SFGAN: Unsupervised Generative Adversarial Learning of 3D Scene Flow from the 3D Scene Self

Guangming Wang, Xinrui Wu, Zhe Liu, and Hesheng Wang

***Abstract*—3D scene flow presents the 3D motion of each point in the real physical world. Although the RGBD camera or LiDAR capture discrete 3D points in space, the objects and motions usually are continuous in the macro world. The objects in the physical world keep themselves consistent as they flow from the current frame to the next frame., the constraint for the learning of 3D scene flow can be built just from the raw consecutive 3D point clouds. Based on the insight, we utilize the Generative Adversarial Networks (GAN) to self-learn 3D scene flow with no need for ground truth. The fake point cloud of the second frame is synthesized from the predicted scene flow and the point cloud of the first frame. The adversarial training of the discriminator and generator through discriminating the real point cloud of the second frame and the generated point cloud of the second frame, which makes the generated point cloud more and more similar to the real one and thus makes the scene flow estimation more accurate. We try various possible discriminator network structures and do the ablation studies to analyze the designed discriminator structure. Finally, the experiments demonstrated the best structure for the learning of 3D scene flow. The experiments also show that our method realizes promising results although the 3D scene flow estimation network just learns from the point clouds of two consecutive frames without ground truth. Just like a human observing a real-world scene, Our approach is capable of determining the consistency of the scene at different momentsin spite of the exact flow value of each point is unknown in advance.**

***Index Terms*—scene flow estimation, 3D point clouds, Generative Adversarial Network (GAN), soft correspondence, unsupervised learning**

I. Introduction

JUST like estimating 2D optical flow from a pair of images, estimating 3D scene flow from two frames of 3D point clouds is an essential task in computer vision. 3D scene flow can be applied to object detection and tracking [1]–[3], LIDAR odometry [4], action recognition [5], etc. Recently, some works [6]–[10] has been done to realize supervised estimation of 3D scene flow from point clouds of two consecutive frames. However, just as it is difficult to obtain the true value of optical flow [11], [12], the true value of 3D scene flow is also difficult to obtain. Therefore, it is very important to perform unsupervised learning of 3D scene flow. The dynamic environment changes over time, which will cause some edge points of the 3D point cloud scene to be lost. For example, some points at the edges of the point cloud in the first frame are used to characterize a person, and as the scene is shifted, the point describing the person may disappear in the second frame. Existing unsupervised learning methods for 3D scene flow always have some assumptions, which do not completely conform to the real situation. For example, some works [13], [14] introduce Chamfer loss to self-supervised learning of scene flow. Chamfer loss assumes that the coordinates of each sampled point in the predicted point cloud of the second frame and the coordinates of each sampled point in the real point cloud of the second frame are exactly the same in geometric space. Chamfer loss aims to minimize the distance between the nearest points in both the predicted point cloud and the real point cloud. Due to the discrete sampling of adjacent frames, the points that characterize the same object do not correspond point by point. Chamfer loss violates the discreteness of adjacent frames in the scene and the sampling fact. Mittal et al. [15] proposed Cycle Consistency Loss that predicts the reverse flow in order to transform the predicted point cloud of the second frame into the position of the first frame. Which minimizes the distance between the nearest point in the predicted point cloud of the first frame and the true point cloud of the first frame. In order to make the point cloud structure of the estimated reverse flow obvious, recent work [15] has modified the starting point of the reverse flow. This artificial manipulation of the original data violates the true data distribution. The curvature loss proposed by PointPWC-Net [13] assumes that the predicted point clouds have the same Laplace coordinates at the same locations in space as the real point clouds, which is also inconsistent with the fact of discrete sampling of adjacent frame point clouds. How to perform unsupervised learning from raw data without assumptions is a challenge.

In this paper, we use the scene flow estimation network as a generator and design a robust discriminator to discriminate the generated point clouds and the real point clouds. The real labels are not utilized in the optimization of the scene flow generator. Just like human perception, the discriminator discriminates the consistency between the 3D scene represented by the generated point cloud and the 3D scene represented by the real point cloud to optimize the accuracy of the scene flow estimation.

Our main contributions in this work are shown as follows:

- A novel self-supervised learning framework for 3D scene flow is proposed, in which generative adversarial ideas are introduced to learn 3D scene flow. The adversarial learning between the scene flow generator and the point

*This work was supported in part by the Natural Science Foundation of China under Grant U1613218 and 61722309. The first two authors contributed equally. Corresponding Author: Hesheng Wang.

G. Wang, X. Wu, and H. Wang are with Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China and Key Laboratory of System Control and Information Processing, Ministry of Education of China.

Z. Liu is with the Department of Computer Science and Technology, University of Cambridge.

cloud discriminator makes the point clouds generated by the generator closer and closer to the real point clouds, thus making the scene flow estimation more accurate.
- Four different types of point cloud discriminators are designed, which can be used to discriminate whether the point clouds are from training data or generated data. The best discriminator structure is finally verification by ablation experiments.
- The experimental results in KITTI dataset [16] demonstrate that the introduction of adversarial learning ideas in scene flow estimation is effective to improve the performance of scene flow estimation.

Our paper is divided into five sections in total. Section II shows the related work. Section III introduces our approach, describing the overall framework, the detailed structure of the scene flow generator and the point cloud discriminator and their adversarial learning process, respectively. The experimental part including training details, dataset description, evaluation metrics, result analysis, and ablation experiments are presented in Section IV. Section V shows the conclusion.

II. Related Work

Scene flow is a 3D motion field formed by the movement of scenes in 3D space, which has an important role in the autonomous driving field. Motion information is essential for the understanding of dynamic environments, but most sensors cannot directly collect motion information.

Some previous works [17]–[20] popularly use RGB data to estimate scene flow. Huguet et al. [17] predict scene flow through synthesizing optical flow between continuous frame images and depth map from dense stereo matching. Cech et al. [20] proposed the simple seed growing algorithm, the basic principle of which is to find correspondences in small neighborhoods around the initial seed correspondence set. Based on this principle, the disparity of the stereo image and the optical flow between consecutive images are calculated. Many researchers have also worked on scene flow estimation tasks based on RGB-D camera, which provide a depth channel for images. Some works [21], [22] extends the 2D approach to 3D to predict scene flow based on RGBD data. RGB-D flow [21] extends the two-frame variational 2D flow algorithm to 3D, and the predicted dense 3D flow applies to rigid motion segmentation. RAFT-3D [23] estimates pixelwise 3D motion on RGBD data or stereo images. RAFT-3D [23] introduces rigid-motion embeddings of pixelwise SE3, which is based on the optical flow estimation framework, RAFT [24].

The introduction of PointNet [25] has caused a wave of point cloud deep learning, which is the first deep model that processes 3D point clouds directly. PointNet [25] learns the corresponding spatial encoding for each point in the input point clouds, then uses the features of all points to obtain a global point cloud feature. PointNet [25] mainly focuses on the extraction of global point cloud features but lacks the extraction and processing of local features. The feature extraction layer of PointNet++ [26] contains sampling layer, grouping layer and pointnet layer, which provides the network with focus on extracting local features. Many recent works [6], [7], [13] are devoted to recovering 3D scene flow directly from 3D point cloud data. FlowNet3D [6] learns point cloud features based on PointNet++ [26] and introduces a new flow embedding to learn point motion. FlowNet3D [6] is a classic supervised model that estimates the scene flow directly from the raw point cloud. HPLFlowNet [27] uses bilateral convolutional layers as the base module and then recovers the 3D scene flow using a similar structure to FlowNet3D (downsampling-flow embedding-upsampling). Inspired by the optical flow estimation framework PWC-Net [28], PointPWC-Net [13] introduces a new cost volume layer based on PointConv [29] and computes the flow in a coarse-to-fine style. Wang et al. [7] introduced a hierarchical attention network in the task of the scene flow estimation, and propose a new flow embedding of dual attention to learn 3D scene flow.

The ground truth values of scene flow in real world scenes are difficult to obtain, which leads to the scarcity of scene flow label data. So self-supervised learning scene flow has important research values for scene perception. Some recent works [15], [30]–[33] has been done on unsupervised learning of scene flow. Mittal et al. [15] proposes nearest neighbor loss and cycle consistency loss for self-supervised learning of 3D scene flow, and achieves outstanding performance. Pontes et al. [30] introduces a geometrically interpretable objective function to recover the scene flow from a pair of point clouds without ground truth. PointPWC-Net [13] introduced three self-supervised losses including Chamfer distance loss, Smoothness constraint loss, and Laplacian regularization loss in their framework for scene flow estimation. Based on the HPLFlowNet [27], Tishchenko et al. [31] conducted end-to-end joint learning of ego-motion and non-rigid flow by feature learning of 3D point clouds, and used the loss functions of Mittal et al. [15] for self-supervised learning of 3D scene flow.

Goodfellow et al. [34] first proposed GAN (Generative Adversarial Nets). GAN is great at unsupervised learning and generating data, and it also has powerful representation capabilities. Many works have migrated the training ideas of GAN to different research fields. GANVO [35] introduced joint unsupervised learning of pose and depth maps based on GAN and proposed a novel adversarial technique to generate depth images without ground truth. PoseGAN [36] applied the idea of GAN to the camera localization framework. PoseGAN [36] designs the image generator by pose-to-image, based on a conditional discriminator to discriminate whether the image comes from generated or trained data. MFGAN [37] transfers beneficial features from bright scenes to images of scenes with poor lighting conditions based on generative adversarial networks, and this style transfer approach improves performance in the visual odometry task.

III. SFGAN for unsupervised learning method of 3D scene flow

In this section, a new unsupervised learning structure of 3D scene flow is proposed. As shown in Figure 1, we introduce the game idea of GAN (Generative Adversarial Network) into unsupervised 3D scene flow estimation. The new structure includes a scene flow generator G_{sf} and a point

cloud discriminator D_{pc}. The generator G_{sf} learns the 3D scene flow SF from a pair of point clouds PC_1 and PC_2. The predicted point cloud PC_2^* of the second frame can be synthesized based on the learned scene flow SF and the point cloud PC_1 of the first frame. The designed discriminator can discriminate the probability of the point cloud being real data. The discriminator considers synthesized cloud point of the second frame as a fake point cloud. The probability value of the discriminator output reflects the degree of truth or fake of the input point cloud data. A higher probability value represents the greater the possibility that the input point cloud is from real data. The range of probability values is 0 to 1. The discriminator plays the adversarial role against the generator. The estimated accuracy of the 3D scene flow is continuously improved in the process of adversarial learning. Details of the model structure are presented in the following subsections.

A. Scene Flow Generator

The first part of the proposed model is to directly estimate the 3D scene flow from the raw point cloud pair by the scene flow generator G_{sf}, which is based on the FlowNet3D [6]. The detailed structure of the generator is shown in the upper part of Figure 2. The set conv layer processes a point cloud $PC = \{c_i, pf_i | i = 1, 2, ..., n\}$ and returns a new point cloud $PC' = \{c'_j, pf'_j | j = 1, 2, ..., m\}$. $c_i \in \mathbb{R}^3$ means the XYZ coordinate of a point. $pf_i \in \mathbb{R}^l$ represents the features of the point. l means the feature dimension of the point cloud. c'_j is the updated coordinate after the set conv layer. $pf_i \in \mathbb{R}^{l'}$ is the updated point cloud feature, where l' is the updated feature dimension. The flow embedding layer learns a flow embedding d_i for each point in the point cloud of the first frame. The output of the flow embedding layer is represented as $\{c_k, d_k | k = 1, 2, ..., n\}$, where $d_k \in \mathbb{R}^{l_1}$. l_1 means the dimension of flow embedding feature. And then the flow embedding associated with the intermediate points is upsampled to the raw point cloud. This upsampling process is implemented through the learnable set upconv layer. The learnable upconv layer propagates flow embedding by aggregating features of neighboring points. The upconv layer learns how to weight the features of nearby points during network training. The scene flow $\overrightarrow{sf}$ of the raw point cloud is predicted in the last layer, which is realized by the full connection layer FC (fully connected layer).

B. Scene Flow Warping

As shown in Figure 1, we can generate the predicted point cloud PC_2^* of the second frame according to the predicted scene flow SF. The predicted scene flow from the generator G_{sf} is represented as $SF = \{\overrightarrow{sf_i} \in \mathbb{R}^3\}_{i=1}^{N_1}$. The point cloud PC_2^* of the predicted second frame is fed into the discriminator. Predicted point cloud PC_2^* is synthesized from the point cloud $PC_1 = \{pc_{1,i} \in \mathbb{R}^3\}_{i=1}^{N_1}$ of the first frame and the predicted scene flow SF. The formula for the synthesis process is as follows:

$$PC_2^* = \{pc_{2,i}^* = pc_{1,i} + \overrightarrow{sf_i} | pc_{1,i} \in PC_1, \overrightarrow{sf_i} \in SF\}_{i=1}^{N_1}. \quad (1)$$

C. 3D Structure Consistency

As shown in Figure 1, The scene flow generator G_{sf} is the main object of optimization in the whole network. We adopt a total of 5 loss functions to optimize the scene flow generator G_{sf}. They are Chamfer Loss $\mathscr{L}_C$, Curvature Loss Φ_C, Smooth Loss $\mathscr{L}_S$, Cycle Consistency Loss $\mathscr{L}_{CC}$, and GAN Loss $\mathscr{L}_G$. So that the scene flow generator G_{sf} can produce more accurate scene flow $\overrightarrow{sf}$. The first four loss functions are originated from existing unsupervised learning work [13], [15].

The PC_2^* and the PC_2 represent the point cloud from the scene flow warping and the actual point cloud of the second frame. The average distance between the points in the first frame point cloud and the points in the second frame point cloud is used to represent the distance between the two frames. The purpose of Chamfer loss is to minimize the distance between point cloud PC_2^* and point cloud PC_2. The Chamfer loss $\mathscr{L}_C$ is defined as the following:

$$\begin{aligned}\mathscr{L}_C(PC_2^*, PC_2) = &\sum_{pc_2^* \epsilon PC_2^*} \min_{pc_2 \in PC_2} \|pc_2^* - pc_2\|_2^2 + \\ &\sum_{pc_2 \epsilon PC_2} \min_{pc_2^* \in PC_2^*} \|pc_2^* - pc_2\|_2^2.\end{aligned} \quad (2)$$

To prevent large differences in the scene flow within a local space and to keep local smoothing. Smooth loss function $\mathscr{L}_S(D)$ assumes that the scene flow $SF(pc_i)$ at a point pc_i should be similar to the scene flow $SF(pc_j)$ at a point pc_j in the local space $N(pc_i)$ at pc_i. $N(pc_i)$ represents the point cloud within a local space in the point cloud PC_1. $\|N(pc_i)\|$ represents the number of points in the local space $N(pc_i)$. The detailed calculation process of $\mathscr{L}_S$ is as follows:

$$\mathscr{L}_S(D) = \sum_{pc_i \in PC_1} \frac{1}{\|N(pc_i)\|} \sum_{pc_j \in N(pc_i)} \|SF(pc_j) - SF(pc_i)\|_2^2. \quad (3)$$

The points in the 3D point cloud exists only on the surface of the object. The Curvature Loss function aims that the surface features of the predicted point cloud should be similar to the surface features of the actual point cloud. Based on the assumption of consistency of objects in front and back frames, the predicted points pc_2^* and interpolated point pc_{in} should have the same Laplace coordinates $\delta(pc)$, where pc_{in} comes from a point cloud PC_2 interpolated into the predicted point cloud PC_2^*. The curvature loss are defined as follows:

$$\delta(pc_i) = \frac{1}{N(pc_i)} \sum_{pc_j \in N(pc_i)} (pc_j - pc_i). \quad (4)$$

$$\Phi_C(\delta(pc_2^*), \delta(pc_{in})) = \sum_{pc_2^* \in PC_2^*} \|\delta(pc_2^*) - \delta(pc_{in})\|_2^2. \quad (5)$$

According to the predicted scene flow $\overrightarrow{sf}$, the coordinates c_i of the current point can be transformed to the coordinates c'_i of the next frame point. On the contrary the reverse 3D scene flow sf'_i can be estimated based on the point cloud of the next frame and the point cloud of the current frame. Based on the

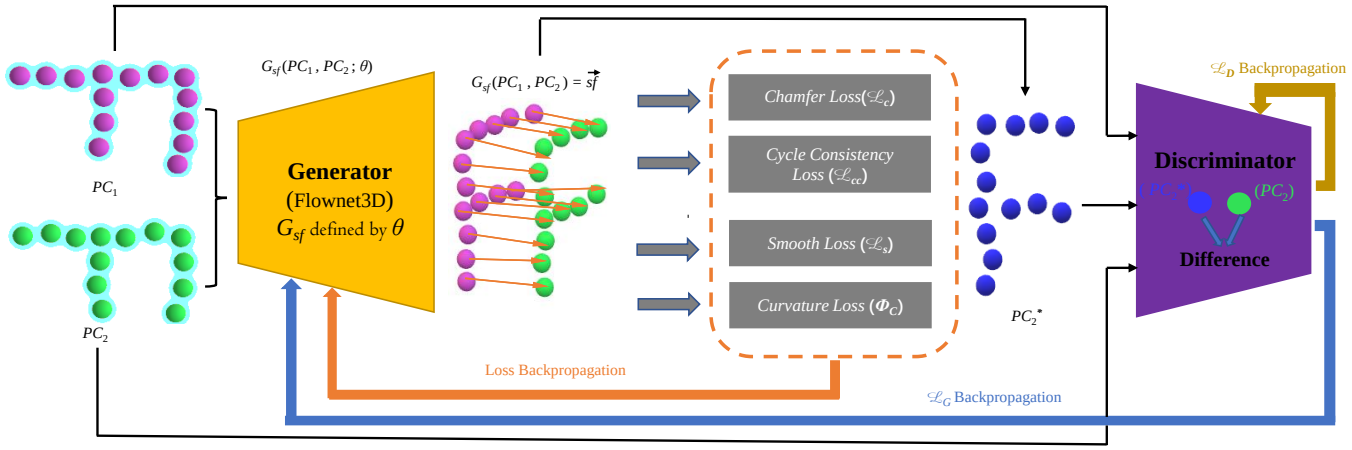

Figure 1. **Unsupervised learning model of 3D scene flow based on GAN.** The point clouds of consecutive frames (purple point cloud PC_1 and green point cloud PC_2) are fed to the scene flow generator G_{sf}, and the output is the 3D scene flow for each point in point cloud PC_1, with θ being the learnable parameter of G_{sf}. The predicted point cloud PC_2^* of the second frame is generated by scene flow warping (PC_1 + SF). We design generator loss $\mathscr{L}_G$ and discriminator loss $\mathscr{L}_D$ according to the probability values generated by comparing the difference between PC_2 and PC_2^*, which are used to optimize the scene flow generator and point cloud discriminator, respectively.

predicted second frame point cloud and the reverse flow, the predicted point cloud of the first frame can be synthesized. the aim of the Cycle consistency loss function $\mathscr{L}_{CC}$ is to minimize the distance between the predicted point cloud of the first frame and the real point cloud of the first frame. In order to make the reverse 3D scene flow sf_i' estimation more stable and reliable, a new second frame point cloud is reconstructed from sampling the predicted point cloud of the second frame and the actual second frame point cloud separately. Finally the coordinates $\{c_i''\}_i^N$ of the predicted point cloud of the first frame are obtained. The goal of the cycle consistency loss $\mathscr{L}_{CC}$ is to minimize the distance between coordinate c_i and coordinate c_i'', which is defined as follows:

$$\mathscr{L}_{CC} = \sum_i^N \|c_i'' - c_i\|^2. \tag{6}$$

As shown in the bottom half of Figure 2, the discriminator D_{pc} discriminates the predicted point cloud PC_2^* of the second frame and the real point cloud PC_2 of the second frame at the same time and outputs two probability values. More details of the discriminator will be described in the next subsection. The purpose of the scene flow generator G_{sf} is to produce a more accurate 3D scene flow that can fool the discriminator. In the training process of the network, information loss is produced when the distribution P_g of the generated data is fitted to approximate the distribution P of the real data, and the information loss is back-propagated to the generator G_{sf}.

The original GAN randomly samples from noise prior z and feeds into the generator network to generate new data $G(z)$. The scene flow generator G_{sf} inputs point cloud data of consecutive frames. and returns the predicted 3D scene flow SF. The predicted point cloud data PC_2^* of the second frame is synthesized from the real point cloud data PC_1 of the first frame and the prediction flow $\overrightarrow{sf}$. Sampling n samples $\{pc_2^{*(1)}, ..., pc_2^{*(n)}\}$ from the predicted point cloud PC_2^* of the second frame. The discriminator D_{pc} discriminates the generated data of the point cloud and generates a probability value of the point cloud coming from the real data by the sigmoid function, where the probability value reflects the difference between the generated data and the real data. The goal of G_{sf} is to minimize the difference. the GAN loss function $\mathscr{L}_G$ is defined as follows:

$$\mathscr{L}_G = \frac{1}{n}\sum_i^n log(1 - D_{pc}(pc_2^{*(i)})). \tag{7}$$

Five loss functions work together to optimize the scene flow generator G_{sf}. Each loss function has its own loss weight factor. The total loss function of the generator G_{sf} is shown as follows:

$$\mathscr{L}_{total} = \lambda_c\mathscr{L}_C + \lambda_s\mathscr{L}_S + \lambda_\Phi\Phi_C + \lambda_{cc}\mathscr{L}_{CC} + \lambda_g\mathscr{L}_G. \tag{8}$$

Where λ_c, λ_s, λ_Φ, λ_{cc}, and λ_g represent the weight of each loss.

D. Structural Similarity Discriminator

In pursuit of better discrimination, we designed four different discriminator structures, which can be divided into $(a)(b)$ and $(c)(d)$ in Figure 3 according to the feature extraction layer of the point cloud whether weight sharing. According to the predicted point cloud PC_2^* with PC_1 or PC_2 for flow embedding, we can classify the four discriminators into two kinds, $(a)(c)$ and $(b)(d)$. In the $(a)(c)$ structure, PC_2^* performs feature embedding with PC_1. In the $(b)(d)$ structure, PC_2^* performs feature embedding with PC_2. In the IV section, we will discuss the advantages and disadvantages of the four discriminators with the experimental data. We determine the best discriminator structure of the point cloud by ablation

experiments. As shown in the bottom half of Figure 2, point cloud PC_1 of the first frame, point cloud PC_2 of the second frame, and the predicted point cloud PC_2^* of the second frame are fed into the discriminator. First, the input point cloud is downsampled, and then the soft corresponding point of each point in the first frame point cloud is found in the second frame point cloud by the flow embedding layer. After learning the flow embedding for each point in the point cloud of the first frame, we continue to downsample using the Set Conv layer. In this process, the dimension of the point cloud becomes smaller and the dimension of the point cloud features becomes larger. However reducing the dimension of the point cloud is not the main purpose, the main purpose is to make the spatial distance of the values more meaningful. Therefore, the discriminator does not need to upsample the points to the original number of points and still can discriminate between the predicted point cloud PC_2^* and the real point cloud PC_2. This designed approach achieves both the aim of discriminating point clouds and saving computational resources. Lastly, the probability of the point cloud is calculated directly by the Multi-Layer Perceptron (MLP) and the Sigmoid function.

Generators and discriminators are trained by optimizing the loss function. In fact, they separately have their own loss functions. The real data $\{pc_2^{(1)}, ..., pc_2^{(n)}\}$ and the new data $\{pc_2^{*(1)}, ..., pc_2^{*(n)}\}$ generated by G_{sf} are fed into the D_{pc} network together for true-false discrimination. The aim of training the discriminator D_{pc} is to maximize the probability $log(D_{pc}(PC_2))$ of discriminating the real point cloud PC_2 and maximize the difference $log(1 - D_{pc}(PC_2^*))$ between the data distribution $P_g(pc^*)$ of the generated point cloud and the data distribution $P(pc)$ of the real point cloud. As the discriminative capability of D_{pc} becomes more and more powerful, an balance is eventually reached, which ensures that the point cloud data distribution generated by G_{sf} belongs to the same class as the real data distribution of the point cloud. Therefore, Better performance of the point cloud discriminator D_{pc} results in superior performance of the scene flow generator G_{sf}. The discriminator $\mathscr{L}_D$ loss is defined as follows:

$$\mathscr{L}_D = \frac{1}{n}\sum_{i}^{n}[log(D_{pc}(pc_2^{(i)})) + log(1 - D_{pc}(pc_2^{*(i)}))]. \quad (9)$$

E. The Adversarial Training

In the unsupervised framework of scene flow estimation proposed in this paper, the scene flow generator and the point cloud discriminator are trained alternately. In the adversarial learning process, the scene flow generator and the point cloud discriminator both play a minimax game. Poor discriminator performance at the beginning of training can cause the generator to develop in a bad trend. So discriminator D_{pc} should learn earlier than generator G_{sf}. This ensures that D_{pc} is adequately trained. At the beginning of training, although the predicted point cloud data are in the same feature space as the real point cloud data, the differences in their distributions are obvious, so the discriminator can easily distinguish the two. In the training phase of the generator G_{sf}, the information loss generated from the discriminator is used to optimize the generator, which makes the distribution of the predicted fake point cloud gradually coincide with the distribution of the real point cloud, and the degree of difference between both is represented by $V(G_{sf}, D_{pc})$. In adversarial learning process of the two models, the goal of D_{pc} is to make $V(G_{sf}, D_{pc})$ as large as possible and the goal of G_{sf} is to make $V(G_{sf}, D_{pc})$ as small as possible. The process of the two player game for G_{sf} and D_{pc} is as follows:

$$\begin{aligned}\min_{G_{sf}}\max_{D_{pc}} V(G_{sf}, D_{pc}) = \mathbb{E}_{pc\sim P(pc)}[log(D_{pc}(pc))] + \\ \mathbb{E}_{pc^*\sim P_g(pc^*)}[log(1 - D_{pc}(pc^*))].\end{aligned} \quad (10)$$

Where $V(G_{sf}, D_{pc})$ denotes the degree of difference between the data distributions of the generated data and the training data.

IV. Experiments

We implemented experiments with different training methods and ran self-supervised training on different datasets. The initial model is trained on a large virtual dataset at first, then unsupervised fine-tuning is run on a miniature real dataset containing scene flow annotations. Next, we explore the influence of each loss function on the model by ablation experiments, and also discussed the influence of different GAN loss weights. We designed four different discriminators of the point cloud and the effects of the different discriminators on the results were also compared in the ablation experiments.

A. Implementation Details

Our model is pre-trained in the FlyingThing3D dataset [38] by means of self-supervised learning, and the network framework is mainly based on FlowNet3D [6]. The pre-trained model was fine-tuned in the KITTI dataset [16]. The scene flow generator and the point cloud discriminator are inputted with a point cloud containing 2048 points for each frame. The feature input of the raw point cloud is given as 0. The generator and the discriminator are trained, separately. The best results have been demonstrated in experiments with alternating training methods such as training an epoch generator followed by an epoch discriminator. Rather than interval of several epochs between generator and discriminator training cycles.

The whole network framework in this paper is built on the deep learning framework Tensorflow. Our model is trained on an NVIDIA Quadro RTX 6000 GPU, with the Adam optimizer [39] used to optimize the network weights. The settings for each parameter of the Adam optimizer are a learning rate of 0.01, β_1 = 0.9, β_2 = 0.999, and a batch size of 4. The generator and the discriminator have the same optimizer configuration.

B. Datasets and Data Preprocessing

1) FlyingThings3D [38]: The dataset contains about 32k stereo images, where each pair of stereo images has its corresponding ground-truth optical flow map and ground-truth disparity map. These images are randomly extracted by ShapeNet [40] from multiple moving objects to form a

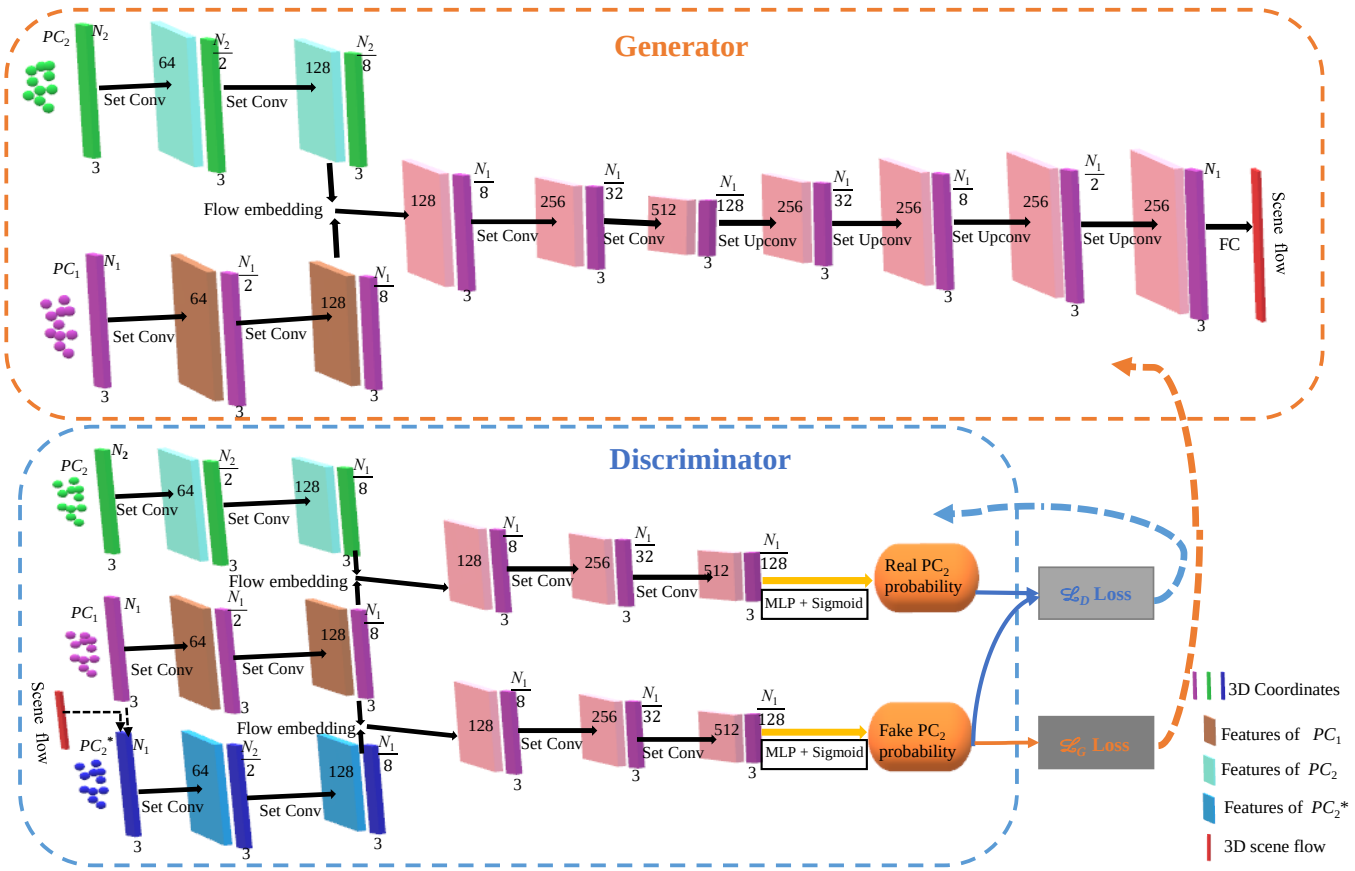

Figure 2. **Adversarial learning network framework for 3D scene flow estimation.** The FlowNet3D architecture is used as a generator to predict the scene flow at each point of the first frame PC_1 and to obtain PC_2^*. The discriminator generates the probability that PC_2 is true and the probability that PC_2^* is true, from which the loss function is designed to train the generator and the discriminator respectively. FC represents fully connected layer.

large 3D object dataset, from which 200000 are randomly selected to form the training set for our model. Our model predicts the scene flow directly from the 3D point cloud instead of RGB images. We use the same data preprocessing approach as FlowNet3D [6]. 3D point clouds are generated from ground truth disparity maps, and scene flow are recovered from ground truth optical flow maps and 3D point cloud pairs. The generated paired 3D point clouds are used for self-supervised learning of the scene flow.

2) KITTI Scene Flow Evaluation 2015 [16]: Since the evaluation of the results of the image-based scene flow estimation is realized in the image space. In the scene flow evaluation 2015, the scene flow vector stored by two frames of the disparity map and an optical flow map. The real 3D scene flow vector can be obtained by accessing the calibration file. In the training and result evaluation experiments of the point cloud-based model for 3D scene flow estimation, The ground truth data of 3D scene flow is stored by a pair of 3D point clouds at different times and a 3D scene flow vector for each point. Scene flow data was collected from 150 LIDAR data scenes in KITTI with multiple scans using 64 beam LIDAR and annotated using scene flow ground truth [41]. We select 100 scenes for training, and the remaining 50 scene are used for the evaluation of the predicted scene flow results. The same data pre-processing as FlowNet3D [6] is conducted that removes the ground points for each scene.

C. Results and Analysis

The results of the 3D scene flow estimated by SFGAN were evaluated by using the data with scene flow annotations in KITTI. We quantitatively evaluate the results of the predicted scene flow using four metrics. EPE3D represents the average error of the predicted scene flow in meters, which is expressed by the following equation: $\frac{1}{N_1}\sum_{i=1}^{N_1}\left\|\widehat{sf_i}-sf_i\right\|$, where N_1 represents the total number of scene flow. The predicted scene flow $\widehat{sf_i}$ at a point pc_i is subtracted with the scene flow ground truth sf_i, which directly reflects the mean effects of the scene flow estimation. Accuracy of scene flow estimation is measured with ACC3D and Outliers3D. ACC3D includes the absolute and relative errors of the scene flow, and sets two thresholds for the errors. ACC3D Strict specifically is expressed as: $\left\|\widehat{sf_i}-sf_i\right\|<0.05$ or $\frac{\left\|\widehat{sf_i}-sf_i\right\|}{sf_i}<5\%$; ACC3D Relax specifically is expressed as: $\left\|\widehat{sf_i}-sf_i\right\|<0.1$ or $\frac{\left\|\widehat{sf_i}-sf_i\right\|}{sf_i}<10\%$. Outliers3D represents the percentage of large errors in the predicted scene flow to the overall scene flow. Outliers3D specifically is expressed as: $\left\|\widehat{sf_i}-sf_i\right\|>0.3$ or $\frac{\left\|\widehat{sf_i}-sf_i\right\|}{sf_i}>10\%$.

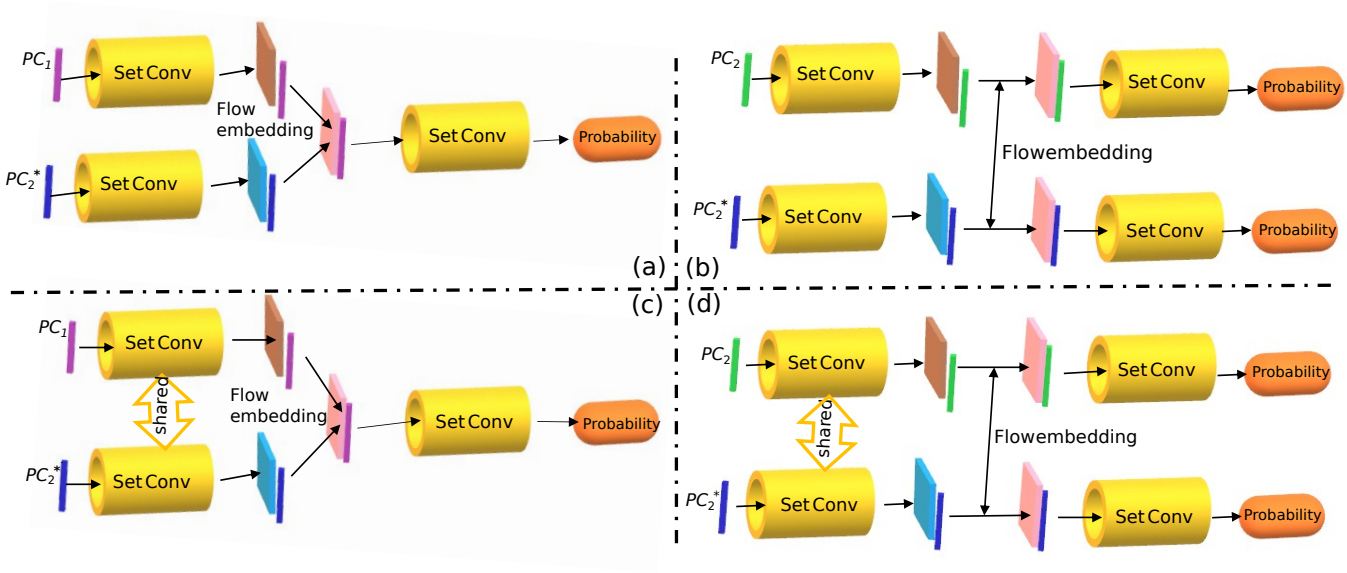

Figure 3. **Different discriminator designs.** Design base discriminator (a) no weight sharing for feature extraction of PC_1 and PC_2^*, then feature embedding of PC_1 and PC_2^*, and finally, the probability of fake PC_2 is obtained. (The probability of calculating the true PC_2 replaces the input with PC_1 and PC_2.) Discriminator (c) sets weight sharing during feature extraction. Discriminator $(b)(d)$ performs feature embedding using PC_2 and PC_2^*

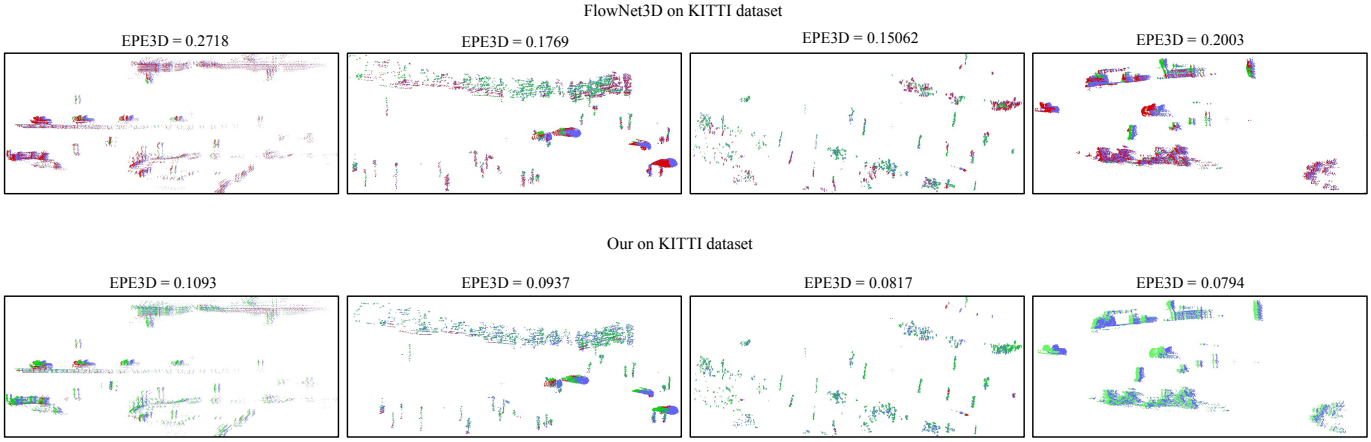

Figure 4. **Visualization of the accuracy of 3D scene flow evaluation in the KITTI dataset.** The top half shows the evaluation results of the predicted scene flow from FlowNet3D [6], and the bottom half shows the evaluation results of the predicted scene flow from our model. The point cloud of the first frame is represented by the blue points. the predicted point cloud synthesized from the predicted flow $\widehat{sf}$ and the point cloud of the first frame. We categorized the predicted points into incorrect points and correct points by utilizing the Acc3D Relax metric. Correct points are shown in green and incorrect points are shown in red. We evaluated all points of the whole scene.

As shown in Table I, first SFGAN obtains a pretrained model by utilizing for FlyingThings3D dataset, and then we did supervised fine-tuning. The supervised model fine-tuning in the KITTI dataset has achieved remarkable results. The effect of our supervised fine-tuning is obviously better than the existing supervised fine-tuning methods [15]. In no access to the ground truth of the scene flow in the KITTI dataset, the pre-trained model is fine-tuned using our method with a self-supervised manner. SFGAN has a significant improvement on the metric EPE3D, surpassing the existing self-supervised fine-tuning methods [15]. The metric EPE3D reflects the global mean error of the predicted scene flow. In fact, when compared with other methods of unsupervised learning of scene flow, SFGAN has its own special characteristic. Existing unsupervised learning methods for scene flow perceived estimation errors from the scene flow prediction results themselves, and constructed the self-supervised loss more from the estimation results for each point or localized points. However, there is no global grasp of the predicted scene flow results. SFGAN focuses more on the overall structure of the decision process and improves the whole data distribution of the prediction. Thus, the performance of self-supervised scene flow estimation can be improved. This advantage is well demonstrated by the metric EPE3D. The metric EPE3D is less than 0.1 without accessing the ground truth of scene flow. In addition, SFGAN outperforms Mittal et al. [15] on the metric ACC3D Strict.

Our model only learns the 3D scene flow from point cloud pairs. Figure 4 and Figure 5 show the visualization

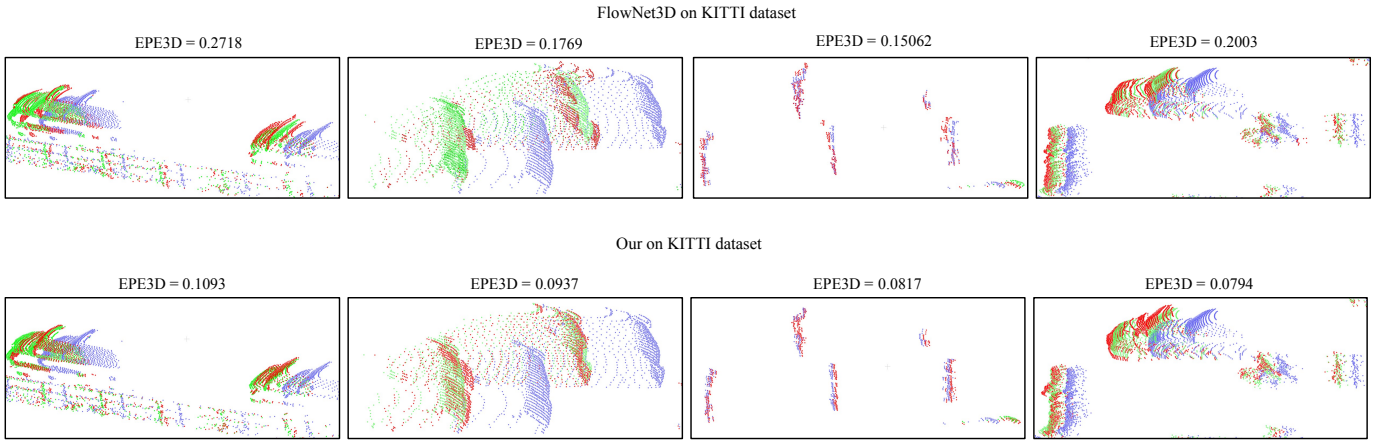

Figure 5. **Detailed visualization of scene flow estimation in KITTI dataset** The top half shows the prediction results of the scene flow of FlowNet3D [6]. The bottom half shows the prediction results of scene flow of our method.The point cloud of the first frame is blue points. The predicted point cloud and the point cloud of the second frame are red points and green points , respectively.

Table I
EVALUATION RESULTS OF THE SCENE FLOW ESTIMATES IN THE KITTI DATASET. ↑ REPRESENTS LARGER VALUES AS BETTER AND ↓ REPRESENTS SMALLER VALUES AS BETTER. W_{gan} REPRESENTS THE WEIGHT COEFFICIENT OF THE GAN LOSS FUNCTION FOR ALL MODELS, THEY WERE FIRST TRAINED IN THE FLYINGTHINGS3D DATASET AND THEN FINE-TUNED IN KITTI DATASET, AND THE GROUND POINTS OF THE SCENES WERE REMOVED DURING ALL TRAINING.

Method	EPE3D↓	Acc3D Strict↑	Acc3D Relax↑
FlowNet3D [6]	0.122	0.2537	0.5785
Mittal et al. (Supervised ft) [15]	0.100	0.3142	0.6612
Our (Supervised ft)	**0.075**	**0.4980**	**0.8117**
PointPWC-Net (Self-Supervised ft) [13]	0.163	0.2117	0.5409
Mittal et al. (Self-Supervised ft) [15]	0.126	0.3200	0.7364
Our (Self-Supervised ft) (W_{gan}=2)	0.098	0.3022	0.6823
Our (Self-Supervised ft) (W_{gan}=3)	0.102	0.3205	0.6854

of the results of the scene estimation for our method and FlowNet3D [6]. The impressive results were achieved after the baseline was fine-tuned by SFGAN. The red points in Figure 4 represent the points where the scene flow is estimated incorrectly, and the green points represent the points where it is estimated correctly. Our self-supervised method has better estimation results compared to FlowNet3D [6]. As shown in Figure 5, The point cloud (red) predicted by our method is highly similar in geometric shape to the point cloud (green) of the second frame.

D. Ablation Studies

The main focus of this section is to perform a series of ablation experiments on the loss function and point cloud discriminator of our network framework. Ablation Studies includes the contribution of each loss function to the self-supervised learning of scene flow and the effect of the weights of different GAN loss functions on the prediction accuracy of scene flow. In the ablation experiments, we also explored the effect of four different discriminators on the scene flow estimation.

In this paper, five self-supervised losses including Chamfer Loss $\mathscr{L}_C$, Curvature Loss Φ_C, Smooth Loss $\mathscr{L}_S$, Cycle Consistency Loss $\mathscr{L}_{CC}$, and GAN Loss $\mathscr{L}_{GAN}$ are used to apply to the scene flow generator. When removing the GAN losses and experimenting with only the four existing self-supervised losses, the evaluation results show a significant degradation in scene flow prediction performance. This demonstrates that introducing adversarial learning into scene flow estimation effectively improves scene flow prediction performance. In the process of adversarial learning, the stability of the distribution of predicted point cloud data can be maintained. The stability of the data distribution of the predicted point cloud was maintained during the adversarial learning process. The average scene flow endpoint error EPE3D was kept stable at about 0.1m in the absence of scene flow annotation. The performance of scene flow estimation is also degraded by removing the curvature loss function and smoothing loss function at the training process, respectively. The SFGAN framework still needs to be teamed with some existing losses to make the results optimal.

In order to make the point cloud discriminator in adversarial learning correctly discriminate whether the input point cloud comes from the generated data or the training data. As shown in Figure 3, we propose four kinds of discriminators. Figure 3 shows only the feature embedding part that PC_1 or PC_2 individually with the predicted point cloud PC_2^* of second frame (used to calculate the probability that the input point cloud is from the generated data), ignoring the feature embed-

Table II
ABLATION EXPERIMENTS ON LOSS FUNCTIONS. ALTHOUGH THESE EXISTING SELF-SUPERVISED LOSS FUNCTIONS HAVE SOME DRAWBACKS, THEIR ADVANTAGES CAN STILL IMPROVE THE PERFORMANCE OF SCENE FLOW ESTIMATION. OUR METHOD TAKES A DIFFERENT PERSPECTIVE AND COMPLEMENTS THE ADVANTAGES OF OTHER LOSS FUNCTIONS. THE EFFECT OF DIFFERENT SELF-SUPERVISED LOSSES ON THE EVALUATION RESULTS WAS STUDIED. SINCE THE REMOVAL OF CHAMFER LOSS AND CYCLE CONSISTENCY LOSS HAS GREAT FLUCTUATIONS IN THE PREDICTION RESULTS, THESE TWO LOSSES ARE ALWAYS RETAINED IN THE EXPERIMENT.

$\mathscr{L}_C$	$\mathscr{L}_{CC}$	$\mathscr{L}_S$	Φ_C	$\mathscr{L}_{GAN}$	EPE3D↓	Acc3D Strict↑	Acc3D Relax↑	Outliers↓
-	✓	✓	✓	✓	0.4061	0.0072	0.0775	0.9979
✓	-	✓	✓	✓	0.4269	0.0029	0.0166	0.9839
✓	✓	-	✓	✓	0.1314	0.1576	0.5463	0.6805
✓	✓	✓	-	✓	0.1218	0.1822	0.5470	0.6929
✓	✓	✓	✓	-	0.1194	0.1851	0.5812	0.6571
✓	✓	✓	✓	✓	**0.0987**	**0.3022**	**0.6823**	**0.5584**

Table III
THE EFFECT OF THE CHANGE IN WEIGHT VALUE W_{GAN} OF GAN LOSS ON SCENE FLOW PREDICTION. BOTH THE SCENE FLOW GENERATOR AND THE POINT CLOUD DISCRIMINATOR SHARE THE GAN LOSS WEIGHT VALUES W_{GAN} IN THE RESPECTIVE BACK PROPAGATION. MODELS ARE SELF-SUPERVISED TRAINED IN FLYINGTHINGS3D AND KITTI.

W_{GAN}	EPE3D↓	Acc3D Strict↑	Acc3D Relax↑	Outliers↓
1.0	0.1077	0.2520	0.6403	0.6555
2.0	**0.0987**	0.3022	0.6823	0.5584
3.0	0.1021	**0.3205**	**0.6854**	**0.5532**
4.0	0.1078	0.3167	0.6799	0.5604

Table IV
ABLATION EXPERIMENTS OF DESIGNING POINT CLOUD DISCRIMINATORS. THE ARROW DIRECTION MEANS THE DIRECTION OF FEATURE EMBEDDING. THE TWO ENDS OF THE ARROW INDICATE THE OBJECT WHICH IS TO PERFORM FEATURE EMBEDDING. SHARED INDICATES WHETHER THE SET CONV LAYER SHARES THE WEIGHTS, WHERE THE SET CONV LAYER IS THE FEATURE EXTRACTION LAYER OF THE POINT CLOUD.

Embading method	shared	EPE3D↓	Acc3D Strict↑	Acc3D Relax↑	Outliers↓
$PC_1 \Rightarrow PC_2^*$ $PC_1 \Rightarrow PC_2$	✓	**0.0987**	**0.3022**	**0.6823**	**0.5584**
$PC_1 \Rightarrow PC_2^*$ $PC_1 \Rightarrow PC_2$	-	0.1021	0.2934	0.6768	0.5688
$PC_2 \Rightarrow PC_2^*$ $PC_2^* \Rightarrow PC_2$	✓	0.1048	0.2608	0.6673	0.5941
$PC_2 \Rightarrow PC_2^*$ $PC_2^* \Rightarrow PC_2$	-	0.1155	0.1996	0.5928	0.6845

ding part that PC_1 or PC_2^* individually with the true point cloud PC_2 of second frame (used to calculate the probability that the input point cloud is from the training data). Since both have the same internal structure and only the input is different. As shown in Table IV, we perform ablation experiments for different point cloud discriminators. The most performance improvement in scene flow prediction is achieved when PC_1 and PC_2^* perform feature embedding and PC_1 and PC_2 perform feature embedding. When determining the objects for feature embedding, setting the Set Conv layer shared weights is more beneficial to improve the performance of scene flow prediction.

V. CONCLUSION

In the paper, we propose a novel framework for self-supervised learning of scene flow, introducing generative adversarial learning methods in scene flow learning. We use the scene flow estimator as the scene flow generator G_{sf}, and design a new point cloud discriminator D_{pc} and GAN loss function. Experimental results demonstrate the effectiveness of G_{sf} and D_{pc} adversarial learning for the task of scene flow estimation. No scene flow ground truth was used in the training process of the scene flow estimation network. The proposed method outperforms the baseline and some existing unsupervised learning methods in scene flow estimation and achieves more accurate prediction results of the scene flow on the KITTI, a widespread real-world autonomous driving dataset.

REFERENCES

[1] A. Behl, O. Hosseini Jafari, S. Karthik Mustikovela, H. Abu Alhaija, C. Rother, and A. Geiger, "Bounding boxes, segmentations and object coordinates: How important is recognition for 3d scene flow estimation in autonomous driving scenarios?" in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2574–2583.

[2] P. Lenz, J. Ziegler, A. Geiger, and M. Roser, "Sparse scene flow segmentation for moving object detection in urban environments," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, 2011, pp. 926–932.

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

[4] G. Wang, X. Wu, Z. Liu, and H. Wang, "Pwclo-net: Deep lidar odometry in 3d point clouds using hierarchical embedding mask optimization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 910–15 919.

[5] P. Wang, W. Li, Z. Gao, Y. Zhang, C. Tang, and P. Ogunbona, "Scene flow to action map: A new representation for rgb-d based action recognition with convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 595–604.

[6] X. Liu, C. R. Qi, and L. J. Guibas, "Flownet3d: Learning scene flow in 3d point clouds," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[7] G. Wang, X. Wu, Z. Liu, and H. Wang, "Hierarchical attention learning of scene flow in 3d point clouds," *IEEE Transactions on Image Processing*, vol. 30, pp. 5168–5181, 2021.

[8] G. Puy, A. Boulch, and R. Marlet, "Flot: Scene flow on point clouds guided by optimal transport," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII 16*. Springer, 2020, pp. 527–544.

[9] R. Li, G. Lin, T. He, F. Liu, and C. Shen, "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 (CVPR)*, June 2021, pp. 364–373.

[10] B. Li, C. Zheng, S. Giancola, and B. Ghanem, "Sctn: Sparse convolution-transformer network for scene flow estimation," *arXiv preprint arXiv:2105.04447*, 2021.

[11] G. Wang, S. Ren, and H. Wang, "Nccflow: Unsupervised learning of optical flow with non-occlusion from geometry," *arXiv preprint arXiv:2107.03610*, 2021.

[12] G. Wang, C. Zhang, H. Wang, J. Wang, Y. Wang, and X. Wang, "Unsupervised learning of depth, optical flow and pose with occlusion from 3d geometry," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[13] W. Wu, Z. Y. Wang, Z. Li, W. Liu, and F. Li, *PointPWC-Net: Cost Volume on Point Clouds for (Self-)Supervised Scene Flow Estimation*. Computer Vision – ECCV 2020, 2020.

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

[15] H. Mittal, B. Okorn, and D. Held, "Just go with the flow: Self-supervised scene flow estimation," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[16] A. Geiger, P. Lenz, and R. Urtasun, "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.

[17] F. Huguet and F. Devernay, "A variational method for scene flow estimation from stereo sequences," in *International Conference on Computer Vision (ICCV)*, 2007.

[18] J. P. Pons, R. Keriven, and O. Faugeras, "Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score," *International Journal of Computer Vision*, vol. 72, no. 2, pp. 179–193, 2007.

[19] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Computer Vision Pattern Recognition*, 2015.

[20] J. Cech, J. Sanchez-Riera, and R. Horaud, "Scene flow estimation by growing correspondence seeds," in *IEEE Conference on Computer Vision Pattern Recognition*, 2011.

[21] E. Herbst, X. Ren, and D. Fox, "Rgb-d flow: Dense 3-d motion estimation using color and depth," in *IEEE International Conference on Robotics Automation*, 2013.

[22] M. Jaimez, M. Souiai, J. Gonzalez-Jimenez, and D. Cremers, "A primal-dual framework for real-time dense rgb-d scene flow," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[23] Z. Teed and J. Deng, "Raft-3d: Scene flow using rigid-motion embeddings," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 8375–8384.

[24] Z. Teed and J. Deng, "Raft: Recurrent all-pairs field transforms for optical flow," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 402–419.

[25] C. R. Qi, H. Su, K. Mo, and L. 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*, 2017, pp. 652–660.

[26] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *CoRR*, vol. abs/1706.02413, 2017. [Online]. Available: http://arxiv.org/abs/1706.02413

[27] X. Gu, Y. Wang, C. Wu, J. L. Yong, and P. Wang, "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)*, 2019.

[28] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[29] W. Wu, Z. Qi, and L. Fuxin, "Pointconv: Deep convolutional networks on 3d point clouds," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9613–9622.

[30] J. K. Pontes, J. Hays, and S. Lucey, "Scene flow from point clouds with or without learning," in *2020 International Conference on 3D Vision (3DV)*, 2020, pp. 261–270.

[31] I. Tishchenko, S. Lombardi, M. R. Oswald, and M. Pollefeys, "Self-supervised learning of non-rigid residual flow and ego-motion," in *2020 International Conference on 3D Vision (3DV)*, 2020, pp. 150–159.

[32] R. Li, G. Lin, and L. Xie, "Self-point-flow: Self-supervised scene flow estimation from point clouds with optimal transport and random walk," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 15 577–15 586.

[33] B. Ouyang and D. Raviv, "Occlusion guided self-supervised scene flow estimation on 3d point clouds," *arXiv preprint arXiv:2104.04724*, 2021.

[34] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Advances in Neural Information Processing Systems*, vol. 3, pp. 2672–2680, 2014.

[35] Y. Almalioglu, M. Saputra, P. Gusmao, A. Markham, and N. Trigoni, "Ganvo: Unsupervised deep monocular visual odometry and depth estimation with generative adversarial networks," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019.

[36] K. Liu, Q. Li, and G. Qiu, "Posegan: A pose-to-image translation framework for camera localization," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 166, pp. 308–315, 2020.

[37] E. Jung, N. Yang, and D. Cremers, "Multi-frame gan: image enhancement for stereo visual odometry in low light," in *Conference on Robot Learning*. PMLR, 2020, pp. 651–660.

[38] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. 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 (CVPR)*, June 2016.

[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

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

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