	Self	0.1135	0.3445	0.3991	0.0152	0.3645	0.2118	0.5042	0.0141
MSBRN [4]	Full	0.5309	0.0055	0.3761	0.0098	0.6036	0.0056	0.4926	0.0081
GMSF [51]	Full	0.0334	0.9037	0.3078	0.1278	0.0442	0.8764	0.0574	0.8135
Ours	Self	**0.0115**	**0.9693**	**0.0264**	**0.9192**	**0.0414**	**0.9020**	**0.0208**	**0.9595**

mates 3D scene flow on KITTI. GMSF [51] utilizes a transformer architecture for powerful fitting learning, but it lacks cross-domain generalization. The proposed method consistently shows better alignment between predicted and target point clouds across all scenes. Additionally, a visualization of the scene flow error on the nuScenes dataset is presented in Fig. 6. In two randomly selected test scenes, our method keeps the scene flow EPE3D mostly within $0.02m$, clearly outperforming other baselines. More visual comparisons will be presented in the supplementary material.

Table 4 provides quantitative results, demonstrating the generalization of our 3DSFlabelling combined with the existing method (GMSF [51]) on new datasets. For instance, we train a model on the Argoverse dataset and directly evaluate it on the nuScenes dataset. These two datasets belong to different domains, posing a domain generalization problem. The results in Table 4 indicate that our framework performs exceptionally well on the new dataset, consistently achieving an EPE3D of less than $5cm$, and even reaching an average endpoint error of less than $2cm$.

Table 5. Ablation study of 3D scene flow data augmentation. "No Aug" and "Trad. Aug" represents no data augmentation and traditional data augmentation [49], respectively. Our data augmentation method has a very positive impact on the model.

Model	Data Augmentation Methods			KITTI		Argoverse		nuScenes	
	No Aug	Trad. Aug	Our Aug	EPE3D	ACC3DS	EPE3D	ACC3DS	EPE3D	ACC3DS
Ours (FLOT)	✓	–	–	0.0601	0.7291	0.0492	0.8015	0.7364	0.6642
	–	✓	–	0.0540	0.7622	0.0430	0.8679	0.0610	0.7417
	–	–	✓	**0.0189**	**0.9666**	**0.0107**	**0.9711**	**0.0554**	**0.7601**
Ours (MSBRN)	✓	–	–	0.0131	0.9781	0.0180	0.9411	0.0797	0.8510
	–	✓	–	0.0129	0.9790	0.0177	0.9427	0.0793	0.8547
	–	–	✓	**0.0123**	**0.9797**	**0.0150**	**0.9482**	**0.0235**	**0.9413**
Ours (GMSF)	✓	–	–	0.0103	0.9901	0.0139	0.9637	0.0213	0.9468
	–	✓	–	0.0081	0.9918	0.0137	0.9663	0.0212	0.9473
	–	–	✓	**0.0078**	**0.9924**	**0.0093**	**0.9780**	**0.0185**	**0.9534**

4.5. Ablation Study

This section explores the advantages of global-local data augmentation. In Table 5, we compare existing 3D scene flow data augmentation [49] with our proposed global-local data augmentation method. Our augmentation strategy shows significant enhancement in all evaluation metrics. This is attributed to the effective simulation of various motion patterns in autonomous driving by global-local data augmentation. The introduction of various motion transformations excellently utilizes the limited training data to extend a variety of 3D scene flow styles. More ablation studies are referring to the supplement material.

5. Conclusion

We package 3D point clouds into boxes with different motion attributes. By optimizing the motion parameters for each box and warping the source point cloud into the target point cloud, we create pseudo 3D scene flow labels. We also design a global-local data augmentation method, introducing various scene motion patterns and significantly increasing the diversity and quantity of 3D scene flow labels. Tests on multiple real-world datasets show that our 3D scene flow auto-labelling significantly enhances the performance of existing models. Importantly, this approach eliminates the need for 3D scene flow estimation models to depend on manually annotated 3D scene flow labels.

6. Acknowledgements

This work was supported by PhiGent Robotics.

References

[1] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9297–9307, 2019. 7

[2] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 2, 5, 6, 7, 8

[3] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3d tracking and forecasting with rich maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8748–8757, 2019. 1, 2, 5, 6, 7, 8

[4] Wencan Cheng and Jong Hwan Ko. Multi-scale bidirectional recurrent network with hybrid correlation for point cloud based scene flow estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10041–10050, 2023. 1, 2, 6, 7, 8

[5] David Deng and Avideh Zakhor. Rsf: Optimizing rigid scene flow from 3d point clouds without labels. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1277–1286, 2023. 2, 3, 4, 6, 7

[6] Fangqiang Ding, Zhijun Pan, Yimin Deng, Jianning Deng, and Chris Xiaoxuan Lu. Self-supervised scene flow estimation with 4-d automotive radar. *IEEE Robotics and Automation Letters*, 7(3):8233–8240, 2022. 2

[7] Guanting Dong, Yueyi Zhang, Hanlin Li, Xiaoyan Sun, and Zhiwei Xiong. Exploiting rigidity constraints for lidar scene flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12776–12785, 2022. 3

[8] Emeç Erçelik, Ekim Yurtsever, Mingyu Liu, Zhijie Yang, Hanzhen Zhang, Pınar Topçam, Maximilian Listl, Yılmaz Kaan Caylı, and Alois Knoll. 3d object detection with a self-supervised lidar scene flow backbone. In *European Conference on Computer Vision*, pages 247–265. Springer, 2022. 1

[9] Jingyun Fu, Zhiyu Xiang, Chengyu Qiao, and Tingming Bai. Pt-flownet: Scene flow estimation on point clouds with point transformer. *IEEE Robotics and Automation Letters*, 8(5): 2566–2573, 2023. 1, 2

[10] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. 6, 7, 8

[11] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 7

[12] Zan Gojcic, Or Litany, Andreas Wieser, Leonidas J Guibas, and Tolga Birdal. Weakly supervised learning of rigid 3d scene flow. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5692–5703, 2021. 3, 6, 7, 8

[13] Xiuye Gu, Yijie Wang, Chongruo Wu, Yong Jae Lee, and Panqu Wang. Hplflownet: Hierarchical permutohedral lattice flownet for scene flow estimation on large-scale point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3254–3263, 2019. 2

[14] Shengyu Huang, Zan Gojcic, Jiahui Huang, Andreas Wieser, and Konrad Schindler. Dynamic 3d scene analysis by point cloud accumulation. In *European Conference on Computer Vision*, pages 674–690. Springer, 2022. 1

[15] Hafsa Iqbal, Abdulla Al-Kaff, Pablo Marin, Lucio Marcenaro, David Martin Gomez, and Carlo Regazzoni. Detection of abnormal motion by estimating scene flows of point clouds for autonomous driving. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 2788–2793. IEEE, 2021. 1

[16] Chaokang Jiang, Guangming Wang, Jinxing Wu, Yanzi Miao, and Hesheng Wang. Ffpa-net: Efficient feature fusion with projection awareness for 3d object detection. *arXiv preprint arXiv:2209.07419*, 2022. 1

[17] Chaokang Jiang, Guangming Wang, Yanzi Miao, and Hesheng Wang. 3-d scene flow estimation on pseudo-lidar: Bridging the gap on estimating point motion. *IEEE Transactions on Industrial Informatics*, 19(6):7346–7354, 2023. 6, 7

[18] Zhao Jin, Yinjie Lei, Naveed Akhtar, Haifeng Li, and Munawar Hayat. Deformation and correspondence aware unsupervised synthetic-to-real scene flow estimation for point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7233–7243, 2022. 1, 6, 7

[19] Yair Kittenplon, Yonina C Eldar, and Dan Raviv. Flowstep3d: Model unrolling for self-supervised scene flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4114–4123, 2021. 7

[20] Itai Lang, Dror Aiger, Forrester Cole, Shai Avidan, and Michael Rubinstein. Scoop: Self-supervised correspondence and optimization-based scene flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5281–5290, 2023. 3

[21] Bing Li, Cheng Zheng, Silvio Giancola, and Bernard Ghanem. Sctn: Sparse convolution-transformer network for scene flow estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1254–1262, 2022. 2

[22] Ruibo Li, Chi Zhang, Guosheng Lin, Zhe Wang, and Chunhua Shen. 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*, pages 16959–16968, 2022. 2, 3

[23] Xueqian Li, Jhony Kaesemodel Pontes, and Simon Lucey. Neural scene flow prior. *Advances in Neural Information Processing Systems*, 34:7838–7851, 2021. 3, 6, 7

[24] Xueqian Li, Jianqiao Zheng, Francesco Ferroni, Jhony Kaesemodel Pontes, and Simon Lucey. Fast neural scene flow. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9878–9890, 2023. 1, 3, 7

[25] Jiuming Liu, Guangming Wang, Chaokang Jiang, Zhe Liu, and Hesheng Wang. Translo: A window-based masked point transformer framework for large-scale lidar odometry. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1683–1691, 2023. 3

[26] Jiuming Liu, Guangming Wang, Zhe Liu, Chaokang Jiang, Marc Pollefeys, and Hesheng Wang. Regformer: an efficient projection-aware transformer network for large-scale point cloud registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8451–8460, 2023. 3

[27] Jiuming Liu, Guangming Wang, Weicai Ye, Chaokang Jiang, Jinru Han, Zhe Liu, Guofeng Zhang, Dalong Du, and Hesheng Wang. Difflow3d: Toward robust uncertainty-aware scene flow estimation with diffusion model. *arXiv preprint arXiv:2311.17456*, 2023. 1

[28] Xingyu Liu, Charles R Qi, and Leonidas J Guibas. Flownet3d: Learning scene flow in 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 529–537, 2019. 2, 6, 7

[29] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. 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*, pages 4040–4048, 2016. 1, 2, 6

[30] Moritz Menze, Christian Heipke, and Andreas Geiger. Joint 3d estimation of vehicles and scene flow. *ISPRS annals of the photogrammetry, remote sensing and spatial information sciences*, 2:427–434, 2015. 1, 2, 6, 7

[31] Moritz Menze, Christian Heipke, and Andreas Geiger. Object scene flow. *ISPRS Journal of Photogrammetry and Remote Sensing*, 140:60–76, 2018. 1, 2, 6, 7

[32] Himangi Mittal, Brian Okorn, and David Held. Just go with the flow: Self-supervised scene flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11177–11185, 2020. 2, 3, 6, 7

[33] Mahyar Najibi, Jingwei Ji, Yin Zhou, Charles R Qi, Xinchen Yan, Scott Ettinger, and Dragomir Anguelov. Motion inspired unsupervised perception and prediction in autonomous driving. In *European Conference on Computer Vision*, pages 424–443. Springer, 2022. 1

[34] Chensheng Peng, Guangming Wang, Xian Wan Lo, Xinrui Wu, Chenfeng Xu, Masayoshi Tomizuka, Wei Zhan, and Hesheng Wang. Delflow: Dense efficient learning of scene flow for large-scale point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16901–16910, 2023. 2

[35] Jhony Kaesemodel Pontes, James Hays, and Simon Lucey. Scene flow from point clouds with or without learning. In *2020 International Conference on 3D Vision (3DV)*, pages 261–270, 2020. 3, 5, 6, 7

[36] Gilles Puy, Alexandre Boulch, and Renaud Marlet. Flot: Scene flow on point clouds guided by optimal transport. In *ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII 16*, pages 527–544, 2020. 1, 2, 6, 7

[37] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 1

[38] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 1

[39] Yaqi Shen, Le Hui, Jin Xie, and Jian Yang. Self-supervised 3d scene flow estimation guided by superpoints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5271–5280, 2023. 1, 2

[40] Ivan Tishchenko, Sandro Lombardi, Martin R Oswald, and Marc Pollefeys. Self-supervised learning of non-rigid residual flow and ego-motion. In *2020 international conference on 3D vision (3DV)*, pages 150–159. IEEE, 2020. 3

[41] Kavisha Vidanapathirana, Shin-Fang Chng, Xueqian Li, and Simon Lucey. Multi-body neural scene flow. *arXiv preprint arXiv:2310.10301*, 2023. 7

[42] Guangming Wang, Xinrui Wu, Zhe Liu, and Hesheng Wang. Hierarchical attention learning of scene flow in 3d point clouds. *IEEE Transactions on Image Processing*, 30:5168–5181, 2021. 2

[43] Guangming Wang, Yunzhe Hu, Zhe Liu, Yiyang Zhou, Masayoshi Tomizuka, Wei Zhan, and Hesheng Wang. What matters for 3d scene flow network. In *European Conference on Computer Vision*, pages 38–55. Springer, 2022. 2

[44] Guangming Wang, Chaokang Jiang, Zehang Shen, Yanzi Miao, and Hesheng Wang. Sfgan: Unsupervised generative adversarial learning of 3d scene flow from the 3d scene self. *Advanced Intelligent Systems*, 4(4):2100197, 2022. 2, 3

[45] Yun Wang, Cheng Chi, and Xin Yang. Exploiting implicit rigidity constraints via weight-sharing aggregation for scene flow estimation from point clouds. *arXiv preprint arXiv:2303.02454*, 2023. 2

[46] Ziyi Wang, Yi Wei, Yongming Rao, Jie Zhou, and Jiwen Lu. 3d point-voxel correlation fields for scene flow estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 2

[47] Yi Wei, Ziyi Wang, Yongming Rao, Jiwen Lu, and Jie Zhou. Pv-raft: Point-voxel correlation fields for scene flow estimation of point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6954–6963, 2021. 1, 2, 7

[48] Pengxiang Wu, Siheng Chen, and Dimitris N Metaxas. Motionnet: Joint perception and motion prediction for autonomous driving based on bird's eye view maps. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11385–11395, 2020. 1

[49] Wenxuan Wu, Zhi Yuan Wang, Zhuwen Li, Wei Liu, and Li Fuxin. Pointpwc-net: Cost volume on point clouds for (self-) supervised scene flow estimation. In *European Conference on Computer Vision*, pages 88–107, 2020. 2, 3, 4, 6, 7, 8

[50] Yi Zhang, Yuwen Ye, Zhiyu Xiang, and Jiaqi Gu. Sdp-net: Scene flow based real-time object detection and prediction from sequential 3d point clouds. In *Proceedings of the Asian Conference on Computer Vision*, 2020. 1

[51] Yushan Zhang, Johan Edstedt, Bastian Wandt, Per-Erik Forssén, Maria Magnusson, and Michael Felsberg. Gmsf: Global matching scene flow. *arXiv preprint arXiv:2305.17432*, 2023. 1, 2, 6, 7, 8

[52] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3d: A modern library for 3d data processing. *arXiv preprint arXiv:1801.09847*, 2018. 3