Deformation and Correspondence Aware Unsupervised Synthetic-to-Real Scene Flow Estimation for Point Clouds

Zhao Jin[1]   Yinjie Lei[1,*]   Naveed Akhtar[2]   Haifeng Li[3]   Munawar Hayat[4]

[1]Sichuan University [2]The University of Western Australia [3]Central South University [4]Monash University

jinzhao@stu.scu.edu.cn   yinjie@scu.edu.cn   naveed.akhtar@uwa.edu.au

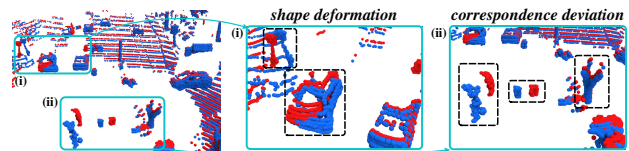lihaifeng@csu.edu.cn   munawar.hayat@monash.edu

Abstract

Point cloud scene flow estimation is of practical importance for dynamic scene navigation in autonomous driving. Since scene flow labels are hard to obtain, current methods train their models on synthetic data and transfer them to real scenes. However, large disparities between existing synthetic datasets and real scenes lead to poor model transfer. We make two major contributions to address that. First, we develop a point cloud collector and scene flow annotator for GTA-V engine to automatically obtain diverse realistic training samples without human intervention. With that, we develop a large-scale synthetic scene flow dataset GTA-SF. Second, we propose a mean-teacher-based domain adaptation framework that leverages self-generated pseudo-labels of the target domain. It also explicitly incorporates shape deformation regularization and surface correspondence refinement to address distortions and misalignments in domain transfer. Through extensive experiments, we show that our GTA-SF dataset leads to a consistent boost in model generalization to three real datasets (i.e., Waymo, Lyft and KITTI) as compared to the most widely used FT3D dataset. Moreover, our framework achieves superior adaptation performance on six source-target dataset pairs, remarkably closing the average domain gap by 60%. Data and codes are available at https://github.com/leolyj/DCA-SRSFE

1. Introduction

Scene flow estimation aims to predict the 3D motion field from two consecutive input frames. As a generalization of 2D optical flow, scene flow represents 3D motion of objects and can be used to predict their movement in the future, which is meaningful in robotic navigation and autonomous driving. In the early years, scene flow was estimated from stereo or RGB-D images [13, 44, 45, 49]. With the recent advances in 3D sensing and data driven technologies, learning scene flow directly from point clouds has

First Frame   Second Frame   Prediction   Ground Truth
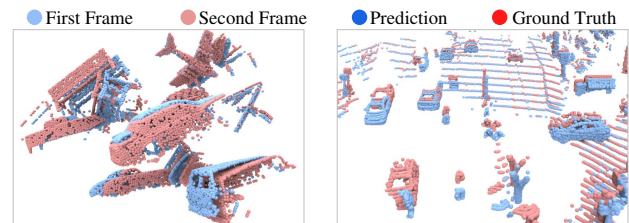
(a) Differences between FT3D [27] (left) and Waymo [17, 40] (right)

shape deformation   *correspondence deviation*

(b) Inaccurate predictions (*i.e.,* results of adding estimated scene flow to 1st frame) caused by training on FT3D [27] and testing on Waymo [17,40]

Figure 1. Challenges for Synthetic-to-Real Scene Flow Estimation (SRSFE). (a) Existing synthetic dataset FT3D [27] (left) stacks and moves ShapeNet [38] objects for data generation, resulting in unnatural scenes distinct from real data (right), *e.g.,* Waymo [17, 40]. (b) Due to domain shift, SRSFE fails to maintain local structure and accurate movement, leading to shape deformation and correspondence deviation. The lack of appropriate synthetic datasets and performance drop in SRSFE motivate our work.

gained significant research attention [2, 7, 8, 17, 24, 32].

Obtaining training data for scene flow estimation requires 3D motion vector annotation of each point in the scene, which is extremely challenging. One pragmatic solution is to employ synthetic data for training, whose annotations can be directly generated. However, training on synthetic data and testing on real data for scene flow estimation, *i.e.,* Synthetic-to-Real Scene Flow Estimation (SRSFE), faces two major challenges. First, SRSFE research on point clouds is still in its infancy, and currently there is a lack of synthetic data that adequately captures the real-world dynamics for this task. The only public synthetic data for SRSFE on point clouds, *i.e.,* FlyingThings3D [27] (FT3D), is generated by randomly moving 3D objects sampled from ShapeNet [38]. This simplistic process leads to unnatural scene flows in the data, see Fig. 1(a). Second,

---

*Corresponding Author: Yinjie Lei (yinjie@scu.edu.cn)

SRSFE must overcome the inevitable domain gap caused by the synthetic-to-real setting. In recent years, extensive studies have been conducted on Unsupervised Domain Adaptation (UDA), which adapts a model to unseen unlabeled data to mitigate the domain gap problem. However, most of the existing UDA methods [3,4,6,9,10,41,42] are designed for 2D tasks to address the domain gap caused by the variance of image texture, color and illumination. Much less attention has been paid to UDA for point clouds [34, 53–56]. This is especially true for SRSFE, for which no systematic study is available to date. Compared to other static point cloud tasks, SRSFE has a peculiar requirement of learning correlations between dynamic points. Hence, existing UDA methods are not readily transferable to this task.

In this paper, we address the above two problems. First, we propose a synthetic point cloud scene flow dataset GTA-V Scene Flow (GTA-SF), to address the lack of dataset. Our data leverages GTA-V engine [5] to simulate LiDAR scanning and autonomously annotate scene flow by aligning the identical entities rendered by the engine. Compared to FT3D, GTA-SF has more realistic scenes and point cloud representation. Secondly, to bridge the synthetic-to-real domain gap, we propose a UDA framework specifically designed for the SRSFE task. Observing that '*shape deformation*' and '*correspondence deviation*' are the key contributors to performance degradation in SRSFE - Fig. 1(b), our technique learns deformation and correspondence under a mean-teacher strategy. We constrain the teacher predictions with rigid shapes and induce a deformation-aware student model to learn desirable scene flow. To address correspondence deviation, we leverage object surface relationships to let the model learn better correspondence on real data.

Our extensive experiments show that our dataset GTA-SF shows remarkable generalization to real-world data, and the proposed framework is highly effective in reducing the domain gap for the point cloud SRSFE problem. In brief, our contributions can be summarized as follows

- We present the first (to the best of our knowledge) systematic study on bridging the domain gap in synthetic to real-world scene flow estimation for point clouds.

- We develop a point cloud sequence collector and scene flow annotator for GTA-V engine, and create a large-scale dataset GTA-SF for the SRSFE task.

- We propose a mean-teacher domain adaptation framework for point cloud SRSFE that explicitly addresses shape deformations and correspondence deviation.

- With extensive experiments, we demonstrate our GTA-SF is closer to real as it enables better performance on real dataset, and our technique consistently surpass common UDA methods across multiple datasets.

2. Related Works

Scene Flow Estimation on Point Clouds: Scene flow estimation problem is first introduced and defined as a 3D motion field of points in [43]. Early works [11, 13, 44–46, 49] estimate 3D scene flow from stereo images or RGB-D sequences [13, 44, 45, 49]. More recently, with the popularity of 3D sensors, an increasing number of techniques focus on learning scene flow directly from 3D point clouds [7, 8, 20, 22, 24, 30, 32, 50, 52]. For instance, Liu *et al.* [24] leveraged PointNet++ [33] for feature extraction and proposed a flow embedding layer for cross-frame geometric relation learning. In [8], Gu *et al.* took advantage of permutohedral lattice projection and designed a Bilateral Convolutional Layer for two consecutive frames. Optimal transport is utilized to guide scene flow learning in [23, 32]. Wei *et al.* [50] constructed point-voxel correlation fields to capture local and long-range relations among points. Gojcic *et al.* [7] proposed a weakly supervised approach to learn rigid scene flow by only using binary background segmentation and ego-motion annotations. Instances of self-supervised approaches to scene flow estimation can also be found in [30, 52]. Although scene flow estimation from 3D point clouds has shown promises, the domain gap between the real and synthetic data dramatically degrades generalization abilities of the current models in real-world settings. We address this limitation in our proposed framework.

Unsupervised Domain Adaptation (UDA): aims to generalize a model trained on a source domain to an unlabeled target domain. The UDA has demonstrated remarkable performance on 2D vision tasks. For 3D point clouds, UDA is also explored for shape classification [1, 34], semantic segmentation [15, 31, 51, 55] and object detection [26, 48, 53, 54, 56]. Among these contributions, Qin *et al.* [34] proposed to learn domain-invariant point cloud representation by global and local feature alignment. Yi *et al.* [55] used a surface completion network to transform both source and target point clouds into a canonical domain, and trained a shared segmentation network. Yang *et al.* [54] adopted self-training with memory bank-based pseudo-label generation and curriculum data augmentation for UDA on 3D detection. Luo *et al.* [26] addressed the problem of inaccurate box-scale by adopting multi-level consistency regularization for the target domain with teacher-student paradigm. In general, successful application of UDA requires addressing task-specific domain shift challenges. For the synthetic-to-real scene flow estimation task, this problem is still unaddressed in the existing literature. Hence, it is the main contribution of this paper.

Synthetic-to-Real Transfer Learning: Training with synthetic data is widely used to avoid laborious annotation process [16, 21, 27, 35–37]. Gaming engines, e.g., Grand Theft Auto V (GTA-V), have proven useful for generating synthetic data for various 2D vision tasks, *e.g.*, seman-
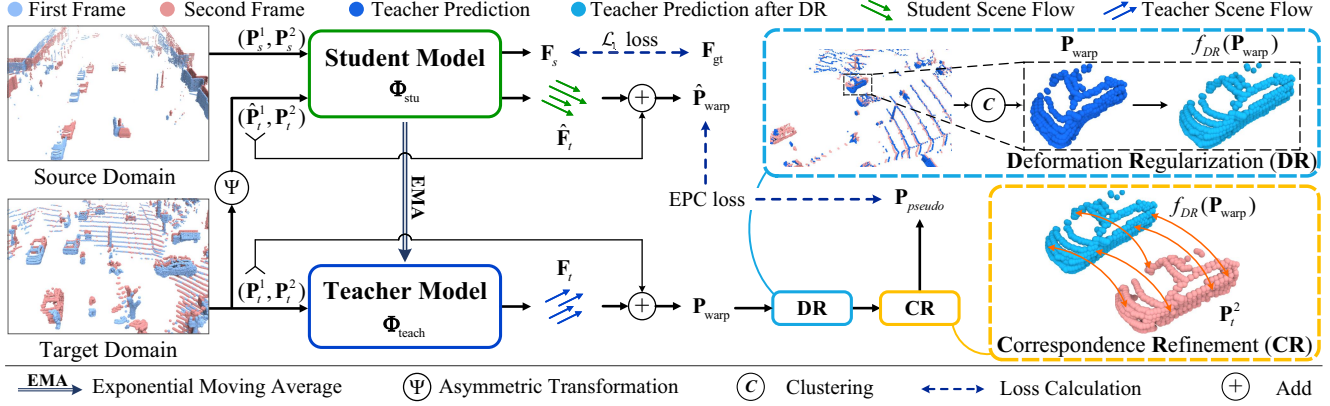
Figure 2. Framework schematics: Our teacher model Φ_{teach} is updated as the Exponential Moving Average (EMA) of the student model weights Φ_{stu}. The first and second frames of the source domain, $\mathbf{P}_s^1$ and $\mathbf{P}_s^2$, are input to the student for which ground truth scene flow $\mathbf{F}_{\text{gt}}$ is available for supervision. The student also expects a transformed version $(\hat{\mathbf{P}}_t^1, \mathbf{P}_t^2)$ of the target domain inputs $(\mathbf{P}_t^1, \mathbf{P}_t^2)$ provided to the teacher. An End-Point Consistency (EPC) is eventually imposed between the scene flow predictions of the teacher $\mathbf{F}_t$ and the student $\hat{\mathbf{F}}_t$. For EPC, both predictions are first added with the first frame of inputs to obtain $\mathbf{P}_{\text{warp}}$ and $\hat{\mathbf{P}}_{\text{warp}}$. The teacher prediction is subsequently regularized for deformation and refined for correspondence, thereby providing high-quality pseudo-labels $\mathbf{P}_{pseudo}$ to compute EPC loss.

tic segmentation [36], optical flow [21, 35], object detection [16] and crowd counting [47]. Recently, GTA-V has also been used for point cloud data generation for 3D object detection [14], semantic segmentation [51] and 3D mesh reconstruction [12]. For scene flow estimation, the existing synthetic dataset FT3D [27] constructs scenes with ShapeNet [38] objects moving along random 3D trajectories. Although useful, significant differences between such scenes and real-world scenarios lead to poor model generalisation on real data. We address it by leveraging the established efficacy of GTA-V to generate more realistic scene flow dataset.

3. Methodology

3.1. Problem Formulation

For our problem, we consider a set of labeled data, $\mathcal{S} = \{(\mathbf{P}_s^i, \mathbf{P}_s^{i+1}, \mathbf{F}_{\text{gt}}^i) \mid_{i=1}^{N_s}\}$ for the source domain. Here, $\mathbf{P}_s^i$ and $\mathbf{P}_s^{i+1}$ are two successive point cloud frames and $\mathbf{F}_{\text{gt}}^i$ is the ground truth scene flow between them. We have $|\mathcal{S}| = N_s$ samples available for the source domain. Correspondingly, we have a set $\mathcal{T} = \{(\mathbf{P}_t^i, \mathbf{P}_t^{i+1}, \mathbf{F}_{\text{test}}^i) \mid_{i=1}^{N_t}\}$ for the target domain, for which $\mathbf{P}_t^i$ and $\mathbf{P}_t^{i+1}$ are consecutive frames and N_t denotes the number of samples. For $\mathcal{T}$, the scene flow $\mathbf{F}_{\text{test}}^i$ is unknown. The objective of synthetic-to-real scene flow estimation is to compute an estimator $\Lambda(\mathcal{S}, \mathbf{P}_t^i) \rightarrow \mathbf{F}_{\text{test}}^i, \forall \mathbf{P}_t^i \in \mathcal{T}$ such that the source domain is restricted to synthetic data only, and the target domain is the real-world. Due to the large domain shift between the considered $\mathcal{S}$ and $\mathcal{T}$, the estimator $\Lambda(.)$ needs to be robust to the domain gap. We adopt the unsupervised domain adaption (UDA) paradigm to address that. The problem is referred to as SRSFE for synthetic-to-real scene flow estimation.

3.2. Technique Overview

Teacher-Student Paradigm Schematics of our UDA framework for point cloud SRSFE is provided in Fig. 2. Our framework employs a student model Φ_{stu} and a teacher model Φ_{teach}. In the text, we alternatively use these symbols to refer to model weights for brevity. In our technique, we apply back-propagation to update Φ_{stu}, whereas stop gradient is used for Φ_{teach}. For the latter, we use the Exponential Moving Average (EMA) of Φ_{stu} to iteratively update the weights as

$$\Phi_{\text{teach}}^{\text{updated}} \leftarrow \alpha \Phi_{\text{teach}} + (1 - \alpha)\Phi_{\text{stu}}, \tag{1}$$

where α is a smoothing coefficient that dictates the update rate of the teacher model.

Asymmetric Transformation We encourage domain invariance in the student model by eventually forcing its prediction to match the teacher prediction for an input that is a known transform of the teacher input from the target domain. As scene flow is learned from dynamic point cloud sequences, the correlation between two consecutive frames plays a key role in scene flow estimation. Considering that, we define an asymmetric transformation operation $\Psi(.,.)$, for the input (*i.e.,* two consecutive frames of point clouds). The operator Ψ stochastically applies a transformation to the first frame of the input, and leaves the second frame unchanged. We consider global rotation and translation for $\Psi(.,.)$, which alter the position without disrupting shapes of objects. Consider an input with $\mathbf{P}_t^1$ as the first frame and $\mathbf{P}_t^2$ as the next frame, the transformation is performed over the input as

$$\Psi(\mathbf{P}_t^1, \mathbf{P}_t^2) = (\hat{\mathbf{P}}_t^1, \mathbf{P}_t^2), \tag{2}$$

where $\hat{\mathbf{P}}_t^1$ is the transformed version of $\mathbf{P}_t^1$. Through $\Psi(.,.)$, the models robustly comprehend the notion of correlations

between consecutive frames.

End-Point Consistency Ideally, adding scene flow to the first frame should provide an estimate of the second frame. The first frame added by scene flow is referred as 'warped frame'. We can promote the teacher-student consistency by enforcing the target domain warped frame prediction for the two models to be similar. For this, we introduce the notion of End-Point Consistency (EPC) between $\mathbf{P}_{\mathrm{warp}}$ and $\hat{\mathbf{P}}_{\mathrm{warp}}$, such that $\mathbf{P}_{\mathrm{warp}} = \mathbf{P}_t^1 + \mathbf{F}_t$ and $\hat{\mathbf{P}}_{\mathrm{warp}} = \hat{\mathbf{P}}_t^1 + \hat{\mathbf{F}}_t$, where $\mathbf{F}_t$ and $\hat{\mathbf{F}}_t$ are the predicted scene flows of the teacher and student models respectively.

As the teacher model provides pseudo labels for the student model, we propose to improve the quality of pseudo labels to teach a better student on the target domain. To that end, for the teacher prediction, the warped frame is further processed by deformation regularization (§ 3.3) to maintain rigid shapes. We also propose a subsequent correspondence refinement (§ 3.4) for better surface alignment of objects. By forcing the student prediction on the original target domain input (after applying Ψ) to be consistent with the teacher model prediction, we effectively encourage domain invariance in the student by enhancing its robustness to input perturbations. It also promotes target domain deformation and correspondence awareness in the student.

3.3. Deformation Regularization (DR)

For the SRSFE problem, a model induced over synthetic data must generalize to real-world data. Since synthetic data generally does not faithfully capture real-world details, the model may fail to fully comprehend object shape in the target domain at the desired granularity level. This causes scene flow vectors to have distorted object shapes.

To address the problem, we design a Deformation Regularization (DR) module as a deformation corrector for rigid bodies. Specifically, for the warped point cloud $\mathbf{P}_{\mathrm{warp}}$ of the teacher model, we first segment it into several distinct clusters $\{\mathbf{C}_l \mid_{l=1}^{N_c}\}$, where N_c is the number of clusters. We then employ the Kabsch algorithm [18] to estimate a rigid motion $(\mathbf{R}_l, \mathbf{t}_l)$ for each cluster $\mathbf{C}_l$ from $\mathbf{P}_t^1$ to $\mathbf{P}_{\mathrm{warp}}$, where $\mathbf{R}_l \in \mathbb{R}^{3\times3}$ and $\mathbf{t}_l \in \mathbb{R}^3$ denote the rotation and translation matrices. The reconstructed cluster $\mathbf{C}_l'$ are then obtained as

$$\mathbf{C}_l' = \left\{ (\mathbf{C}_l^1 \cdot \mathbf{R}_l + \mathbf{t}_l) \mid_{i=1}^{n_l} \right\}, \quad (3)$$

where $\mathbf{C}_l^1$ indicates the corresponding points in $\mathbf{P}_t^1$ of $\mathbf{C}_l$ and n_l is the number of points in $\mathbf{C}_l$. Let us write that after applying DR, our teacher warped result $\mathbf{P}_{\mathrm{warp}}$ is reconstructed as $f_{DR}(\mathbf{P}_{\mathrm{warp}}) = \{\mathbf{C}_l' \mid_{l=1}^{N_c}\}$. Then, EPC ensures consistency between the student warped results $\hat{\mathbf{P}}_{\mathrm{warp}}$ and $f_{DR}(\mathbf{P}_{\mathrm{warp}})$. This encourages shape distortion awareness in the student model, thereby allowing it to learn adaptive deformations for the target domain. The $f_{DR}(\mathbf{P}_{\mathrm{warp}})$ is later improved with correspondence refinement.
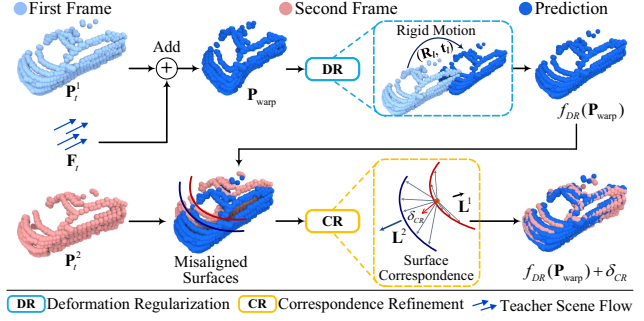

Figure 3. Illustration of the proposed DR and CR to generate high-quality pseudo labels using the teacher model stream for guiding the student. The DR improves object shape and CR improves surface correspondence by refining local geometry.

3.4. Correspondence Refinement (CR)

Ideally, a model for SRSFE must be able to maintain correct correspondence between the objects in consecutive frames in the target domain. However, synthetic objects in the source domain have distinct boundaries and geometrically simplified surfaces, whereas the real-world object shapes are much more complex. This is problematic for automatically preserving the desired correspondences when the model is applied in the targeted domain.

To address that, we explicitly encourage cross-frame surface correspondence in our model. Let us briefly consider the second frame $\mathbf{P}_t^2$ in an input $(\mathbf{P}_t^1, \mathbf{P}_t^2)$ as the target frame. The scene flow $\mathbf{F}_t$ identifies per-point translations that move the first frame $\mathbf{P}_t^1$ to match the second frame $\mathbf{P}_t^2$. In our setup, if the scene flow vectors are correctly estimated, the objects in $\mathbf{P}_{\mathrm{warp}}$ and $\mathbf{P}_t^2$ will have the same surfaces. The accuracy of the estimated scene flow therefore directly depends on how well the geometric surfaces match across the frames. To account for cross-frame surface correspondence, we employ the Laplacian coordinate [39] that records local geometric characteristics of 3D surfaces. We start with computing the Laplacian coordinate $\mathbf{L}^1$ of each point p_{warp}^j within the warped point cloud $\mathbf{P}_{\mathrm{warp}}$ as

$$\mathbf{L}^1(p_{\mathrm{warp}}^j) = \frac{1}{|\mathcal{N}(p_{\mathrm{warp}}^j, \mathbf{P}_{\mathrm{warp}})|} \sum_{k=1}^{K} (p_{\mathrm{warp}}^k - p_{\mathrm{warp}}^j), \quad (4)$$

where $\mathcal{N}(p_{\mathrm{warp}}^j, \mathbf{P}_{\mathrm{warp}})$ calculates K nearest neighbors of p_{warp}^j in $\mathbf{P}_{\mathrm{warp}}$ and $p_{\mathrm{warp}}^k \in \mathcal{N}(p_{\mathrm{warp}}^j, \mathbf{P}_{\mathrm{warp}})$. We then extend to cross-frame correspondence by querying neighboring points in the second frame $\mathbf{P}_t^2$ as

$$\mathbf{L}^2(p_{\mathrm{warp}}^j) = \frac{1}{|\mathcal{N}(p_{\mathrm{warp}}^j, \mathbf{P}_t^2)|} \sum_{k=1}^{K} (p_2^k - p_{\mathrm{warp}}^j), \quad (5)$$

where $\mathcal{N}(p_{\mathrm{warp}}^j, \mathbf{P}_t^2)$ means neighboring points of p_{warp}^j in $\mathbf{P}_t^2$ and $p_2^k \in \mathcal{N}(p_{\mathrm{warp}}^j, \mathbf{P}_t^2)$. The discrepancy between $\mathbf{L}^1(p_{\mathrm{warp}}^j)$ and $\mathbf{L}^2(p_{\mathrm{warp}}^j)$ will provide cues for the surface

mis-alignment, which can be utilized to make refinement for pseudo labels of the teacher model. As individual objects are separated and reconstructed for DR, we further compute refinement vectors for each reconstructed cluster $\mathbf{C}'_l$ to maintain the rigid shape as

$$\widetilde{\mathbf{C}}'_l = \mathbf{C}'_l + \frac{1}{N_{\mathbf{C}'_l}} \sum_{p_u \in \mathbf{C}'_l} \left(\mathbf{L}^2(p_u) - \mathbf{L}^1(p_u) \right), \quad (6)$$

where $\widetilde{\mathbf{C}}_l$ represents the refined cluster and $N_{\mathbf{C}'_l}$ is the number of points in cluster $\mathbf{C}'_l$. After the refinement of individual clusters, we denote the refined warped point clouds as $\mathbf{P}_{pseudo}$, which is the final pseudo labels and obtained by

$$\mathbf{P}_{pseudo} = f_{DR}(\mathbf{P}_{\text{warp}}) + \delta_{CR} = \{\widetilde{\mathbf{C}}'_l|_{l=1}^{N_c}\}, \quad (7)$$

where δ_{CR} denotes refinement vectors of our CR. By improving the reconstructed point clouds with CR, teacher scene flow is adjusted for better surface alignment, such that the teacher model stream is able to provide more reliable pseudo labels for the warped point clouds, which are subsequently exploited in student training. We illustrate CR applied after DR for pseudo label improvement in Fig. 3.

3.5. Network Training

Our network comprises a student model $\mathbf{\Phi}_{\text{stu}}$ and a teacher model $\mathbf{\Phi}_{\text{teach}}$. In each iteration, $\mathbf{\Phi}_{\text{stu}}$ is trained with a supervised loss $\mathcal{L}_{\text{source}}$ defined over the source domain and a consistency loss $\mathcal{L}_{\text{EPC}}$ over the target domain. $\mathcal{L}_{\text{source}}$ is the $\mathcal{L}_1$ loss of the between the ground truth scene flow $\mathbf{F}_{\text{gt}}$ and student estimated scene flow $\mathbf{F}_s$, i.e.,

$$\mathcal{L}_{\text{source}}\left(\mathbf{\Phi}_{\text{stu}}\right) = ||\mathbf{F}_s - \mathbf{F}_{\text{gt}}||_1. \quad (8)$$

For $\mathcal{L}_{\text{EPC}}$, we use the $\mathcal{L}_1$ loss between the warped results of the student $\hat{\mathbf{P}}_{\text{warp}}$ and the teacher $\mathbf{P}_{\text{warp}}$, after applying DR and CR, i.e.,

$$\mathcal{L}_{\text{EPC}}\left(\mathbf{\Phi}_{\text{stu}}\right) = ||\hat{\mathbf{P}}_{\text{warp}} - (f_{DR}(\mathbf{P}_{\text{warp}}) + \delta_{\text{CR}})||_1. \quad (9)$$

The total loss $\mathcal{L}_{\text{stu}}$ for the student model is given by

$$\mathcal{L}_{\text{stu}}\left(\mathbf{\Phi}_{\text{stu}}\right) = \mathcal{L}_{\text{source}}\left(\mathbf{\Phi}_{\text{stu}}\right) + \mathcal{L}_{\text{EPC}}\left(\mathbf{\Phi}_{\text{stu}}\right). \quad (10)$$

For the teacher model, we update its weights $\mathbf{\Phi}_{\text{teach}}$ after each iteration using Eq. (1).

4. GTA-V Scene Flow (GTA-SF) Dataset

Another major contribution of this paper is the curation of a large-scale synthetic scene flow dataset, generated using GTA-V [5]. Below, we first describe our method to collect consecutive LiDAR point clouds and annotate scene flow labels for them automatically in GTA-V engine (§ 4.1). We then discuss the properties of our dataset in comparison with the existing synthetic scene flow datasets (§ 4.2).

4.1. Data Collection

We collect data using GTA-V engine [5] based on Scrip Hook V [1] and PreSIL [14]. Specifically, we first build a scenario with an autonomous driving car on the road. Then, we attach a synthetic LiDAR collector on the top of the car and collect point clouds at a predefined frequency (e.g., 10Hz). With the help of Scrip Hook V, we are able to interact with GTA-V and load properties (e.g., position, belonging entity) of each point. In order to annotate scene flow vectors, we follow the rigidity assumption of Jund et al. [17], and calculate rigid motion for each entity. During the game running, each individual object is assigned a unique entity ID, which is considered a rigid body. We can directly compute scene flow $\mathbf{f}_i$ for each point $\mathbf{p}_i$ by loading location $\{x, y, z\}$ and pose $\{\alpha, \beta, \gamma\}$ of its entity e_i as

$$\mathbf{f}_i = \left((\mathbf{p}_i - \mathbf{P}_{e_i}) \cdot \mathbf{R}_{e_i}^{-1} \cdot \dot{\mathbf{R}}_{e_i} + \dot{\mathbf{P}}_{e_i} \right) - \mathbf{p}_i, \quad (11)$$

where $\mathbf{P}_{e_i}$ and $\mathbf{R}_{e_i}$ are respectively the entity position and rotation matrix in current frame, while $\dot{\mathbf{P}}_{e_i}$ and $\dot{\mathbf{R}}_{e_i}$ are those matrices in the next frame.

For the points without a corresponding entity in the next frame, we compute ego-motion for their scene flow analogous to entity-motion. Specifically, the location and pose of the LiDAR are kept consistent with the attached car, and the ego-car motion is computed as the scene flow for the unmatched entities. For scene flow estimation, ground points are uninformative. Hence, they are manually removed by existing works with height thresholding [8, 24]. Since roads are not always flat, thresholding leads to errors, including undesired removal of foreground object points. In GTA-SF, we systematically remove ground points by exploiting the entity information, i.e., remove points belonging to the ground entities, which helps in better data quality.

4.2. Dataset Properties

The proposed GTA-SF is a large-scale synthetic dataset for real-world scene flow estimation. It contains 54,287 pairs of consecutive point clouds with densely annotated scene flow. Compared to existing synthetic datasets, GTA-SF collects more realistic point clouds with larger scale, and annotates scene flow beyond point correspondence assumption to fit to physical truth. In terms of diversity, GTA-SF covers a variety of scenarios including downtown, highway, streets and other driving areas rather than artificial scenes. The point clouds are collected along six different routes pertaining various outdoor areas. Moreover, the collected point clouds are high-quality for scene flow learning since the meaningless ground points are carefully removed. We provide more detailed illustrations and quantitative analyses of the properties discussed above in the supplementary material.

---

[1]http://www.dev-c.com/gtav/scripthookv/

Table 1. Comparison between FT3D [27] and our GTA-SF.

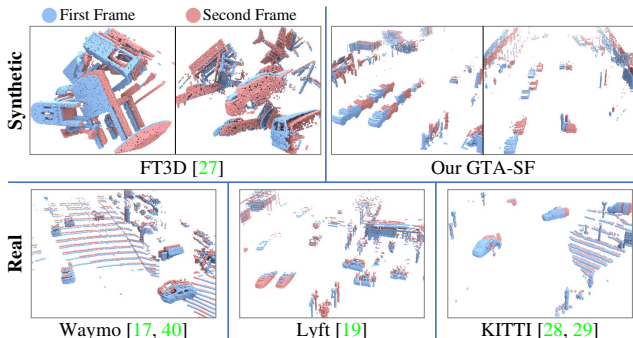|	Frames	Label	Scenes
FT3D [27]	23,464	Unreal	Objects Stacking
GTA-SF (Ours)	54,287	Realistic	Vehicle Driving

Figure 4. Visual comparisons of synthetic datasets FT3D [27] and GTA-SF; and real datasets, Waymo [17, 40], Lyft [19] and KITTI [28, 29]. Compared to FT3D, our GTA-SF is more realistic.

To the best of our knowledge, the FlyingThings3D (FT3D) [27] is the only widely used synthetic dataset for point cloud scene flow estimation. It contains 19,640 training and 3,824 testing samples, which makes it smaller in size as compared to our GTA-SF. It builds scenes by stacking 3D objects from ShapeNet [38] and randomly moving them between two frames. Whereas effective, this strategy is unnatural for real-world scenes. Tab. 1 shows a brief comparison between GTA-SF and FT3D. Fig. 4 shows a visual comparison between FT3D and our GTA-SF. We also provide more comprehensive illustrations in the supplementary material. Our empirical evaluation in § 5.3 also verifies that GTA-SF considerably narrows down the domain gap between the synthetic and real-world data.

5. Experiments

5.1. Real-World Datasets

Waymo. The Waymo Open Dataset (WOD) [40] contains 158,081 training and 39,987 validation frames of point clouds with 3D object detection annotations, captured in real-world. [17] expand WOD with scene flow annotations by using the tracked bounding boxes of objects. Note that Waymo compensates for ego-motion before computing scene flow. It only considers the absolute movements of objects and the annotations for stationary object points are set to zero. However, previous datasets [27–29] make no compensation for ego-motion. The absolute movement and the relative movement caused by ego-motion are estimated simultaneously. For fair comparisons, we follow previous datasets and retrieve ego-motion vectors for the points based on pose information provided in WOD.

Lyft. The Lyft Level 5 dataset [19] consists of 18,900 train-

Table 2. Testing EPE3D (m) of existing methods pretrained on different source datasets on Waymo [17, 40].

Source→Target	FlowNet3D [24]	HPLFlowNet [8]	FLOT [32]	PV-RAFT [50]
FT3D→Waymo	0.3546	0.2505	0.3299	0.2621
GTA-SF→Waymo	0.1589	0.1146	0.1081	0.0585
Waymo→Waymo	0.1067	0.0501	0.0680	0.0433

ing and 3,780 validation frames of LiDAR point clouds. It is usually used for 3D object detection and does not provide scene flow labels. As the detection labels and sensor parameters are given, we follow [17] and generate scene flow labels for Lyft without compensating for ego-motion.

KITTI. KITTI Scene Flow 2015 [28, 29] is a popular benchmark for scene flow estimation. We follow the preprocessing steps in [8] to generate point clouds with scene flow annotation, which contains 142 pairs of frames.

5.2. Evaluation Setup

Implementation Details. For real-world datasets, we first transform them into the same coordinate system and remove ground points with height $< 0.3m$. For our framework, see the supplementary material for implementation details.

Comparison Methods. We provide comparison with the following methods. (1) *Baseline* indicates pretraining the model on source domain and directly evaluating it on the target domain. (2) *MMD* [25] adopts Maximum Mean Discrepancy (MMD) for cross-domain feature alignment. (3) *Self-Ensemble* [4] adopts mean-teacher with $\mathcal{L}_1$ loss between student and teacher estimated scene flow vectors, and (4) *Oracle* trains fully-supervised model on target domain.

Evaluation Metrics. Following [8, 24, 32, 50], we adopt four evaluation metrics. The metrics are calculated between estimated scene flow $\mathbf{F}$ and ground truth $\mathbf{F}_{\text{gt}}$.

EPE3D (EPE) (m): $||\mathbf{F} - \mathbf{F}_{\text{gt}}||_2$ computes the l_2 distance between the estimated and ground truth scene flow vectors.
ACC Strict (AS) (%): is the percentage of points with EPE3D $< 0.05m$ or relative error $< 5\%$.
ACC Relax (AR) (%): is the percentage of points with EPE3D $< 0.1m$ or relative error $< 10\%$.
Outliers (Out) (%): is the percentage of points with EPE3D $> 0.3m$ or relative error $> 10\%$.

5.3. Experimental Results

Comparison of Synthetic Datasets. We first verify the existence of synthetic-to-real domain gap by evaluating recent methods FlowNet3D [24], HPLFlowNet [8], FLOT [32] and PV-RAFT [50]. We train them on three datasets: FT3D [27], GTA-SF and Waymo [17, 40], and then evaluate them on Waymo. Our experimental results in Tab. 2 show that the models trained on FT3D face a serious performance gap, as compared to direct training on Waymo. It verifies the existence of large domain gap between synthetic and real datasets. Comparing FT3D and GTA-SF, we see that GTA-

Table 3. Performance comparison on six source-target pairs. HPLFlowNet [8] is used as the baseline directly transferred from source to target. Comparison considers Synthetic-to-Real (S→R) and Real-to-Real (R→R) transfer. EPE (m), AS (%), AR (%) and Out (%) are scene flow estimation evaluation metrics. '-' indicates Oracle results on KITTI are not available since no training data is provided. Best results for S→R are in bold. ↓ and ↑ respectively indicate negative and positive polarity. E, D and C mean EPC, DR and CR respectively.

	Methods	GTA-SF⟶Waymo				GTA-SF⟶Lyft				GTA-SF⟶KITTI			
		EPE↓	AS↑	AR↑	Out↓	EPE↓	AS↑	AR↑	Out↓	EPE↓	AS↑	AR↑	Out↓
S→R	Baseline [8]	0.1061	32.35	66.21	65.35	0.1802	28.36	73.66	34.93	0.0932	52.29	81.39	33.75
	MMD [25]	0.1068	37.24	68.85	66.91	0.1563	41.91	78.51	31.35	0.0877	45.87	79.45	41.69
	Self-Ensemble [4]	0.0981	44.05	71.05	62.10	0.1681	32.80	75.59	34.05	0.0869	51.11	79.18	37.49
	Ours(E)	0.0894	38.68	75.68	61.68	0.1506	38.75	80.22	29.06	0.0848	51.01	81.18	37.52
	Ours(E+D)	0.0887	40.60	77.30	60.70	0.1454	41.11	81.49	29.06	0.0748	54.11	86.53	31.58
	Ours(E+D+C)	**0.0683**	**58.57**	**87.98**	**47.40**	**0.1277**	**56.35**	**85.50**	**24.62**	**0.0464**	**80.53**	**96.85**	**18.75**
R→R	Oracle	0.0501	74.82	92.20	40.88	0.1058	68.92	87.04	22.68	-	-	-	-

	Methods	FT3D⟶Waymo				FT3D⟶Lyft				FT3D⟶KITTI			
		EPE↓	AS↑	AR↑	Out↓	EPE↓	AS↑	AR↑	Out↓	EPE↓	AS↑	AR↑	Out↓
S→R	Baseline [8]	0.2477	31.59	57.22	77.08	0.8486	13.18	30.42	79.10	0.1169	47.83	77.76	41.03
	MMD [25]	0.2179	24.12	55.09	79.52	0.7158	10.48	29.21	80.05	0.1165	37.42	78.46	42.75
	Self-Ensemble [4]	0.2342	33.20	55.54	78.72	0.7366	13.20	32.52	77.23	0.1166	41.88	77.15	44.11
	Ours(E)	0.2339	28.44	55.76	77.84	0.7330	10.22	29.24	80.35	0.1193	40.50	75.75	46.25
	Ours(E+D)	0.2091	29.56	56.18	78.97	0.5092	13.25	35.61	74.52	0.0992	46.86	81.95	37.83
	Ours(E+D+C)	**0.1251**	**48.87**	**78.40**	**57.29**	**0.4442**	**25.90**	**51.61**	**58.59**	**0.0516**	**79.37**	**96.81**	**18.04**
R→R	Oracle	0.0501	74.82	92.20	40.88	0.1058	68.92	87.04	22.68	-	-	-	-

Table 4. Comparisons of PV-RAFT [50] baseline from synthetic datasets to Waymo. Our UDA framework achieves remarkable performance in closing the domain gap for PV-RAFT. ↓ and ↑ respectively indicate negative and positive polarity.

	Methods	EPE↓	AS↑	AR↑	Out↓
	GTA-SF→Waymo				
S→R	Baseline [50]	0.0585	71.38	90.74	42.15
	Ours	**0.0474**	**79.93**	**94.14**	**35.61**
R→R	Oracle	0.0433	84.70	95.07	33.09
	FT3D→Waymo				
S→R	Baseline [50]	0.2620	43.59	68.25	63.21
	Ours	**0.1219**	**62.29**	**82.53**	**47.71**
R→R	Oracle	0.0433	84.70	95.07	33.09

Table 5. Comparisons of FLOT [32] baseline from synthetic datasets to Waymo. Our UDA framework is compatible with FLOT and shows consistent performance in closing domain gap. ↓ and ↑ respectively indicate negative and positive polarity.

	Methods	EPE↓	AS↑	AR↑	Out↓
	GTA-SF→Waymo				
S→R	Baseline [32]	0.1081	45.36	75.72	57.41
	Ours	**0.0888**	**59.15**	**82.58**	**49.96**
R→R	Oracle	0.0680	72.78	89.66	41.94
	FT3D→Waymo				
S→R	Baseline [32]	0.3299	27.07	48.20	78.63
	Ours	**0.1432**	**52.19**	**75.69**	**56.36**
R→R	Oracle	0.0680	72.78	89.66	41.94

SF→Waymo has a much smaller performance gap. Similar trends are apparent on Lyft and KITTI in Tab. 3, establishing that our GTA-SF is more compatible to real data.

Synthetic-to-Real Transfer. In Tab. 3, we compare our domain adaptation method for synthetic-to-real scene flow estimation with HPLFlowNet [8] Baseline, Oracle and two general-purpose UDA methods (*i.e.,* MMD [25] and Self-Ensemble [4]). The results on six source-target pairs demonstrate the superior performance of our method, and its capability to largely close the performance gap between Baseline and Oracle by 60% in EPE. Note that the Oracle results on KITTI are not available since no training data is provided, we also achieve 55.86% improvement in EPE. Compared with general-purpose UDA methods, our framework surpasses them on all four evaluation metrics because our technique enables scene flow estimator to be deforma-

tion and correspondence aware in the target domain. Also, compared to FT3D, we observe a smaller performance gap for the GTA-SF transferred models.

Our framework also shows remarkable compatibility with other mainstream scene flow estimators. As shown in Tab. 4 and Tab. 5, it achieves superior performance in closing the domain gap for PV-RAFT [50] and FLOT [32], which is consistent with HPLFlowNet [8]. Tab. 4 shows the domain adaptation performance of our framework with PV-RAFT on GTA-SF→Waymo and FT3D→Waymo. We can narrow the performance gap by 62.04% to 72.50% on EPE. In Tab. 5, the results on FLOT also show similar trends.

5.4. Ablation Study

We conduct ablation studies with HPLFlowNet as the baseline to evaluate the contribution of individual components of our framework in the overall performance.

Table 6. Contribution of Mean Teacher (MT) in our framework. ↓ and ↑ respectively indicate negative and positive polarity.

Settings	EPE↓	AS↑	AR↑	Out↓
without MT	0.0768	51.17	85.72	53.81
with MT	**0.0683**	**58.57**	**87.98**	**47.40**

Table 7. Ablation on Asymmetric Transformation ($\mathcal{AT}$) and augmentations. Symmetric Transformation ($\mathcal{ST}$) has same transformation for both frames. $\mathcal{T}$ & $\mathcal{R}$ mean Translation & Rotation. ↓ and ↑ respectively indicate negative and positive polarity.

Methods		EPE↓	AS↑	AR↑	Out↓
$\mathcal{ST}$		0.0988	37.72	73.05	59.63
$\mathcal{AT}$	$\mathcal{T}$	0.0719	54.84	88.31	47.77
	$\mathcal{T}+\mathcal{R}$	0.0702	55.28	88.27	47.73
	$\mathcal{R}$	**0.0683**	**58.57**	**87.98**	**47.40**

Effectiveness of DR and CR. We investigate the effectiveness of the two key modules (*i.e.,* DR and CR) in our framework by progressively adding them. In Tab. 3, Ours (E) indicates our baseline using End-Point Consistency without DR and CR. Addition of DR and CR are shown with +D and +C in the table. It can be seen that DR consistently boosts the performances on all source-target dataset pairs. By further adding CR, our framework achieves 39.4% to 56.8% improvement from FT3D to real datasets and 15.21% to 45.28% improvement from GTA-SF to real datasets. This demonstrates explicit contributions of DR and CR in our framework. Since there is a larger domain gap between FT3D and real data, DR and CR are able to bring more improvements for the transferred models.

Effectiveness of Mean Teacher. Our framework adopts mean teacher to provide pseudo labels for the unlabeled target domain. To verify the effectiveness of our teacher model, we conduct experiments by replacing it with a model identical to the student by setting α in EMA to 0. Tab. 6 shows that the performance of our framework drops after removing the teacher model. This identifies the contribution of mean teacher to provide positive supervision.

Effectiveness of Asymmetric Transformation. We propose Asymmetric Transformation ($\mathcal{AT}$) as the augmentation strategy for the student model in our framework. $\mathcal{AT}$ transforms the first frame of input point cloud pairs with stochastic augmentations. Tab. 7 conducts an ablation study to evaluate the efficacy of $\mathcal{AT}$. As compared to Symmetric Transformation ($\mathcal{ST}$) using the same augmentation for both input frames, $\mathcal{AT}$ enables better performance. We analyze the effects of different augmentations in $\mathcal{AT}$. Compared with translation ($\mathcal{T}$) or translation+rotation ($\mathcal{T}+\mathcal{R}$), using rotation ($\mathcal{R}$) only yields the best results. This makes rotation a more suitable augmentation for SRSFE since it brings realistic scene flow in accord with motor steering.

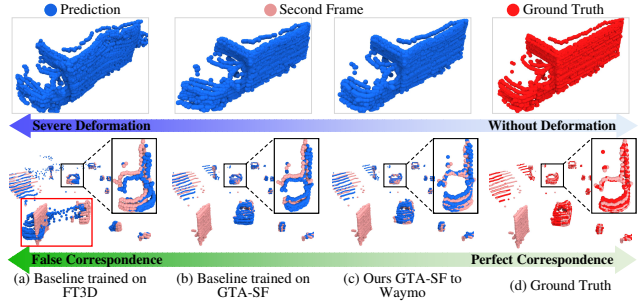**Analysis of Deformation and Correspondence.** Based on

Figure 5. Qualitative comparisons on Waymo dataset. (a) Baseline trained on FT3D seriously deforms objects and provides incorrect correspondence - red box. (b) Replacing FT3D with GTA-SF results in better predictions, but deformation and misalignment still exist - black boxes. (c) Our UDA method with GTA-SF achieves the best results by incorporating DR and CR.

the observation that shape deformation and correspondence deviation are major problems for SRSFE, we make qualitative comparisons in Fig. 5 to show the advantage of addressing them. It can be seen that the baseline trained on FT3D faces severe shape deformation and correspondence deviation when directly transferred to Waymo due to large domain gap. Replacing FT3D with GTA-SF gives a significant improvement on Waymo, though the above two problems are only partially addressed. After applying our framework for unsupervised domain adaptation we achieve the best results by explicitly addressing the mentioned problems.

6. Conclusion and Limitation

We investigated synthetic-to-real scene flow estimation (SRSFE) on point clouds and addressed two major challenges for the task. First, based on the observation that a large domain gap exists between the existing synthetic datasets and real-world scenarios, we build a more realistic dataset for SRSFE using GTA-V engine. Second, to further reduce the domain gap for the induced computational models, we devise a mean-teacher-based framework for domain adaptation in scene flow estimation. Our framework incorporates end-point consistency during training with deformation regularization and correspondence refinement. We conclusively establish the reduced domain gap between our and real data by a quantitative evaluation on three real datasets. We also demonstrate a remarkable domain gap reduction over the existing baselines with our data and proposed framework. For pragmatic reasons, we employ rigidity assumption in our dataset. This only provides an approximation to non-rigid objects (*e.g,* pedestrians). Nevertheless, this does not result in observable performance degradation, yet it provides computational advantages.

We do not forsee any ethical concerns with the dataset and approach proposed in this manuscript, such as sensitive personal information, and bias against gender or race.

References

[1] Idan Achituve, Haggai Maron, and Gal Chechik. Self-supervised learning for domain adaptation on point clouds. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 123–133, 2021. 2

[2] Stefan Andreas Baur, David Josef Emmerichs, Frank Moosmann, Peter Pinggera, Bjorn Ommer, and Andreas Geiger. SLIM: Self-supervised LiDAR scene flow and motion segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13126–13136, 2021. 1

[3] Yi-Hsin Chen, Wei-Yu Chen, Yu-Ting Chen, Bo-Cheng Tsai, Yu-Chiang Frank Wang, and Min Sun. No more discrimination: Cross city adaptation of road scene segmenters. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1992–2001, 2017. 2

[4] Geoffrey French, Michal Mackiewicz, and Mark Fisher. Self-ensembling for visual domain adaptation. *arXiv preprint arXiv:1706.05208*, 2017. 2, 6, 7

[5] Rockstar Games. Grand theft auto V. http://www.rockstargames.com/V. 2, 5

[6] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 1180–1189, 2015. 2

[7] 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 (CVPR)*, pages 5692–5703, 2021. 1, 2

[8] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3254–3263, 2019. 1, 2, 5, 6, 7

[9] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 1989–1998, 2018. 2

[10] Judy Hoffman, Dequan Wang, Fisher Yu, and Trevor Darrell. FCNs in the wild: Pixel-level adversarial and constraint-based adaptation. *arXiv preprint arXiv:1612.02649*, 2016. 2

[11] Michael Hornacek, Andrew Fitzgibbon, and Carsten Rother. SphereFlow: 6 DoF scene flow from RGB-D pairs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3526–3533, 2014. 2

[12] Yuan-Ting Hu, Jiahong Wang, Raymond A Yeh, and Alexander G Schwing. SAIL-VOS 3D: A synthetic dataset and baselines for object detection and 3D mesh reconstruction from video data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1418–1428, 2021. 3

[13] Frédéric Huguet and Frédéric Devernay. A variational method for scene flow estimation from stereo sequences. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 1–7, 2007. 1, 2

[14] Braden Hurl, Krzysztof Czarnecki, and Steven Waslander. Precise synthetic image and lidar (presil) dataset for autonomous vehicle perception. In *Proceedings of IEEE Intelligent Vehicles Symposium (IV)*, pages 2522–2529, 2019. 3, 5

[15] Maximilian Jaritz, Tuan-Hung Vu, Raoul de Charette, Emilie Wirbel, and Patrick Pérez. xMUDA: Cross-modal unsupervised domain adaptation for 3D semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12605–12614, 2020. 2

[16] Matthew Johnson-Roberson, Charles Barto, Rounak Mehta, Sharath Nittur Sridhar, Karl Rosaen, and Ram Vasudevan. Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks? *arXiv preprint arXiv:1610.01983*, 2016. 2, 3

[17] Philipp Jund, Chris Sweeney, Nichola Abdo, Zhifeng Chen, and Jonathon Shlens. Scalable scene flow from point clouds in the real world. *arXiv preprint arXiv:2103.01306*, 2021. 1, 5, 6

[18] Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 32(5):922–923, 1976. 4

[19] R. Kesten, M. Usman, J. Houston, T. Pandya, K. Nadhamuni, A. Ferreira, M. Yuan, B. Low, A. Jain, P. Ondruska, S. Omari, S. Shah, A. Kulkarni, A. Kazakova, C. Tao, L. Platinsky, W. Jiang, and V. Shet. Level 5 perception dataset 2020. https://level-5.global/level5/data/, 2019. 6

[20] Yair Kittenplon, Yonina C Eldar, and Dan Raviv. Flow-Step3D: Model unrolling for self-supervised scene flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4114–4123, 2021. 2

[21] Philipp Krähenbühl. Free supervision from video games. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2955–2964, 2018. 2, 3

[22] Ruibo Li, Guosheng Lin, Tong He, Fayao Liu, and Chunhua 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)*, pages 364–373, 2021. 2

[23] Ruibo Li, Guosheng Lin, and Lihua 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)*, pages 15577–15586, 2021. 2

[24] 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 (CVPR)*, pages 529–537, 2019. 1, 2, 5, 6

[25] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu. Transfer feature learning with joint distribution adaptation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2200–2207, 2013. 6, 7

[26] Zhipeng Luo, Zhongang Cai, Changqing Zhou, Gongjie Zhang, Haiyu Zhao, Shuai Yi, Shijian Lu, Hongsheng Li, Shanghang Zhang, and Ziwei Liu. Unsupervised domain adaptive 3D detection with multi-level consistency. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8866–8875, 2021. 2

[27] 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 (CVPR)*, pages 4040–4048, 2016. 1, 2, 3, 6

[28] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3061–3070, 2015. 6

[29] 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, 2015. 6

[30] 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 (CVPR)*, pages 11177–11185, 2020. 2

[31] Duo Peng, Yinjie Lei, Wen Li, Pingping Zhang, and Yulan Guo. Sparse-to-dense feature matching: Intra and inter domain cross-modal learning in domain adaptation for 3D semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7108–7117, 2021. 2

[32] Gilles Puy, Alexandre Boulch, and Renaud Marlet. FLOT: Scene flow on point clouds guided by optimal transport. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 527–544, 2020. 1, 2, 6, 7

[33] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, pages 5099–5108, 2017. 2

[34] Can Qin, Haoxuan You, Lichen Wang, C-C Jay Kuo, and Yun Fu. PointDAN: A multi-scale 3D domain adaption network for point cloud representation. *arXiv preprint arXiv:1911.02744*, 2019. 2

[35] Stephan R Richter, Zeeshan Hayder, and Vladlen Koltun. Playing for benchmarks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2213–2222, 2017. 2, 3

[36] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 102–118, 2016. 2, 3

[37] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3234–3243, 2016. 2

[38] Manolis Savva, Angel X Chang, and Pat Hanrahan. Semantically-enriched 3D models for common-sense knowledge. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 24–31, 2015. 1, 3, 6

[39] Olga Sorkine. Laplacian mesh processing. In *Eurographics (State of the Art Reports)*, pages 53–70. Citeseer, 2005. 4

[40] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2446–2454, 2020. 1, 6

[41] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7472–7481, 2018. 2

[42] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7167–7176, 2017. 2

[43] Sundar Vedula, Simon Baker, Peter Rander, Robert Collins, and Takeo Kanade. Three-dimensional scene flow. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 722–729, 1999. 2

[44] Christoph Vogel, Konrad Schindler, and Stefan Roth. 3D scene flow estimation with a rigid motion prior. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 1291–1298, 2011. 1, 2

[45] Christoph Vogel, Konrad Schindler, and Stefan Roth. Piecewise rigid scene flow. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1377–1384, 2013. 1, 2

[46] Christoph Vogel, Konrad Schindler, and Stefan Roth. 3D scene flow estimation with a piecewise rigid scene model. *International Journal of Computer Vision (IJCV)*, 115(1):1–28, 2015. 2

[47] Qi Wang, Junyu Gao, Wei Lin, and Yuan Yuan. Learning from synthetic data for crowd counting in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8198–8207, 2019. 3

[48] Yan Wang, Xiangyu Chen, Yurong You, Li Erran Li, Bharath Hariharan, Mark Campbell, Kilian Q Weinberger, and Wei-Lun Chao. Train in germany, test in the usa: Making 3D object detectors generalize. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11713–11723, 2020. 2

[49] Andreas Wedel, Thomas Brox, Tobi Vaudrey, Clemens Rabe, Uwe Franke, and Daniel Cremers. Stereoscopic scene flow computation for 3D motion understanding. *International Journal of Computer Vision (IJCV)*, 95(1):29–51, 2011. 1, 2

[50] 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. 2, 6, 7

[51] Bichen Wu, Xuanyu Zhou, Sicheng Zhao, Xiangyu Yue, and Kurt Keutzer. SqueezeSegV2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In *Proceedings of International Conference on Robotics and Automation (ICRA)*, pages 4376–4382, 2019. 2, 3

[52] 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 *Proceedings of European Conference on Computer Vision (ECCV)*, pages 88–107, 2020. 2

[53] Qiangeng Xu, Yin Zhou, Weiyue Wang, Charles R Qi, and Dragomir Anguelov. SPG: Unsupervised domain adaptation for 3D object detection via semantic point generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15446–15456, 2021. 2

[54] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. ST3D: Self-training for unsupervised domain adaptation on 3D object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10368–10378, 2021. 2

[55] Li Yi, Boqing Gong, and Thomas Funkhouser. Complete & label: A domain adaptation approach to semantic segmentation of lidar point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15363–15373, 2021. 2

[56] Weichen Zhang, Wen Li, and Dong Xu. SRDAN: Scale-aware and range-aware domain adaptation network for cross-dataset 3D object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6769–6779, 2021. 2