SSRFlow: Semantic-aware Fusion with Spatial Temporal Re-embedding for Real-world Scene Flow

Zhiyang Lu[1] Qinghan Chen[1] Zhimin Yuan[1]
Chenglu Wen[1] Ming Cheng* Cheng Wang[1]
[1] Fujian Key Laboratory of Sensing and Computing for Smart Cities, Xiamen University
{zhiyang, chenqinghan, zhiminyuan}@stu.xmu.edu.cn

Abstract

Scene flow, which provides the 3D motion field of the first frame from two consecutive point clouds, is vital for dynamic scene perception. However, contemporary scene flow methods face three major challenges. Firstly, they only consider the context of individual point clouds before flow embedding, leading to embedded points struggling to perceive the consistent semantic relationship of another frame. To address this issue, we propose a novel approach called Dual Cross Attentive (DCA) for the latent fusion and alignment between two frames based on semantic contexts. This is then integrated into Global Fusion Flow Embedding (GF) to initialize flow embedding based on global correlations in both contextual and Euclidean spaces. Secondly, deformations exist in non-rigid objects after the warping layer, which distorts the spatiotemporal relation between the consecutive frames. For a more precise estimation of residual flow at next-level, the Spatial Temporal Re-embedding (STR) module is devised to update the point sequence features at current-level. Lastly, poor generalization is often observed due to the significant domain gap between synthetic and LiDAR-scanned datasets. We leverage novel domain adaptive losses to effectively bridge the gap of motion inference from synthetic to real-world. Experiments demonstrate that our approach achieves state-of-the-art (SOTA) performance across various datasets, with particularly outstanding results in real-world LiDAR-scanned situations.

1. Introduction

3D scene flow estimation captures the motion information of objects from two consecutive point clouds and produces the motion vector for each point in the source frame. It serves as a foundational component for perceiving dynamic environments and provides important motion features to downstream tasks, such as object tracking [42, 45, 46], point cloud label propagation [44] and pose estimation [7]. Early approaches[1, 14, 31, 37] rely on stereo or RGB-D images as input. While recent advances in deep learning-based point cloud processing have paved the way for numerous end-to-end algorithms specifically designed for scene flow prediction [4, 18, 21, 28, 33, 40]. Among them, FlowNet3D [21] presents a pioneering approach that integrates deep learning into the estimation of scene flow. By incorporating the principles of the PWC (Pyramid, Warp and Cost volume) optical flow algorithm [30], PointPWC [40] introduces the coarse-to-fine strategy to scene flow prediction. However, the PWC frameworks [4, 5, 18, 32, 34, 40] only account for scene flow regression of the local receptive field within each level, which neglects global feature matching. Hence, it is difficult to estimate precise motion for long-distance displacements and complex situations such as repetition and occlusion.

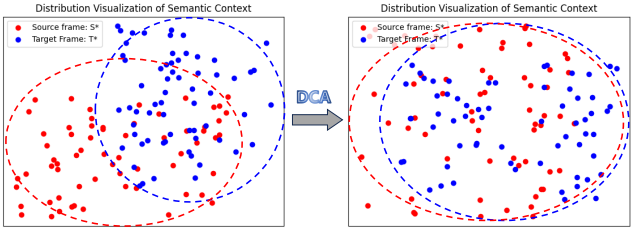


Figure 1. Visualization of the distribution of the highest level of point cloud feature pyramid before and after employing DCA. It is evident that DCA Fusion enhances the semantic alignment of the two consecutive point clouds, thereby facilitating subsequent global flow embedding.

FlowStep3D[16] and WM3DSF[33] tackle this issue by using global flow initialization in the all-to-all manner. However, they neglect the alignment of semantic space between the embedded points and the context of another frame, see Figure 1. This hard approach to global flow embedding results in ambiguous flows. Hence, inspired by the fusion and alignment capability of cross-attention[11, 26, 41], we introduce the Dual Cross Attentive (DCA) Fusion to merge the semantic contexts of point clouds from two frames in

*Corresponding author.

latent space, which allows for perceiving the semantic context of another frame before embedding. By integrating into the Global Fusion Flow Embedding (GF) module for global flow embedding, DCA Fusion aggregates embedded features in both context and Euclidean spaces, leveraging the global correlations of two consecutive point clouds.

The second issue is attributed to the warping layer, which upsamples sparse scene flow from the previous level and accumulates to the current level. Previous methods [4, 5, 17, 33, 40, 43] simply employ the information preceding the warping layer to predict the residual flow for the subsequent layer. However, the temporal relation between the consecutive frames changes during warping since the two frames become closer, and the relative spatial position of points within the source frame also transforms. Utilizing the original features could introduce bias in residual flow estimation after warping layer-by-layer. To overcome this limitation, we propose a Spatial Temporal Re-embedding (STR) module to re-embed the temporal features between the warped source frame and target frame, along with spatial features within the warped source frame per se.

Furthermore, as a point-level task, obtaining the ground truth (GT) of scene flow from real-world point clouds is difficult[24, 25], and previous methods resort to synthetic datasets[23] for training. However, they suffer from domain gaps when applied to real-world LiDAR-scanned scenes. To address this issue, we propose novel Domain Adaptive Losses (DA Losses) based on the intrinsic properties of point cloud motion, including local rigidity of dynamic objects and the cross-frame feature similarity after motion, suggesting promising results when generalized to real-world datasets.

Overall, our contributions are as follows:

- Our GF module leverages the dual cross-attentive mechanism to fuse and align the semantic context from both frames and further matches the all-to-all point-pairs globally from both latent context space and Euclidean space, enabling accurate flow initialization for subsequent residual scene flow prediction.
- We elaborate the STR module to tackle the problems caused by distortion in surface spatiotemporal sequence features of two consecutive frames after warping.
- We propose novel DA Losses that address the synthetic-to-real challenge of the scene flow task by considering the local rigidity and cross-frame feature similarity.
- Experiments demonstrate that our model achieves SOTA performance on datasets of various patterns and exhibits strong generalization on real-world LiDAR-scanned datasets.

2. Related Works

FlowNet3D [21] pioneers in leveraging deep learning network PointNet++[29] for scene flow embedding based on raw point clouds, which surpasses traditional methods by a large margin. Afterward, HPLFlowNet[13] proposed novel DownBCL, UpBCL, and CorrBCL operations inspired by Bilateral Convolutional layers to abstract and fuse structural information from consecutive point clouds. FlowNet3D++ [36] enhances FlowNet3D by incorporating geometric constraints based on point-to-plane distance and angular alignment. FESTA [35] expands on FlowNet3D by utilizing a trainable aggregate pooling to stably down-sample points instead of Farthest Point Sampling (FPS). These above methods employ the SetConv[21], composed of PointNet++, to conduct local flow embedding from two frames. However, this local embedding approach lacks global representation and fails to multi-scale processing.

Inspired by [30] in optical flow, PointPWC [40] incorporates the Pyramid, Warp, and Cost volume (PWC) to scene flow estimation. PointPWC utilizes semantic features from point cloud pyramids at different levels to generate the cost volume, which is then used to compute patch-to-patch local flow embedding. HALF [32] introduces a novel double attentive flow embedding in cost volume. RMS-FLowNet [2] integrates random sampling to efficiently process large-scale scenes instead of FPS. Inspired by BERT [6], Bi-PointFlow [4] applies bidirectional flow embedding to produce cost volume using the sequence information. Res3DSF [34] presents a novel context-aware set convolution layer to enhance the detection of recurrent patterns in 3D space. Nonetheless, these coarse-to-fine methods focus on local flow regression layer-by-layer which lacks global information. To address this issue, Some methods adopt an all-to-all approach. FLOT[28] redefines scene flow prediction as an optimal transmission problem, gauging the transmission cost by evaluating the global cosine similarity of semantic features. FlowStep3D [16] aims to directly compute the initial scene flow by leveraging an unlearnable feature similarity matrix in the global unit. WM3DSF [33] proposes an all-to-all point mixture module with backward reliability validation. Additionally, PT-Flow [9] and PV-RAFT equip the point-voxel branches to the flow embedding to enlarge the receptive field.

However, these methods solely consider the individual point cloud semantic to make a hard flow embedding, which lacks fusion of the global semantic context of another embedded frame. We propose the GF module to fuse the semantic features of two frames and perceive the global correlation in each other's context, which is essentially in the global flow estimation of long-range and complex geometric situations such as occluded[15] and repetitive[34] pattern.

3. Methodology

3.1. Problem Definition

The scene flow task aims to estimate point-wise 3D motion information between two consecutive point cloud frames.

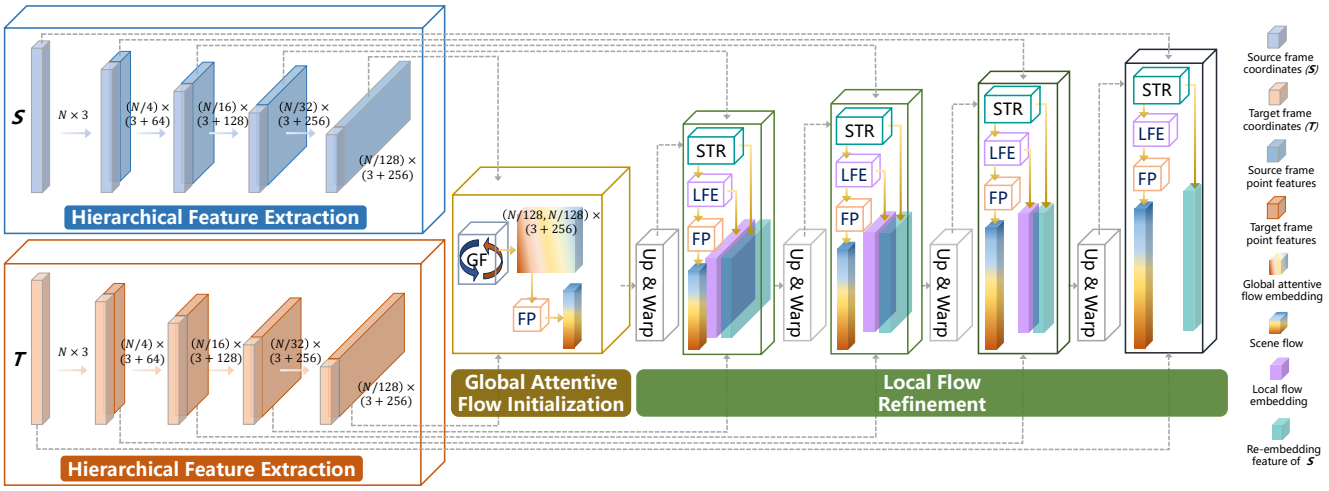


Figure 2. Illustration of the proposed network. Firstly, semantic features are hierarchically extracted and sent to GF to achieve global embedding between the two point clouds at the highest level. Then, the Flow Prediction (FP) module produces the initial scene flow. Subsequently, the flow and features are upsampled level by level, and the upsampled flow is accumulated onto the source frame by the warping layer. Afterwards, Spatial Temporal Re-embedding (STR) and Local Flow Embedding (LFE) are performed in turn, and FP yields the refined flow at a specific level.

The input includes the source frame $S = \{s_i\}_{i=1}^{N} = \{x_i, f_i\}_{i=1}^{N}$ and target frame $T = \{t_j\}_{j=1}^{M} = \{y_j, g_j\}_{j=1}^{M}$, where x_i, $y_j \in \mathbb{R}^3$ are 3D coordinates of the points, and f_i, $g_j \in \mathbb{R}^d$ represent the feature of the corresponding point at a specific level. It should be noted that N and M may not be equal due to the uneven point density and occlusion. The prediction of the model is the 3D motion vector $SF = \{sf_i \in \mathbb{R}^3\}_{i=1}^{N}$ of each source frame point, representing the non-rigid motion towards the target frame.

3.2. Hierarchical Feature Extraction

The overview of our proposed network is shown in Figure 2. We utilize PointConv [39] as the feature extraction backbone to build a pyramid network. To extract the higher-level semantic feature S_{l+1} of level $(l+1)$, we apply a three-step process to the previous lower-level feature S_l. Farthest Point Sampling (FPS) is first employed to extract N_{l+1} center points from S_l, where $N_{l+1} < N_l$. Next, K-Nearest Neighbor (KNN) is used to group the neighbor points around each center point. Finally, PointConv is utilized to aggregate the local features for each group, resulting in the desired semantic feature S_{l+1}.

3.3. Global Fusion Flow Embedding

The GF module is designed to capture the global relation between consecutive frames during the flow initialization. After performing the multi-level feature extraction, we obtain S^* and T^* at the highest level of the semantic pyramid. Following that, the global fusion flow embedding is constructed from S^* to all points in T^* in both semantic context space and Euclidean space, as shown in Figure 3. The previous algorithms[16, 33] merely utilized the individual and unaligned semantic features of two consecutive point clouds for hard global embedding. However, the flow embedding of a point is generated in response to the semantic context of another frame, necessitating the simultaneous consideration of the fused features in a consistent semantic space between the two frames during embedding. To enhance mutual understanding of semantic context between two frames of point clouds, we first utilize the DCA module to fuse and align semantic context from both frames. This equips each frame with the ability to perceive the global semantic environment of the other frame, leading to a more reliable latent correlation.

Specifically, within the DCA module, we employ a cross-attentive mechanism to merge the semantic context of the highest layers in the feature pyramid, yielding an attentive weight map used for subsequent global aggregation, as illustrated in Figure 3. During the dual cross-attentive fusion phase, the semantic context in the latent feature space is obtained for S^* and T^* through linear networks Q, K, and V. Subsequently, Q(T^*) serves as the Query, while K(S^*) serves as the Key for computing the cross-attention map $A^{S \to T}$. The final fused features from S^* to T^* are calculated via V(S^*).

$$A^{S \to T} = \sigma\left(\frac{\mathrm{Q}(T^*) \cdot \mathrm{K}(S^*)}{\sqrt{d_a}}\right), \tag{1}$$

$$Fusion^{S \to T} = A \cdot \mathrm{V}(S^*), \tag{2}$$

where d_a is the output dimension of linear network K and V, and σ denotes SoftMax. The fusion of context features from T^* to S^* follows a similar procedure.

After acquiring the fused features in the semantic space, the GF module initializes the flow embedding for two frames.

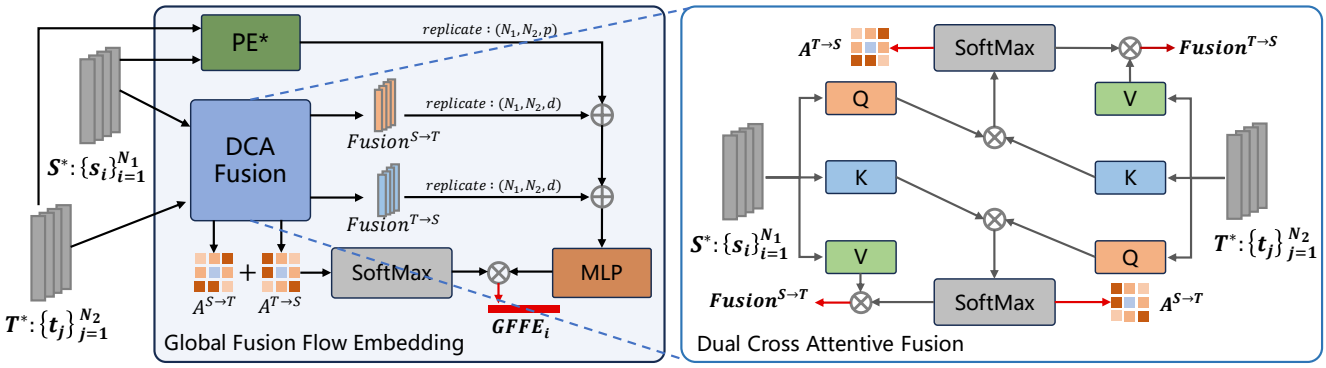


Figure 3. Flowchart of global flow embedding. $\otimes$ and $\oplus$ denote multiplication and concatenation, respectively.

We elucidate the process using a point s_i from S^* as a case to enhance clarity. Firstly, to establish the relative positional association between each point-pair, a position encoder PE^* in Euclidean space is introduced as follows, where η denotes concatenation.

$$PE_{ij} = \eta(x_i, y_j, y_j - x_i), \tag{3}$$

$$PE^*_{ij} = \eta(PE_{ij}, \mathrm{MLP}(PE_{ij})). \tag{4}$$

The external position encoder PE^* instead of internal integration in the DCA module provides explicit position context during global flow embedding. Then we proceed to construct the initial global flow embedding $GFE = \{GFE_{ij}\}$ from both fusion semantic context and Euclidean space. The latent embedding of point-pair s_i and t_j is represented as:

$$GFE_{ij} = \mathrm{MLP}(\eta(Fusion_i^{T\to S}, Fusion_j^{S\to T}, PE^*_{ij})), \tag{5}$$

where η denotes dimension concatenation. After obtaining the dual cross-attentive maps $A^{S\to T}$ and $A^{T\to S}$ within the DCA module, we perform element-wise addition and then pass them through SoftMax to obtain the aggregation weights $W = \{W_{ij}\}$ for the global flow embedding aggregation, which was later proven to be superior to the MaxPooling. Finally, the initial global flow embedding is aggregated by utilizing the aggregation map W, to obtain the global fusion flow embedding from the specific point s_i to all points in the target:

$$GFFE_i = \textstyle\sum_j W_{ij} \cdot GFE_{ij}. \tag{6}$$

Once $GFFE = \{GFFE_i\}$ has been obtained, it is then fed into the flow predictor (described in Section 3.6) to generate the global initial scene flow.

3.4. Warping Layer

We employ distance-inverse interpolation to upsample the coarse sparse scene flow from level $(l+1)$ to obtain the coarse dense scene flow of level l. The obtained coarse dense flow is directly accumulated onto the source frame S_l to generate the warped source frame $WS_l = \{ws_i\}_{i=1}^{N_l} = \{wx_i = x_i + sf_i, f_i\}_{i=1}^{N_l}$, which brings the source and target frames closer and allows the subsequent layers to only consider the estimation of residual flow [4, 9, 16, 33, 38, 40].

3.5. Spatial Temporal Re-embedding

After the warping layer, the spatiotemporal relation between the consecutive frames may change. Specifically, the temporal features of points from the warped source frame to the target change since the position between the two point clouds is closer. Furthermore, dynamic non-rigid objects in the source frame may encounter surface distortion during warping, resulting in different spatial features. Therefore, it is necessary to re-embed spatiotemporal point features before the Local Flow Embedding (LFE), which is implemented in a patch-to-patch manner between the two frames following [40]. Based on this consideration, we re-embed the spatiotemporal features of each point ws_i at level l based on the warped source frame WS_l and the target frame T_l, as depicted in Figure 4.

Temporal Re-embedding First, we locate the K nearest neighbor points group $\mathcal{N}_T(ws_i)$ of point ws_i in T_l. For each target point $t_j \in \mathcal{N}_T(ws_i)$, by employing position encoder as (3), a 9D positional feature PE_{ij} is acquired for this group, representing the positional relation between the two frames after warping. Then, the initial temporal re-embedding feature is derived using the following formula:

$$TRF_{ij} = \mathrm{MLP}(\eta(g_j, f_i, PE_{ij})). \tag{7}$$

Instead of employing the hard aggregation method of MaxPooling, which results in flow bias due to the non-corresponding points between the two frames, we leverage local similarity map $LM_i = \{LM_{ij}\}$ in both feature space and Euclidean space to derive the soft aggregation weights,

$$LM_{ij} = \sigma(\mathrm{MLP}(\eta(TRF_{ij}, \mathrm{MLP}(PE_{ij})))), \tag{8}$$

$$TRF_i^* = \textstyle\sum_j LM_{ij} TRF_{ij}. \tag{9}$$

Spatial Re-embedding Spatial Re-embedding shares the same framework as Temporal Re-embedding, with the only

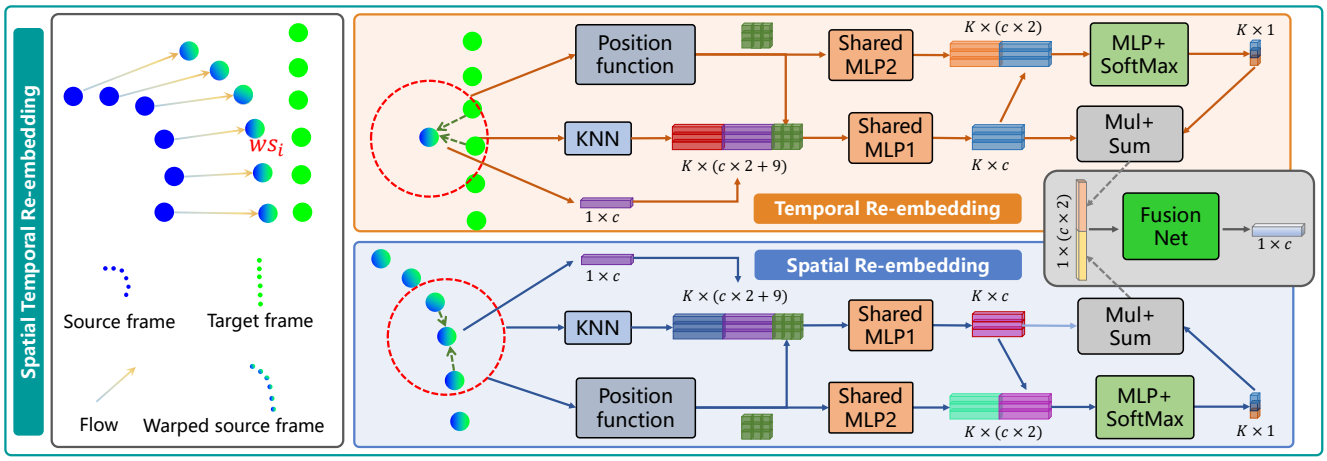


Figure 4. The details of STR module.

distinction being that the embedding object changes to the warped source frame itself. Upon acquiring the temporal re-embedding features $TRF^* = \{TRF_i^*\}$ and spatial re-embedding features $SRF^* = \{SRF_i^*\}$ of each point in the warped source frame of level l, we fuse them by leveraging the Fusion Net module to derive the ultimate comprehensive features

$$STRF_i = \text{MLP}(\eta(TRF_i^*, SRF_i^*)), \tag{10}$$

and the warped frame updates to $WS_l = \{wx_i, STRF_i\}_{i=1}^{N_l}$. As shown in Figure 2, the STR module is followed by LFE, which computes the patch-to-patch cost volume of each point ws_i by utilizing the spatiotemporal re-embedding features.

3.6. Flow Prediction

This module is constructed by combining PointConv, MLP, and a Fully Connected (FC) layer. For each point s_i in the source frame, its local flow embedding feature, along with the warped coordinates and $STRF_i$ are input into the module. PointConv is first employed to incorporate the local information of each point, followed by non-linear transformation in the MLP layer. The final output is the scene flow sf_i, regressed through the FC layer.

4. Training Losses

4.1. Hierarchical Supervised Loss

A supervised loss is directly hooked to the GT of scene flow, and we leverage multi-level loss functions as supervision to optimize the model across various pyramid levels. The GT of scene flow at level l is represented as $\tilde{SF}_l = \{\tilde{sf}_i^l\}_{i=1}^{N_l}$ and the predicted flow is $SF_l = \{sf_i^l\}_{i=1}^{N_l}$. The multi-level supervised loss is as follows:

$$\mathcal{L}_{sup} = \sum_{l=1}^{5} \frac{\delta_l}{N_l} \sum_{i=1}^{N_l} \|\tilde{sf}_i^l - sf_i^l\|_2, \tag{11}$$

where δ is the penalty weight, with $\delta_1 = 0.02, \delta_2 = 0.04, \delta_3 = 0.08, \delta_4 = 0.16$, and $\delta_5 = 0.32$.

4.2. Domain Adaptive Losses

Local Flow Consistency (LFC) Loss Dynamic objects in real-world scenes may not exhibit absolute rigid or regular motion. Instead, they typically undergo local rigid motion, which is manifested through the consistency of local flow. The degree of predicted flow difference between each point s_i and its KNN+Radius points group $\mathcal{N}_S^R(s_i)$ in the source frame at the full resolution level ($N_1 = 8192$) is defined as the LFC loss, where point $p \in \mathcal{N}_S^R(s_i)$ denotes $p \in \mathcal{N}_S(s_i)$ and the ℓ_2 distance between p and s_i is less than R. The KNN+Radius search strategy effectively mitigates the influence of noise points resulting from occlusion and sparsity in point clouds, as demonstrated in Supple.Sec 8.2. Formally, the LFC loss is represented as follows:

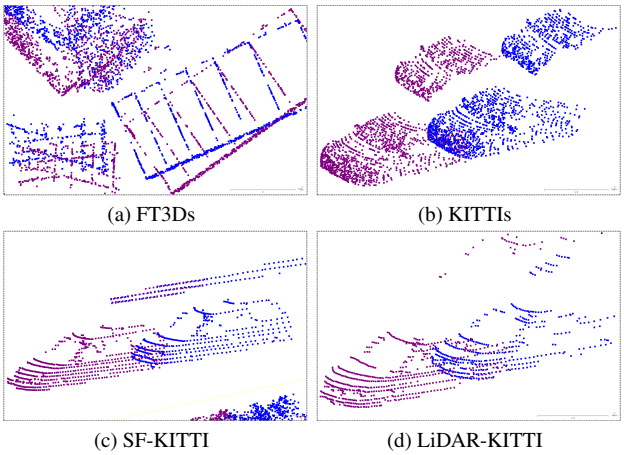
$$\mathcal{L}_{lfc} = \frac{1}{N_1} \sum_{i=1}^{N_1} \frac{1}{|\mathcal{N}_S^R(s_i)|} \sum_{s_j \in \mathcal{N}_S^R(s_i)} \|sf_i - sf_j\|_2, \tag{12}$$

where $|\cdot|$ is the number of points in a group.

Cross-frame Feature Similarity (CFS) Loss The semantic features of the points in the warped source frame are similar to those in the surrounding target frame, as they should be in a dynamic registered state. Specifically, we accumulate the GT scene flow $\tilde{sf}$ directly onto the source frame at the full resolution level, as described by $\tilde{ws}_i = \{x_i + \tilde{sf}_i, STRF_i\}$. Next, we utilize cosine similarity to compute the similarity between $\tilde{ws}_i$ and $t_j \in \mathcal{N}_T^R(\tilde{ws}_i)$ in the target frame:

$$CS(\tilde{ws}_i, t_j) = \frac{STRF_i \odot g_j}{\|STRF_i\|_2 \|g_j\|_2}. \tag{13}$$

We utilize the features of source frame points derived from the last layer of the STR module (the rightmost re-embedding feature in Figure 2) as input instead of the initially extracted



(a) FT3Ds (b) KITTIs

(c) SF-KITTI (d) LiDAR-KITTI

Figure 5. Comparisons of scene flow datasets, including (a) synthetic stereo, (b) real-world stereo, and (c)(d) real-world LiDAR-scanned. Blue and purple denote the source and target frames, respectively.

features. Lastly, we establish a similarity threshold TH and employ function F to penalize points that exhibit a similarity lower than TH:

$$\mathcal{L}_{cfs} = \frac{1}{N_1}\sum_{i=1}^{N_1}\sum_{t_j \in \mathcal{N}_T^R(\tilde{ws}_i)} \frac{F(CS(\tilde{ws}_i, t_j) - TH)}{|\mathcal{N}_T^R(\tilde{ws}_i)|}, \quad (14)$$

where $F(x) = -x$ if $x < 0$ and 0 otherwise, and g_j is updated by the Temporal Re-embedding module of the STR module with the warped source frame for a reliable and precise loss. The final loss of our model is :

$$\mathcal{L}_{all} = \lambda_1 \mathcal{L}_{sup} + \lambda_2 \mathcal{L}_{lfc} + \lambda_3 \mathcal{L}_{cfs}, \quad (15)$$

where $R = 0.05$, $TH = 0.95$, and $\lambda_1 = 0.7$, $\lambda_2 = 0.15$, $\lambda_3 = 0.15$ by default.

5. Experiments

5.1. Datasets and Data Preprocessing

The experiments were performed on four datasets: the synthetic dataset FlyThings3D (FT3D) [23] and three real-world datasets including Stereo-KITTI [24, 25], SF-KITTI [8], and LiDAR-KITTI [10, 12], as shown in Figure 5. These datasets are preprocessed in two ways [13, 21]: FT3Ds and KITTIs remove non-corresponding points between consecutive frames, while FT3Do and KITTIo retain occluded points using mask labels. Further dataset details can be found in Supple.Sec 10.

5.2. Experimental Settings

Implementation Details Our model is implemented with PyTorch 1.9, and both training and testing are conducted on a single NVIDIA RTX3090 GPU. The AdamW optimizer [22] with $\beta_1 = 0.9$ and $\beta_2 = 0.99$ is used for model tuning during the training phase, with an initial learning rate of 0.001 which was decayed by half every 80 epochs. We train our model in an end-to-end manner for 900 epochs (or reached convergence) with batch size 8. The cross-attention is utilized with head $= 8$ and $d_a = 128$. Our model code and weights will be released upon publication. More architectural details are listed in Supple.Sec 7.1.

Evaluation Metrics Following previous methods [4, 8, 12, 28, 33, 40], we employ the same evaluation metrics for fair comparisons, including EPE3D, AS3D, AR3D, Out3D, EPE2D, and Acc2D, which are discussed in detail in Supple.Sec 8.1.

5.3. Results and Analysis

Our method exhibits remarkable generalization ability across various scenarios, encompassing both synthetic and real-world scenes, as well as dense or sparse point clouds. In contrast, some previous methods are tailored to specific datasets.

FT3Ds and KITTIs We compare with recent SOTA methods on the FT3Ds and KITTIs datasets. The quantitative results presented in Table 1 indicate that SSRFlow outperforms the other methods by a large margin, especially in real-world datasets. Specifically, on the FT3Ds dataset, SSRFlow is on par with previous SOTA[20] while achieving a 56% reduction in inference time, as listed in Table 2. Further, our model exhibits exceptional generalization performance on the KITTIs dataset, surpassing the second place by 24% on EPE3D. Qualitative analysis is shown in Figure 6.

FT3Do and KITTIo Similar to the above, we train our model on FT3Do and test on KITTIo without any fine-tuning. The experimental results are listed in Table 3, which reveal the good performance of our model even with occlusion. Specifically, our model achieves 24% improvement over the previous SOTA method [20] on FT3Do. Furthermore, SSRFlow outperforms [20] in terms of EPE3D on the real-world occluded KITTIo dataset. Visualized experimental results are provided in the Figure 7.

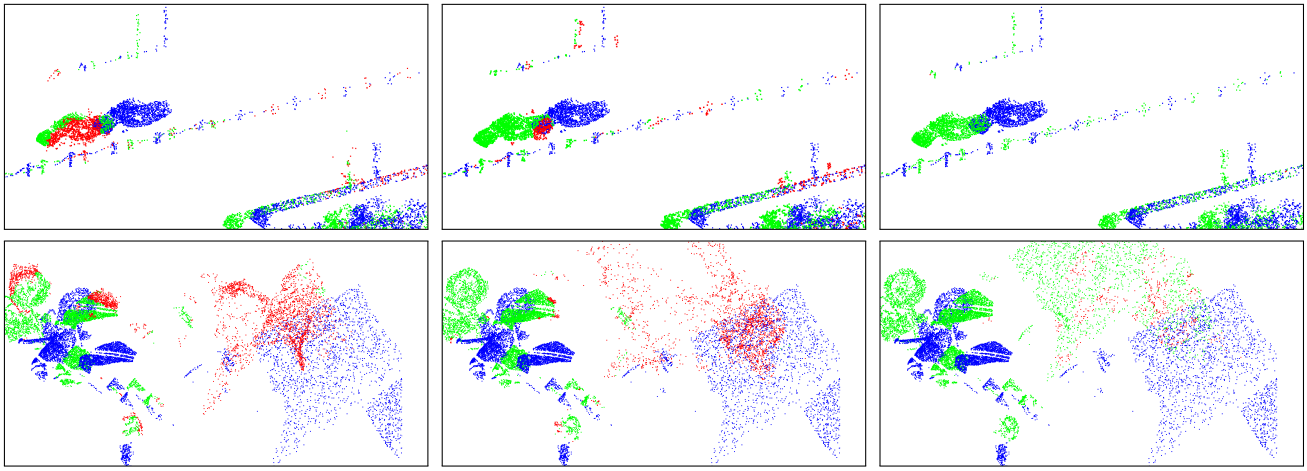
Generalization on LiDAR-KITTI To validate the generalization on real-world LiDAR-scanned datasets, we train our model on FT3Ds and SF-KITTI datasets separately, followed by evaluation on the LiDAR-KITTI dataset. The results are shown in Table 4 and Figure 7. Specifically, SSRFlow reduces EPE3D by 41% and 22% compared to the second place [5] under training on FT3Ds and SF-KITTI datasets, respectively.

5.4. Ablation Study

To investigate the distinct impacts of GF, STR, and DA Losses, a set of ablation experiments are conducted to perform functional analysis. The comprehensive results of the ablation experiments can be found in Table 5, while detailed information is presented in Table 6 and Table 7.

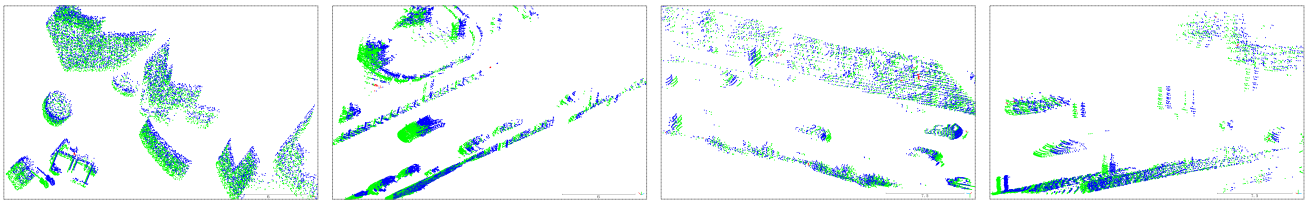
Method	FT3Ds						KITTIs					
	EPE3D↓	AS3D↑	AR3D↑	Out3D↓	EPE2D↓	Acc2D↑	EPE3D↓	AS3D↑	AR3D↑	Out3D↓	EPE2D↓	Acc2D↑
FlowNet3D[21]	0.1136	0.4125	0.7706	0.6016	5.9740	0.5692	0.1767	0.3738	0.6677	0.5271	7.2141	0.5093
PointPWC[40]	0.0588	0.7379	0.9276	0.3424	3.2390	0.7994	0.0694	0.7281	0.8884	0.2648	3.0062	0.7673
FLOT[28]	0.0520	0.7322	0.9276	0.3578	–	–	0.0560	0.7550	0.9080	0.2420	–	–
PV-RAFT[38]*	0.0461	0.8169	0.9574	0.2924	–	–	0.0560	0.8226	0.9372	0.2163	–	–
HCRF[19]	0.0488	0.8337	0.9507	0.2614	2.5652	0.8704	0.0531	0.8631	0.9444	0.1797	2.0700	0.8656
FlowStep3D[16]	0.0455	0.8162	0.9614	0.2165	–	–	0.0546	0.8051	0.9254	0.1492	–	–
SCTN[17]	0.0383	0.8474	0.9681	0.2686	–	–	0.0375	0.8730	0.9592	0.1793	–	–
Bi-PointFlow[4]	0.0282	0.9184	0.9781	0.1436	1.5822	0.9296	0.0307	0.9202	0.9603	0.1414	1.0562	0.9493
WM3DSFNet[33]	0.0281	0.9290	0.9817	0.1458	1.5229	0.9279	0.0309	0.9047	0.9580	0.1612	1.1285	0.9451
RPPformer-Flow[18]	0.0270	0.9211	0.9783	0.1178	–	–	0.0284	0.9220	0.9756	0.1410	–	–
PT-Flow[18]*	0.0304	0.9142	0.9814	0.1735	1.6150	0.9312	0.0224	0.9551	0.9838	0.1186	0.9893	0.9667
MSBRN[5]*	0.0158	0.9733	0.9923	0.0568	0.8335	0.9703	0.0118	0.9713	0.9893	0.0856	0.4435	0.9853
DifFlow[20]*	**0.0114**	**0.9836**	**0.9949**	**0.0350**	**0.6220**	**0.9824**	0.0078	0.9817	0.9924	0.0795	0.2987	0.9932
SSRFlow (Ours)	0.0122	0.9790	0.9942	0.0575	0.7891	0.9821	**0.0059**	**0.9961**	**0.9993**	**0.0762**	**0.2292**	**0.9981**

Table 1. Performance comparisons on the FT3Ds and KITTIs datasets. All models in the table are only trained on FT3Ds and no fine-tuning is applied when tested on KITTIs. The best results for each dataset are marked in bold. * denotes the methods with an inference time exceeding 200ms which are high latency.



PointPWC [40] Bi-PointFlow [4] SSRFlow (Ours)

Figure 6. Visualization comparisons on KITTIs (first row) and FT3Ds (second row). The blue points represent the source frame, and the green points represent the result of warping the source frame using predictions. The red signifies incorrectly predicted warped points whose EPE3D > 0.1m.



(a) FT3Do [23] (b) KITTIo [4] (c) SF-KITTI [8] (d) LiDAR-KITTI [10]

Figure 7. Illustration of results on other datasets of our proposed SSRFlow method. Colors mean the same as Figure 6. More visualization results are exhibited in Supplementary Material, Sec 11

GF In (a) of Table 7, we provide a detailed list of the importance of the DCA Fusion, location of position encoder, aggregation style, and all-to-all point-pair concatenation. Firstly, we exclude the DCA Fusion and the subsequent global attentive aggregation in GF and directly utilize the original semantic feature for global flow embedding. Secondly, we test the internal and external position encoder of cross-attention in DCA Fusion. Additionally, we replace the

Method	Runtime	Method	Runtime
FlowStep3D[16]	292.1ms	PV-RAFT[38]	780.2ms
PT-Flow[9]	376.2ms	MSBRN[5]	225.9ms
DifFlow[20]	231.7ms	SSRFlow (Ours)	101.1ms

Table 2. Runtime of the methods evaluated on KITTIs. Compared with iterative methods, our end-to-end model is efficient.

Dataset	Method	EPE3D↓	AS3D↑	AR3D↑	Out3D↓
FT3Do	WM3DSF[33]	0.0630	0.7911	0.9090	0.2790
	MSBRN[5]	0.0535	0.8364	0.9261	0.2314
	DifFlow[20]	0.0430	0.8910	0.9440	0.1330
	SSRFlow (Ours)	**0.0326**	**0.9152**	**0.9742**	**0.1308**
KITTIo	WM3DSF[33]	0.0730	0.8190	0.8900	0.2610
	MSBRN[5]	0.0448	0.8732	0.9500	0.2085
	DifFlow[20]	0.0310	0.9550	0.9660	0.1080
	SSRFlow (Ours)	**0.0298**	**0.9606**	**0.9740**	**0.1037**

Table 3. Comparisons on the FT3Do and KITTIo datasets. All methods are trained only on FT3Do.

Methods	FT3Ds→LiDAR-KITTI				SF-KITTI→LiDAR-KITTI			
	EPE3D↓	AS3D↑	AR3D↑	Out3D↓	EPE3D↓	AS3D↑	AR3D↑	Out3D↓
FlowNet3D[21]	0.722	0.030	0.122	0.965	0.289	0.107	0.334	0.749
PointPWC[40]	0.390	0.387	0.550	0.653	0.275	0.151	0.405	0.737
FLOT[28]	0.653	0.155	0.313	0.837	0.271	0.133	0.424	0.725
FH-R[8]	0.472	0.369	0.432	0.805	0.156	0.341	0.636	0.612
MSBRN[5]	0.351	0.400	0.592	0.685	0.138	0.433	0.790	0.412
SSRFlow (Ours)	**0.205**	**0.498**	**0.712**	**0.552**	**0.108**	**0.570**	**0.892**	**0.401**

Table 4. Evaluation results on real-world LiDAR-scanned scene flow dataset LiDAR-KITTI.

STR	GF	DA Loss	FT3Ds EPE3D↓	KITTIs EPE3D↓
✔		✔	0.0319	0.0208
	✔	✔	0.0301	0.0221
✔	✔		0.0183	0.0124
✔	✔	✔	**0.0122**	**0.0059**

Table 5. Ablation studies of distinct modules. All module combinations are trained on FT3Ds.

$\mathcal{L}_{cfs}$	$\mathcal{L}_{lfc}$	KNN	Radius	FT3Ds EPE3D↓	KITTIs EPE3D↓
✔		✔	✔	0.0171	0.0109
	✔	✔	✔	0.0169	0.0101
✔	✔	✔		0.0136	0.0082
✔	✔	✔	✔	**0.0122**	**0.0059**

Table 6. Detailed ablations of the DA Losses. KNN and Radius signify different neighborhood search ways.

attentive weighted aggregation with MaxPooling. Finally, we substitute the all-to-all match method with KNN. After removing the DCA Fusion, the model experienced a substantial decline in accuracy, primarily due to its capability

Method	EPE3D↓
Ours (full equip)	**0.0122**
(a) Global Fusion Flow Embedding	
w/o DCA Fusion	0.0259
r/w attentive weight → MaxPooling	0.0208
w/ internal position encoder	0.0162
r/w all-to-all → KNN	0.0203
(b) Spatial Temporal Re-embedding	
w/o Spatial Re-embedding	0.0171
w/o Temporal Re-embedding	0.0203
r/w Fusion net → element-wise addition	0.0159

Table 7. Detailed ablations on FT3Ds, where r/w A→B denotes replace A with B.

to fuse point features with another frame context before embedding. Moreover, the position encoder outside the DCA Fusion provides additional spatial features that are superior to internal equipment. The KNN method struggles to process long-range distance dependencies.

STR We remove the Spatial and Temporal Re-embedding sub-modules separately to consider their contribution to the STR module. The detailed results are listed in (b) of Table 7. It is observed that the Spatial Re-embedding sub-module has brought greater performance improvement, which is in line with common sense, as the relation between the two frames has been taken into account in the subsequent cost volume calculations.

DA Losses We conduct a series of ablation experiments exploring the effectiveness of the LFC loss and the CFS loss, as well as neighborhood search strategies. The results are listed in Table 6. Detailed analysis of the hyper-parameters is in Supple.Sec 8.3.

Transfer to Other Models To evaluate the effectiveness of the proposed GF and STR modules, we conduct experiments by integrating them directly into PointPWC [40], FlowNet3D [21], Bi-PointFlow[4] and WM3DSF[33]. Following the original training strategy as described in the respective papers, the results are listed in Supple.Table 9. Both modules improve network performance.

6. Conclusion

We propose the SSRFlow network to accurately and robustly estimate scene flow. SSRFlow conducts global semantic feature fusion to effectively align the semantic space of both frames and performs attentive flow embedding in both Euclidean and context spaces. Additionally, it effectively re-embeds deformed spatiotemporal features within local refinement. The DA Losses enhance the generalization ability of SSRFlow on various pattern datasets. Experiments show that our method achieves SOTA performance on multiple distinct datasets and we also discuss our limitations in the Supplementary Material.

References

[1] Tali Basha, Yael Moses, and Nahum Kiryati. Multi-view scene flow estimation: A view centered variational approach. *International journal of computer vision*, 101(1):6–21, 2013. 1

[2] Ramy Battrawy, René Schuster, Mohammad-Ali Nikouei Mahani, and Didier Stricker. RMS-FlowNet: Efficient and robust multi-scale scene flow estimation for large-scale point clouds. In *Proceedings of International Conference on Robotics and Automation*, pages 883–889, 2022. 2

[3] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. ShapeNet: An information-rich 3D model repository. *arXiv preprint arXiv:1512.03012*, 2015. 4

[4] Wencan Cheng and Jong Hwan Ko. Bi-PointFlowNet: Bidirectional learning for point cloud based scene flow estimation. In *Proceedings of European Conference on Computer Vision*, pages 108–124, 2022. 1, 2, 4, 6, 7, 8

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

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 2

[7] Fangqiang Ding, Zhen Luo, Peijun Zhao, and Chris Xiaoxuan Lu. milliFlow: Scene flow estimation on mmwave radar point cloud for human motion sensing. *arXiv preprint arXiv:2306.17010*, 2023. 1

[8] Lihe Ding, Shaocong Dong, Tingfa Xu, Xinli Xu, Jie Wang, and Jianan Li. FH-Net: A fast hierarchical network for scene flow estimation on real-world point clouds. In *Proceedings of European Conference on Computer Vision*, pages 213–229, 2022. 6, 7, 8, 4

[9] Jingyun Fu, Zhiyu Xiang, Chengyu Qiao, and Tingming Bai. PT-FlowNet: Scene flow estimation on point clouds with point transformer. *IEEE Robotics and Automation Letters*, 8 (5):2566–2573, 2023. 2, 4, 8

[10] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012. 6, 7, 4

[11] Mozhdeh Gheini, Xiang Ren, and Jonathan May. Cross-attention is all you need: Adapting pretrained transformers for machine translation. *arXiv preprint arXiv:2104.08771*, 2021. 1

[12] Zan Gojcic, Or Litany, Andreas Wieser, Leonidas J Guibas, and Tolga Birdal. Weakly supervised learning of rigid 3D scene flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5692–5703, 2021. 6, 4

[13] Xiuye Gu, Yijie Wang, Chongruo Wu, Yong Jae Lee, and Panqu Wang. HPLFlowNet: Hierarchical permutohedral lattice FlowNet for scene flow estimation on large-scale point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3254–3263, 2019. 2, 6, 4

[14] Frédéric Huguet and Frédéric Devernay. A variational method for scene flow estimation from stereo sequences. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1–7, 2007. 1

[15] Shihao Jiang, Dylan Campbell, Yao Lu, Hongdong Li, and Richard Hartley. Learning to estimate hidden motions with global motion aggregation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9772–9781, 2021. 2

[16] Yair Kittenplon, Yonina C Eldar, and Dan Raviv. FlowStep3D: Model unrolling for self-supervised scene flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4114–4123, 2021. 1, 2, 3, 4, 7, 8

[17] Bing Li, Cheng Zheng, Silvio Giancola, and Bernard Ghanem. SCTN: Sparse convolution-transformer network for scene flow estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1254–1262, 2022. 2, 7

[18] Hanlin Li, Guanting Dong, Yueyi Zhang, Xiaoyan Sun, and Zhiwei Xiong. RPPformer-Flow: Relative position guided point transformer for scene flow estimation. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 4867–4876, 2022. 1, 7

[19] 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*, pages 364–373, 2021. 7

[20] Jiuming Liu, Guangming Wang, Weicai Ye, Chaokang Jiang, Jinru Han, Zhe Liu, Guofeng Zhang, Dalong Du, and Hesheng Wang. Difflow3d: Toward robust uncertainty-aware scene flow estimation with iterative diffusion-based refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15109–15119, 2024. 6, 7, 8

[21] Xingyu Liu, Charles R Qi, and Leonidas J Guibas. FlowNet3D: Learning scene flow in 3D point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 529–537, 2019. 1, 2, 6, 7, 8, 4

[22] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[23] 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/CVF Conference on Computer Vision and Pattern Recognition*, pages 4040–4048, 2016. 2, 6, 7, 4

[24] 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. 2, 6, 4

[25] Moritz Menze, Christian Heipke, and Andreas Geiger. Object

scene flow. *ISPRS Journal of Photogrammetry and Remote Sensing*, 140:60–76, 2018. 2, 6, 4

[26] Nitish Mital, Ezgi Özyilkan, Ali Garjani, and Deniz Gündüz. Neural distributed image compression with cross-attention feature alignment. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2498–2507, 2023. 1

[27] Bojun Ouyang and Dan Raviv. Occlusion guided self-supervised scene flow estimation on 3D point clouds. In *International Conference on 3D Vision*, pages 782–791, 2021. 3

[28] 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*, pages 527–544, 2020. 1, 2, 6, 7, 8

[29] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 2

[30] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8934–8943, 2018. 1, 2

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

[32] Guangming Wang, Xinrui Wu, Zhe Liu, and Hesheng Wang. Hierarchical attention learning of scene flow in 3D point clouds. *IEEE Transactions on Image Processing*, 30:5168–5181, 2021. 1, 2

[33] Guangming Wang, Yunzhe Hu, Zhe Liu, Yiyang Zhou, Masayoshi Tomizuka, Wei Zhan, and Hesheng Wang. What matters for 3D scene flow network. In *Proceedings of European Conference on Computer Vision*, pages 38–55, 2022. 1, 2, 3, 4, 6, 7, 8

[34] Guangming Wang, Yunzhe Hu, Xinrui Wu, and Hesheng Wang. Residual 3-D scene flow learning with context-aware feature extraction. *IEEE Transactions on Instrumentation and Measurement*, 71:1–9, 2022. 1, 2

[35] Haiyan Wang, Jiahao Pang, Muhammad A Lodhi, Yingli Tian, and Dong Tian. FESTA: Flow estimation via spatial-temporal attention for scene point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14173–14182, 2021. 2

[36] Zirui Wang, Shuda Li, Henry Howard-Jenkins, Victor Prisacariu, and Min Chen. FlowNet3D++: Geometric losses for deep scene flow estimation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 91–98, 2020. 2

[37] Andreas Wedel, Clemens Rabe, Tobi Vaudrey, Thomas Brox, Uwe Franke, and Daniel Cremers. Efficient dense scene flow from sparse or dense stereo data. In *Proceedings of European Conference on Computer Vision*, pages 739–751, 2008. 1

[38] 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*, pages 6954–6963, 2021. 4, 7, 8

[39] Wenxuan Wu, Zhongang Qi, and Li Fuxin. PointConv: Deep convolutional networks on 3D point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9621–9630, 2019. 3

[40] 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*, pages 88–107, 2020. 1, 2, 4, 6, 7, 8

[41] Chuan Xu, Qi Zhang, Liye Mei, Xiufeng Chang, Zhaoyi Ye, Junjian Wang, Lang Ye, and Wei Yang. Cross-attention-guided feature alignment network for road crack detection. *ISPRS International Journal of Geo-Information*, 12(9):382, 2023. 1

[42] Yanding Yang, Kun Jiang, Diange Yang, Yanqin Jiang, and Xiaowei Lu. Temporal point cloud fusion with scene flow for robust 3D object tracking. *IEEE Signal Processing Letters*, 29:1579–1583, 2022. 1

[43] Yushan Zhang, Johan Edstedt, Bastian Wandt, Per-Erik Forssén, Maria Magnusson, and Michael Felsberg. Gmsf: Global matching scene flow. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[44] Junhao Zhao, Weijie Huang, Hai Wu, Chenglu Wen, Bo Yang, Yulan Guo, and Cheng Wang. Semanticflow: Semantic segmentation of sequential lidar point clouds from sparse frame annotations. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–11, 2023. 1

[45] Changqing Zhou, Zhipeng Luo, Yueru Luo, Tianrui Liu, Liang Pan, Zhongang Cai, Haiyu Zhao, and Shijian Lu. Pttr: Relational 3D point cloud object tracking with transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8531–8540, 2022. 1

[46] Songshang Zou, Hao Chen, Hui Feng, Guangyi Xiao, Zhen Qin, and Weiwei Cai. Traffic flow video image recognition and analysis based on multi-target tracking algorithm and deep learning. *IEEE Transactions on Intelligent Transportation Systems*, 2022. 1