Pattern Recognition Letters 191 (2025) 37–43

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

FlowCraft: Unveiling adversarial robustness of LiDAR scene flow estimation

K.T. Yasas Mahima [a,*], Asanka G. Perera [b], Sreenatha Anavatti [a], Matt Garratt [a]

[a] School of Engineering and Technology, University of New South Wales, Canberra, ACT, Australia
[b] School of Engineering, University of Southern Queensland, Brisbane, QLD, Australia

ARTICLE INFO

Editor: Jiwen Lu

Keywords:
Adversarial attacks
LiDAR
Scene flow estimation
Autonomous vehicles

ABSTRACT

With the arrival of deep learning and advanced sensor technologies, the autonomous vehicle domain has gained increased research interest. In particular, deep learning networks developed based on 3D LiDAR sensing data for perception and planning in autonomous vehicles demonstrate remarkable performance. However, recent research reveals vulnerabilities in LiDAR-based perception tasks, such as 3D object detection and segmentation, to intentionally crafted adversarial perturbations. Yet, the adversarial robustness of LiDAR-based regression tasks like scene flow estimation, remains largely unexplored. Therefore, this study introduces a novel point perturbation attack named FlowCraft, based on two loss functions, along with a critical analysis of selecting the adversarial objective against scene flow estimation. In particular, evaluations are conducted on trainable, runtime optimization, supervised, and self-supervised scene flow estimation methods using the Argoverse 2 and Waymo datasets in both black-box and white-box settings. Experimental results on the Argoverse 2 benchmark dataset and the DeFlow network show that FlowCraft achieves a relative endpoint error increment of 2.9, while demonstrating a higher endpoint error increase of 5.5 per unit change in Chamfer Distance compared to PGD and CosPGD attacks. Furthermore, our results demonstrate that the performance of point perturbation attacks against runtime optimization methods involves a trade-off between their success rate and overall imperceptibility.

1. Introduction

Deep learning (DL) technologies demonstrate remarkable performance in complex tasks within the computer vision domain. As a result, research on safety-critical applications such as developing autonomous vehicles (AVs) heavily employs DL technologies [1]. With the advent of 3D sensing technologies such as LiDAR, DL networks are used to perform perception and planning tasks, including 3D object detection, object tracking, segmentation, motion prediction, trajectory prediction, and scene flow estimation [2–4].

Despite DL networks demonstrating state-of-the-art performance, their susceptibility to intentionally crafted perturbations, known as adversarial attacks, is a significant problem. This was initially demonstrated in DL networks based on images and later proved successful in altering DL networks developed for AVs based on LiDAR point clouds for perception tasks [3,5]. However, state-of-the-art studies on adversarial attacks against AVs have primarily focused on single-frame perception tasks, with less research attention given to regression tasks which involve multiple frames, such as flow prediction. Specifically, the adversarial robustness of LiDAR-based flow prediction, known as scene flow estimation, remains an unexplored topic. Generally, developing attacks against regression tasks is challenging due to the continuous nature of the values, and it might require large perturbations to significantly alter the predictions [6,7]. Further, unlike crafting perturbations (adding noise) on images, crafting perturbations (shifting points) on LiDAR point clouds highly affects the imperceptibility of the attack, as perturbations can alter the geometric shapes of the objects.

In this study, we first explore potential attack objectives for scene flow estimation and critical aspects of selecting the point cloud to which the perturbation is added. Then, we propose and develop a novel untargeted attack named FlowCraft, optimizing two loss functions: adversarial objective loss and imperceptibility loss. Our experiments, conducted on the Argoverse 2 [8] and Waymo [9] datasets, reveal that the proposed FlowCraft attack significantly alters scene flow predictions and demonstrates state-of-the-art performance compared to previously introduced l_∞ norm-bounded attacks. The main contributions of this study are as follows:

1. We demonstrate that when adding perturbations to the first point cloud in a pair used for scene flow estimation, it is not

* Corresponding author.
E-mail addresses: yasas.mahima@unsw.edu.au (K.T.Y. Mahima), asanka.perera@unisq.edu.au (A.G. Perera), s.anavatti@unsw.edu.au (S. Anavatti), m.garratt@unsw.edu.au (M. Garratt).

https://doi.org/10.1016/j.patrec.2025.02.029
Received 6 August 2024; Received in revised form 4 February 2025; Accepted 23 February 2025
Available online 4 March 2025

fair to calculate the error between predicted and ground truth flows.
2. Analyze the suitability of adversarial objectives commonly used in attacks against image-based flow estimation (Optic Flow) and reveal their direct inapplicability to alter scene flow estimation methods through point perturbations.
3. Introduce a novel adversarial attack method specifically for LiDAR scene flow estimation, named FlowCraft, optimized with dual loss functions.
4. Evaluation of FlowCraft against trainable, runtime optimization, supervised, and self-supervised scene flow estimation methods using the Argoverse 2 and Waymo datasets demonstrates its effectiveness in terms of attack success rate and imperceptibility under both white-box and black-box settings, compared to attack techniques targeting optical flow networks.

The rest of the article is organized as follows: Section 2 reviews the related works. Section 3 presents the problem formulation and the method for selecting the point cloud to add perturbations. Section 4 discusses the methodology of FlowCraft and the selection of the attack objective. Section 5 details the experimental setup of the study. Section 6 presents the evaluation and benchmarking results of FlowCraft under white-box settings. Section 7 examines the black-box transferability of FlowCraft against supervised, self-supervised, trainable, and runtime-optimization-based scene flow estimation methods. Finally, Section 8 summarizes the research findings and concludes the paper.

2. Related works

2.1. Scene flow estimation

Scene flow estimation involves estimating 3D motion using LiDAR point clouds, whereas optical flow estimation focuses on motion estimation using 2D images. Recent scene flow estimation techniques can be categorized into two main approaches based on their algorithmic approach: (1) trainable methods [10–12] which train a feed-forward network with a human annotated or pseudo-labeled dataset, and (2) runtime optimization methods [13–15] which predict the flow using learning-free approaches or by utilizing Multi-Layer Perceptrons (MLPs) based flow refinement techniques without learned weights. While popular scene flow estimation benchmark datasets such as Argoverse 2 [8] and the Waymo scene flow estimation dataset [9,11] have been recently introduced, collecting and human-annotating a scene flow estimation dataset remains challenging and time-consuming. As a result, there has been growing interest in developing self-supervised scene flow estimation techniques, such as training the network using pseudo-labels generated via a 'teacher' method and data exploration-based approaches [12–15].

2.2. Adversarial attacks against flow estimation

Research into adversarial attacks targeting AVs is undergoing significant development. This encompasses attacks against both 2D and 3D perception tasks, such as object detection, segmentation, and tracking. Specifically, when it comes to adversarial attacks against LiDAR-based 3D perception, threat models such as shifting LiDAR points (Perturbation Attack) [16], injecting LiDAR points [17], removing/filtering out LiDAR points [18], or adversarially optimized objects [19] have been used.

However, a significant gap remains in research on adversarial attacks targeting 3D LiDAR-based perception and regression tasks that incorporate multiple frames or temporal information, such as scene flow estimation and motion prediction. Our study aims to address this gap by focusing on LiDAR scene flow estimation, a dense regression task involving at least two LiDAR frames. To the best of our knowledge, the study in [20] is the first to investigate adversarial attacks against LiDAR scene flow estimation. However, in their experiments, perturbation is added without considering the attack's imperceptibility, which is a crucial aspect in designing adversarial attacks against LiDAR point clouds. In contrast, study [21] proposed a LiDAR scene flow estimation-based approach to identify adversarial point injection-based fake object attacks.

Due to the limited literature, we subsequently review adversarial attacks introduced for image-based optic flow networks, which closely resemble LiDAR scene flow estimation.

Ranjan et al. [22] first developed an adversarial patch-based attack against optical flow networks, placing it in both frames to cause incorrect flow predictions. Schrodi et al. [23] developed a global perturbation attack to make the network predict a given target flow from the same or a different domain, based on the I-FGSM [24] attack, which was originally introduced for image classification tasks. Schmalfuss et al. [7] proposed a gradient optimization-based attack method inspired by the C&W attack [25]. This technique is capable of generating disjoint or joint perturbations for the input frame pair. Additionally, instead of limiting the perturbation to individual frames, they extended their attack to create universal perturbations.

To adapt the projected gradient descent (PGD) [26] attack for dense perception tasks, Agnihotri et al. [27] proposed an approach to scale the pixel-wise adversarial loss using cosine similarity between ground truth and predicted labels, thereby giving more emphasis to pixels where the predictions are close to the ground truth under untargeted settings. Koren et al. [28] emphasized the importance of altering only the flows of critical target objects while leaving the rest of the predictions unaffected. They introduced a masking-based perturbation attack, along with a regularization term to encourage off-target consistency.

The aforementioned attacks against optical flow mainly focused on adding noise perturbation to the input. However, since this is not realistic in real-world instances, Schmalfuss et al. [29] proposed an adversarial weather effects-based attack, using a novel differential weather particle rendering method to generate elements such as snowflakes, rain streaks, or fog clouds, together with an adversarial optimization of the particle parameters (e.g., color) to alter flow predictions.

The attack objectives in the above studies can be classified into four main categories: (1) altering the flow in an untargeted manner; (2) making the network predict the reverse of the initial flow, defined as $\mathbf{F}_t = -\mathbf{F}_i$, where $\mathbf{F}_t$ is the targeted flow and $\mathbf{F}_i$ is the initial flow; (3) forcing the network to predict zero flow, defined as $\mathbf{F}_t = 0$; and (4) making the network predict any other arbitrary target flow. In optic flow noise perturbation attacks, directly optimizing reverse flow or zero flow objectives is feasible since there is no coordinate change of the image pixels. However, in LiDAR scene flow estimation, this remains challenging due to the shifting of the coordinates of the LiDAR points during perturbation. We will explore this further in Section 4.

3. Problem formulation

The main task of scene flow estimation is to utilize two consecutive point clouds, P_t and P_{t+1} at times t and $t+1$ without having a strict one-to-one mapping between points, along with ego-motion represented as the transformation matrix $T_{t,t+1}$, to estimate the motion vector for each point as $\mathbf{F}_{t,t+1}(p) = (x, y, z)^T$, where $p \in P_t$. Notably the flow between two point clouds is decomposed as $\mathbf{F}_{t,t+1} = \mathbf{F}_{ego} + \Delta\mathbf{F}$, where $\mathbf{F}_{ego}$ is the ego motion and $\Delta\mathbf{F}$ is the output of the scene flow estimation method. Typically the training of a scene flow estimation network involves minimizing the endpoint error (Eq. (1)) between ground truth and predicted flows.

$$\mathrm{EPE} = \frac{1}{\|P_t\|} \sum_{p \in P_t} \|\mathbf{F}_{\mathrm{pred}}(p) - \mathbf{F}_{\mathrm{gt}}(p)\|_2. \tag{1}$$

In this study, we consider the problem of altering predictions of scene flow estimation methods by shifting the coordinates of points by

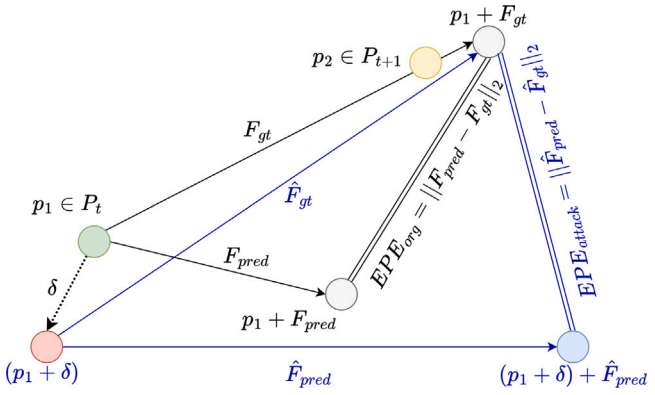

Fig. 1. Scene flow estimation under point perturbation attack on P_t. The attack performance should be evaluated with respect to the shifted points ground truth flow $\hat{\mathbf{F}}_{gt}$. Note: Each circle represents a LiDAR point. Black and blue color lines demonstrate flows and EPE before and after the attack.

perturbation extent δ in an imperceptible manner, such that $\mathbf{F}_{gt} \neq \hat{\mathbf{F}}$, where $\hat{\mathbf{F}}$ represents the flow after the attack. As stated in the previous section, study [20] is the only work that focuses on this problem. Specifically, in this study, an adversarial perturbation δ is added, and the point coordinates in P_t are shifted. However, in such an attack, comparing the predicted flow against the original flow does not provide meaningful insights into the attack's performance. This is because, for each point in P_t, the EPE is calculated as the distance between the predicted and actual flow vectors, with the points in P_t's position serving as the origin. Moreover, though the points in P_t are shifted, the predicted flow might still align well with the altered ground truth flow between shifted points in P_t and their expected positions in P_{t+1}.

Further explained, as shown in Fig. 1, we argue that if δ is added to P_t, it should evaluate the attack performance with respect to the ground truth flow of the shifted points $\hat{\mathbf{F}}_{\text{gt}}$ instead of original ground truth flow $\mathbf{F}_{\text{gt}}$. To mitigate this problem, we decided to add the adversarial perturbation δ to P_{t+1}, as P_{t+1} provides only correspondence and implied motion information without managing any one-to-one mapping between points in P_t, thus enabling the evaluation of the predicted flow of points in P_t with the original ground truth flow $\mathbf{F}_{\text{gt}}$.

4. Proposed approach: FlowCraft

Accurate scene flow estimation is crucial for identifying the motions of critical movable objects, and altering these flow estimations can significantly harm public safety. Hence, in the FlowCraft attack, we only consider perturbing the point clouds of pre-identified critical object categories (e.g., Vehicles, Pedestrians), leaving all other off-target points unchanged. In the real world, an attacker could potentially identify the points of critical objects using a pre-trained LiDAR segmentation network. Let M_{t+1} be the binary mask that represents the critical object points in P_{t+1}, then FlowCraft intends to synthesize the adversarially corrupted $\hat{P}_{t+1}$ by shifting the coordinates of these critical objects' points via perturbation extent δ, as follows.

$$\hat{P}_{t+1} = M_{t+1} \odot (P_{t+1} + \delta) + (1 - M_{t+1}) \odot P_{t+1}. \tag{2}$$

We defined the generation process of $\hat{P}_{t+1}$ as an iterative optimization process inspired by the C&W attack [25] and guided by two loss functions that reflect the FlowCraft attack objective and control the imperceptibility of the attack.

As mentioned previously, two common attack objectives used in optic flow attacks are making the network predict reverse flow $\hat{\mathbf{F}} = -\mathbf{F}_i$ or predicting zero flow $\hat{\mathbf{F}} = 0$. However, shifting point coordinates to achieve this objective might require a large perturbation and result in low imperceptibility. Moreover, it may require a larger number of iterations. Hence, we design FlowCraft to be an untargeted attack with the objective of maximizing the EPE. In particular, given a pre-trained scene flow estimation network S_θ, two consecutive point clouds P_t, P_{t+1} and ground truth flow $\mathbf{F}_{gt}$, FlowCraft intends to maximize EPE of critical objects as follows:

$$\hat{\mathbf{F}} = S_\theta(P_t, M_{t+1} \odot (P_{t+1} + \delta) + (1 - M_{t+1}) \odot P_{t+1}). \tag{3}$$

$$L_{obj} = -\frac{1}{\|P_t\|} \sum_{p \in P_t} \|\hat{\mathbf{F}}(p) - \mathbf{F}_{\text{gt}}(p)\|_2. \tag{4}$$

Specifically, we developed two variants of the final objective loss: (1) Global objective loss: calculates L_{obj} for all points in P_t to obtain gradients for critical objects in P_{t+1}, and (2) Local objective loss: computes $L_{\text{obj}}[M_t]$ specifically for critical object points in P_t, using the corresponding critical object mask M_t and obtaining gradients for critical objects in P_{t+1}.

The shifting of point coordinates should be done imperceptibly. Given that FlowCraft perturbs critical object point clouds with distinct geometric shapes, using a loss function that captures shape similarities is essential to manage the attack imperceptibility. Hence, we adopt Chamfer Distance (Eq. (5)) as the imperceptibility loss function.

$$L_{\text{cd}}(\hat{P}_{t+1}, P_{t+1}) = \sum_{p \in \hat{P}_{t+1}} \min_{q \in P_{t+1}} \|p - q\|_2^2 + \sum_{q \in P_{t+1}} \min_{p \in \hat{P}_{t+1}} \|q - p\|_2^2. \tag{5}$$

The final process of generating adversarial $\hat{P}_{t+1}$ could be formulated as a norm unbounded optimization problem of the above two loss functions as denoted in Eq. (6)

$$\arg\min_{M \odot \hat{P}_{t+1}} \lambda_1 L_{obj} + \lambda_2 L_{cd}, \tag{6}$$

where λ_1, λ_2 are balancing parameters. The main steps of the FlowCraft attack are presented in Algorithm 1.

Algorithm 1 FlowCraft Attack Steps

Require: Point Clouds: P_t, P_{t+1}, Critical object masks in P_t, P_{t+1} : M_t, M_{t+1}, Scene Flow Network: S_θ, Iterations: N, Ground truth flow: $\mathbf{F}_{gt}$, Optimizer :O, Learning Rate: η

```
P̂_{t+1} ← P_{t+1}
n ← 0
Enable Gradient on P̂_{t+1}
Assign P̂_{t+1} to the O
while n < N do
    F̂ ← S_θ(P_t, P̂_{t+1})
    if Local Loss == True then
        Get λ1 L_obj(F̂, F_gt)[M_t] + λ2 L_cd(P̂_{t+1}, P_{t+1})
    else
        Get λ1 L_obj(F̂, F_gt) + λ2 L_cd(P̂_{t+1}, P_{t+1})
    end if
    Calculate gradients ∇_{P̂_{t+1}}
    ∇_{P̂_{t+1}}[!M_{t+1}] ← 0      ▷ Zero the gradients of non targeted points
    P̂_{t+1} −= η × ∇_{P̂_{t+1}}      ▷ Take gradient step using O
    n+ = 1
end while
Return P̂_{t+1}
```

5. Experimental setup

This section summarizes datasets, scene flow estimation methods, and evaluation metrics used to assess the performance of the FlowCraft attack.

5.1. Scene flow estimation methods and datasets

We evaluate the adversarial robustness across all categories of scene flow estimation techniques. In particular, FlowCraft attack performance is evaluated against trainable, runtime optimization, supervised,

Table 1
Scene flow estimation approaches used in the evaluation.

Method	Trainable	Supervision	C_A	C_W
DeFlow [10]	✓	Full	✓	✗
FastFlow3D [11]	✓	Full	✓	✓
ZeroFlow [12]	✓	Self	✓	✓
ICP Flow + FNN [15]	✓	Self	✓	✓
NSFP [13]	✗	Self	NA	NA
FastNSF [14]	✗	Self	NA	NA

Note: C_A, C_W - Checkpoint availability for the Argoverse 2 dataset and the Waymo dataset, respectively.

and self-supervised scene flow estimation approaches. Details of the selected scene flow estimation approaches are summarized in Table 1.

Notably, FastFlow3D is based on a 2D U-Net style network with Pillar encoding [30], while DeFlow improves upon this by integrating Gated Recurrent Units to refine various point features within the same voxel, along with an improved version of EPE loss to adjust the weights based on each point's motion type. On the other hand, NSFP and FastNSF employ a runtime optimization-based data exploration approach grounded on non-trained MLPs to extract flows, which fall under the self-supervised paradigm. NSFP uses the Chamfer distance loss as its main objective function to extract flows, while FastNSF introduces a novel, correspondence-free distance transform loss function to overcome NSFP's slow execution time. Further, ZeroFlow and ICP Flow + FNN use a network identical to FastFlow3D as the main network, while generating pseudo-labels for training using NSFP and an Iterative Closest Point (ICP) algorithm-based approach, respectively. We specifically experimented with the white-box version of FlowCraft using DeFlow and evaluated the black-box transferability of the attack using other approaches.

We use the evaluation subset[1] extracted from the Argoverse 2 [8] validation dataset, which consists of 23,547 LiDAR samples across 150 scenarios, as our main dataset. We also conduct further experiments using the Waymo [9] dataset, which contains over 39k LiDAR samples across 202 scenes. Both datasets were collected at a LiDAR frequency of 10 Hz. Notably, for the evaluations, we subsampled 4040 LiDAR samples from the Waymo dataset by selecting the first 20 samples from each scene. In both datasets, we employ classes related to vehicles and human subjects (e.g., pedestrians, motorcyclists) as critical objects.

We use a desktop computer equipped with an Nvidia RTX 3090 GPU and a configuration featuring an Nvidia Tesla V100 from a supercomputer cluster for our experiments. We utilize PGD [26] and CosPGD [27] attacks as benchmarks, as they have been utilized in studies on attacks against optic flow. In both techniques, we set the step size to 0.01, the maximum allowable perturbation to 0.5, and the number of iterations to 100. In the FlowCraft attack against the DeFlow network, we optimize it for 100 iterations using the Adam optimizer with a learning rate of 0.01, keeping λ_1 at 1 and λ_2 at 0.5. Moreover, all the attacks are initiated with zero perturbation instead of random initialization.

5.2. Evaluation metrics

The most common metric for evaluating scene flow is the EPE, which measures the $L2$ norm between the predicted and ground truth flow vectors. The Argoverse 2 benchmark uses the 3-Way EPE introduced in [31]. This metric calculates the average EPE error across three classifications: foreground dynamic, foreground static, and background static.

Inspired by this approach, and since FlowCraft focuses on altering the flow predictions of critical objects, we calculate the relative increment of EPE before and after the attacks for all critical object points (EPE_R), as defined in Eq. (7). We refer to this as the attack success rate throughout the rest of the article. Additionally, we calculate the same metric for static (EPE_R^S) and dynamic (EPE_R^D) critical objects. Specifically, as per [10,31], a point is classified as static or dynamic based on Eq. (8).

$$\mathrm{EPE}_R = \frac{\mathrm{EPE}_{\text{attacked}} - \mathrm{EPE}_{\text{original}}}{\mathrm{EPE}_{\text{original}}}. \tag{7}$$

$$\text{Dynamic or Static} = \begin{cases} \text{Dynamic} & \text{if } \|\mathbf{F}_{t,t+1} - \mathbf{F}_{\text{ego}}\|_2 >= 0.05\,\text{m} \\ \text{Static} & \text{otherwise} \end{cases} \tag{8}$$

Using Eq. (8), we calculate the metric R_{IOU}, which represents the ratio of the mean intersection over union (mIoU) of static and dynamic points before and after the attack. Finally, we evaluate the imperceptibility of the attack by calculating the increase in EPE error corresponding to a unit change in a distance measurement L_D, such as the chamfer distance or L2, as shown in Eq. (9). Getting a higher value for this metric implies that the attack can achieve a higher error with less change to the point cloud.

$$I^D_{EPE} = \frac{\mathrm{EPE}_{\text{attacked}} - \mathrm{EPE}_{\text{original}}}{L_D(\hat{P}_{t+1}, P_{t+1})}. \tag{9}$$

Notably, following the Argoverse 2 scene flow evaluation protocol, we calculate the EPE and mIoU for the flows of critical objects predicted by the algorithm, excluding those beyond ±50 m in the x and y directions and ±3 m in the z direction from the origin.

6. Evaluation results

We present EPE_R, EPE_R^D, EPE_R^S, R_{IOU}^D, R_{IOU}^S and $\mathrm{I}^D_{\mathrm{EPE}}$ before and after the attack on critical objects in Argoverse 2 dataset while using global and local objective losses under the white-box setting in Table 2. When considering the EPE, these findings demonstrate that FlowCraft significantly outperforms both PGD and CosPGD attacks. Further, when using the local loss, FlowCraft achieves a higher attack success rate, while both CosPGD and PGD show nearly identical performance under both local and global loss methods.

Furthermore, Table 3 presents the same quantitative evaluation of the attack's performance against the DeFlow network trained on the Argoverse 2 dataset but evaluated on the Waymo dataset. These results further highlight the state-of-the-art performance of the FlowCraft attack and demonstrate its strong generalizability to entirely unseen data distributions compared to the one used for training the network. Further, on both datasets, FlowCraft shows slightly better R_{IOU}^D and R_{IOU}^S values with local loss. Additionally, the performance of CosPGD and PGD attacks is nearly identical across both datasets. This behavior is also observed in the CosPGD attack on optical flow networks under untargeted settings [27].

The I^D_{EPE} metric calculated using the Chamfer distance metric and presented in Tables 2 and 3, illustrates that, compared to the other two attacks, FlowCraft significantly increases the EPE for a unit increment in Chamfer distance, highlighting its imperceptibility. Moreover, when considering both imperceptibility and attack performance, FlowCraft using global loss is more effective than the one using local loss. Fig. 2 illustrates several qualitative results of the FlowCraft attack with local loss on the DeFlow network. Meanwhile, Fig. 3 differentiates the P_{t+1} before and after the attack, highlighting how the attack shifts the points without completely vanishing the geometric shapes of the objects in an imperceptible way.

7. Black box transferability analysis

Utilizing the DeFlow network as a surrogate, we evaluate the cross-method transferability of point clouds modified by FlowCraft with global loss in a black-box manner, against FastFlow3D, ZeroFlow, ICP Flow+FNN, NSFP, and FastNSF. Specifically, we set the number of inference iterations to 1000 for the NSFP and FastNSF methods.

[1] https://argoverse.github.io/user-guide/tasks/3d_scene_flow.html

Table 2
FlowCraft performance comparison on DeFlow [10], using the Arogverse 2 Dataset [8]: Global Loss vs. Local Loss.

Loss Type	Attack	$EPE_R \uparrow$	$EPE_R^D \uparrow$	$EPE_R^S \uparrow$	$R_{IoU}^D \downarrow$	$R_{IoU}^S \downarrow$	$I_{EPE}^D \uparrow$
Global Loss	PGD	2.143	1.7205	2.4757	**0.803**	0.716	5.2779
	CosPGD	2.148	1.7218	2.4824	**0.803**	**0.715**	5.2883
	FlowCraft	**2.462**	**1.9137**	**2.7914**	0.817	0.734	**15.613**
Local Loss	PGD	2.162	1.7264	2.5044	0.802	0.716	5.333
	CosPGD	2.152	1.7194	2.4984	0.803	0.718	5.301
	FlowCraft	**2.909**	**2.2578**	**3.5535**	**0.800**	**0.696**	**10.847**

Table 3
FlowCraft performance comparison on DeFlow [10], using the Waymo Dataset [9]: Global Loss vs. Local Loss.

Loss Type	Attack	$EPE_R \uparrow$	$EPE_R^D \uparrow$	$EPE_R^S \uparrow$	$R_{IoU}^D \downarrow$	$R_{IoU}^S \downarrow$	$I_{EPE}^D \uparrow$
Global Loss	PGD	1.1543	0.66956	2.4523	0.8437	0.573	4.393
	CosPGD	1.1561	0.67597	2.4474	**0.8430**	**0.571**	4.395
	FlowCraft	**1.4298**	**0.81796**	**2.9880**	0.8535	0.593	**16.822**
Local Loss	PGD	1.1546	0.67567	2.4401	0.8402	0.572	4.404
	CosPGD	1.1530	0.67278	2.4490	**0.8396**	0.571	4.403
	FlowCraft	**1.7945**	**1.0648**	**3.7017**	**0.8396**	**0.546**	**11.629**

Table 4
Transferability analysis of FlowCraft with Global Loss on the Argoverse 2 Dataset [8].

Network	Attack	$EPE_R \uparrow$	$EPE_R^D \uparrow$	$EPE_R^S \uparrow$	$R_{IoU}^D \downarrow$	$R_{IoU}^S \downarrow$	$I_{EPE}^D \uparrow$
FastFlow 3D [11]	PGD	0.4419	0.3432	0.9009	0.9151	0.9021	1.8394
	CosPGD	0.4442	0.3428	0.8924	**0.9143**	**0.9013**	1.8476
	FlowCraft	**0.5117**	**0.3802**	**0.9627**	0.9176	0.9038	**5.4833**
ZeroFlow [12]	PGD	0.4550	0.29863	**0.9884**	**0.8727**	**0.8660**	1.8342
	CosPGD	0.4574	0.29952	0.9882	0.8739	0.8672	1.8425
	FlowCraft	**0.5186**	**0.34398**	0.9431	0.8964	0.8888	**5.3825**
ICP Flow + FNN [15]	PGD	2.7695	2.2513	5.4090	0.725	**0.7028**	6.8196
	CosPGD	2.7715	2.2498	5.4120	**0.724**	0.7031	6.8179
	FlowCraft	**3.0467**	**2.3312**	**5.7737**	0.764	0.7480	**19.313**
NSFP [13]	PGD	2.5829	1.0240	5.7952	**0.740**	**0.320**	10.494
	CosPGD	**2.6094**	**1.0288**	**5.8686**	0.742	0.321	**10.592**
	FlowCraft	0.7234	0.3922	1.1236	0.863	0.795	7.5671
FastNSF [14]	PGD	1.6132	0.6241	4.7079	**0.8121**	**0.4720**	9.1815
	CosPGD	**1.6182**	**0.6257**	**4.7295**	0.8123	0.4743	**9.2010**
	FlowCraft	0.3461	0.2080	0.3110	0.8934	0.9211	5.0710

Note: NSFP and FastNSF yield slightly different results each time.

Table 5
Transferability analysis of FlowCraft with Global Loss on the Waymo Dataset [9].

Network	Attack	$EPE_R \uparrow$	$EPE_R^D \uparrow$	$EPE_R^S \uparrow$	$R_{IoU}^D \downarrow$	$R_{IoU}^S \downarrow$	$I_{EPE}^D \uparrow$
FastFlow 3D [11]	PGD	**0.3099**	**0.1890**	**0.9661**	0.9363	**0.9473**	1.2714
	CosPGD	0.3055	0.1883	0.9576	0.9388	0.9487	1.2520
	FlowCraft	0.2845	0.1679	0.9039	**0.9338**	0.9513	**3.6080**
ZeroFlow [12]	PGD	**0.3629**	**0.1827**	**1.3481**	**0.8237**	**0.8336**	1.6458
	CosPGD	0.3531	0.1778	1.3336	0.8280	0.8367	1.5996
	FlowCraft	0.3142	0.1406	1.2514	0.8578	0.8732	**4.4055**
ICP Flow + FNN [15]	PGD	0.6040	**0.4489**	3.7377	**0.5435**	0.9057	4.0395
	CosPGD	0.6027	0.4484	3.7740	0.5497	0.9056	4.0259
	FlowCraft	**0.7200**	0.4488	**8.1584**	0.6313	**0.8831**	**14.887**
FastNSF [14]	PGD	**1.6039**	**0.4944**	**4.6151**	0.8125	**0.6340**	**7.2621**
	CosPGD	1.5655	0.4755	4.5340	**0.8103**	0.6372	7.0807
	FlowCraft	0.2912	0.1492	0.3327	0.8673	0.9363	4.0765

Note: FastNSF yields slightly different results each time.

Table 4 presents a quantitative comparison of FlowCraft's transferability against PGD and CosPGD attacks on the Argoverse 2 dataset, along with the results of the I_{EPE}^D metric using the Chamfer distance. These results indicate that point perturbation attacks, including FlowCraft, exhibit a higher degree of transferability on both trainable and runtime optimization scene flow estimation methods. Specifically, FlowCraft outperforms both PGD and CosPGD attacks when using the trainable seen flow estimation techniques in terms of both attack success rate and imperceptibility.

Table 5 demonstrates the black-box cross-method and cross-domain transferability of the three attacks, where the perturbations are generated using the DeFlow network trained on the Argoverse 2 dataset to

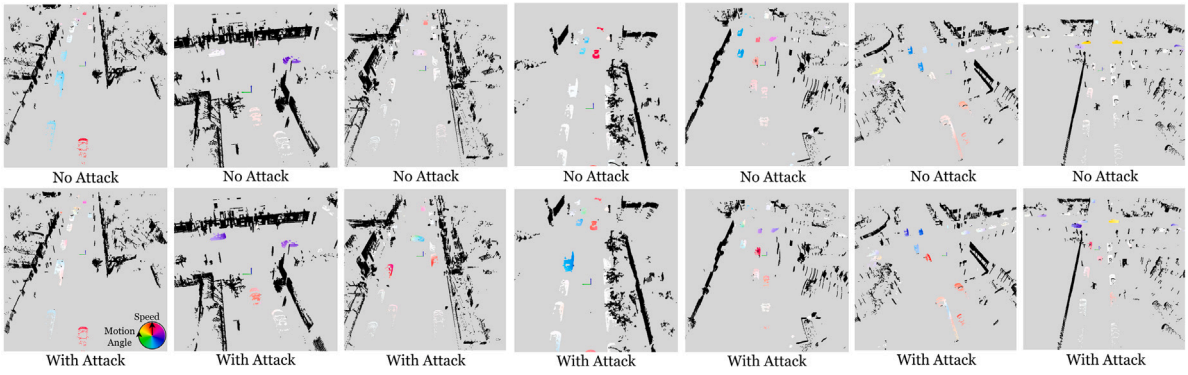

Fig. 2. Qualitative results of flow predictions before and after the attack on DeFlow [10]: Color intensity represents the speed, and the angle of the flow vector is represented by color variation.

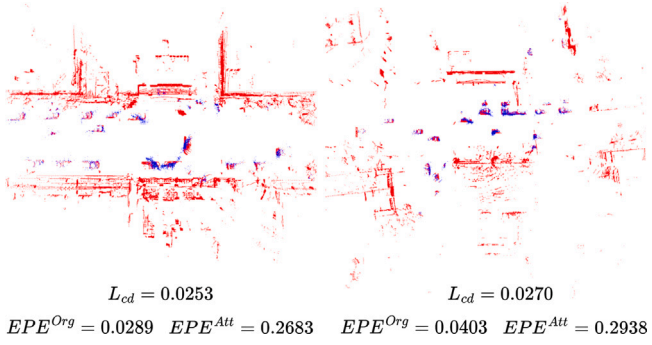

Fig. 3. Visualization of P_{t+1} before and after the attack. Red color: points before the attack. Blue color: points after the attack.

attack scene flow estimation techniques on the Waymo dataset. In this context, under the ZeroFlow and FastFlow3D, the CosPGD and PGD attacks slightly outperform FlowCraft. However, as indicated by the imperceptibility results in the I^D_{EPE} metric, FlowCraft still outperforms these two attacks. Additionally, in both datasets, CosPGD and PGD attacks show relatively better R^D_{IOU} and R^S_{IOU} values than FlowCraft.

Although the ICP Flow + FNN architecture is the same as FastFlow3D and ZeroFlow, it demonstrates comparatively high adversarial vulnerability to the evaluated attacks on both datasets. This might be due to the use of rigid pseudo-labels for scene flows, generated by the ICP algorithm to train the network, which assumes that each moving point cluster (e.g., a vehicle) in the AV domain has a rigid motion.

When considering the two runtime optimization methods, NSFP and FastNSF, both CosPGD and PGD outperform FlowCraft, which exhibit higher Chamfer distance values or lower imperceptibility. This outcome is acceptable, as the objective of the FlowCraft attack is to increase the EPE while decreasing the Chamfer distance. Similarly, NSFP calculates the scene flow using Chamfer distance as a regularization term. Additionally, the original NSFP paper highlighted its vulnerabilities to partial point clouds and occlusions, which indicates the importance of having accurate correspondences and geometric shapes between the two point clouds and their point clusters for precise scene flow estimation. Given these results, it is possible to argue that the success rate of adversarial point perturbation attacks against the runtime optimization techniques, NSFP and FastNSF, is a trade-off between the total attack success rate and its imperceptibility.

8. Conclusions

In this paper, we explore how to perform adversarial attacks against LiDAR scene flow estimation. Specifically, we introduce a point perturbation attack tailored for LiDAR scene flow estimation, considering imperceptibility and analyzing potential attack goals. Given that scene flow estimation involves two consecutive LiDAR point clouds, we first determine which point cloud the perturbation attack will target, ensuring its suitability for accurately evaluating the attack's performance. We then develop the FlowCraft attack, which perturbs the current point cloud as an optimization of two loss functions. Evaluations conducted on trainable scene flow estimation networks using the Argoverse 2 and Waymo datasets reveal that both the white-box and black-box transfer versions of FlowCraft outperform PGD and CosPGD attacks, considering both attack performance and imperceptibility, while also demonstrating strong generalizability to unseen domains during network training. On the other hand, our results on NSFP and FastNSF reveal that the success rate of perturbation attacks against self-supervised runtime optimization methods primarily depends on the lower geometric consistency between the two point clouds.

Future research in this area involves developing physically realizable attacks against scene flow estimation and experimenting with adversarial defense methods to mitigate the highlighted vulnerabilities. Since runtime optimization scene flow estimation techniques demonstrate strong adversarial robustness, we hope that improving these methods will be beneficial and have the potential to be integrated into commercial AVs. Further, investigating concepts such as multi-body rigidity, novel clustering techniques, and multi-sensor fusion methods could be identified as promising research areas for developing robust scene flow estimation techniques. Moreover, universal adversarial perturbations that can be applied to any frame without requiring prior knowledge should also be investigated.

CRediT authorship contribution statement

K.T. Yasas Mahima: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Asanka G. Perera:** Writing – review & editing, Supervision, Methodology, Investigation. **Sreenatha Anavatti:** Writing – review & editing, Supervision, Methodology, Investigation. **Matt Garratt:** Writing – review & editing, Supervision, Methodology, Investigation, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was conducted with the assistance of resources and services from the National Computational Infrastructure (NCI) Australia under the UNSW allocation scheme, which is supported by the Australian Government. The first author would like to acknowledge the University of New South Wales Tuition Fee Scholarship.

Data availability

The data used in this study are publicly available.

References

[1] B.B. Elallid, N. Benamar, A.S. Hafid, T. Rachidi, N. Mrani, A comprehensive survey on the application of deep and reinforcement learning approaches in autonomous driving, J. King Saud University- Comput. Inf. Sci. 34 (9) (2022) 7366–7390.

[2] J. Mao, S. Shi, X. Wang, H. Li, 3D object detection for autonomous driving: A comprehensive survey, Int. J. Comput. Vis. 131 (8) (2023) 1909–1963.

[3] K.T.Y. Mahima, A.G. Perera, S. Anavatti, M. Garratt, Towards Robust 3D Perception for Autonomous Vehicles: A Review of Adversarial Attacks and Countermeasures, IEEE Trans. Intell. Transp. Syst. (2024).

[4] Z. Li, N. Xiang, H. Chen, J. Zhang, X. Yang, Deep learning for scene flow estimation on point clouds: A survey and prospective trends, in: Computer Graphics Forum, vol. 42, (6) Wiley Online Library, 2023, e14795.

[5] Z. Zhu, X. Yang, H. Su, S. Zheng, CamoEnv: Transferable and environment-consistent adversarial camouflage in autonomous driving, Pattern Recognit. Lett. 188 (2025) 95–102.

[6] X. Kong, Z. Ge, Adversarial Attacks on Regression Systems via Gradient Optimization, IEEE Trans. Syst. Man, Cybernetics: Syst. 53 (12) (2023) 7827–7839.

[7] J. Schmalfuss, P. Scholze, A. Bruhn, A perturbation-constrained adversarial attack for evaluating the robustness of optical flow, in: European Conference on Computer Vision, Springer, 2022, pp. 183–200.

[8] B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, B. Pan, R. Kumar, A. Hartnett, J.K. Pontes, D. Ramanan, P. Carr, J. Hays, Argoverse 2: Next Generation Datasets for Self-Driving Perception and Forecasting, in: Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021), 2021.

[9] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. 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, 2020, pp. 2446–2454.

[10] Q. Zhang, Y. Yang, H. Fang, R. Geng, P. Jensfelt, DeFlow: Decoder of scene flow network in autonomous driving, in: 2024 IEEE International Conference on Robotics and Automation, ICRA, 2024, pp. 2105–2111.

[11] P. Jund, C. Sweeney, N. Abdo, Z. Chen, J. Shlens, Scalable scene flow from point clouds in the real world, IEEE Robot. Autom. Lett. 7 (2) (2021) 1589–1596.

[12] K. Vedder, N. Peri, N.E. Chodosh, I. Khatri, E. EATON, D. Jayaraman, Y. Liu, D. Ramanan, J. Hays, ZeroFlow: Scalable Scene Flow via Distillation, in: The Twelfth International Conference on Learning Representations.

[13] X. Li, J. Kaesemodel Pontes, S. Lucey, Neural scene flow prior, Adv. Neural Inf. Process. Syst. 34 (2021) 7838–7851.

[14] X. Li, J. Zheng, F. Ferroni, J.K. Pontes, S. Lucey, Fast Neural Scene Flow, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 9878–9890.

[15] Y. Lin, H. Caesar, ICP-Flow: LiDAR Scene Flow Estimation with ICP, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 15501–15511.

[16] Y. Zhang, J. Hou, Y. Yuan, A comprehensive study of the robustness for LiDAR-based 3D object detectors against adversarial attacks, Int. J. Comput. Vis. (2023) 1–33.

[17] Y. Cao, C. Xiao, B. Cyr, Y. Zhou, W. Park, S. Rampazzi, Q.A. Chen, K. Fu, Z.M. Mao, Adversarial sensor attack on LiDAR-based perception in autonomous driving, in: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, 2019, pp. 2267–2281.

[18] Z. Wang, X. Wang, F. Sohel, M. Bennamoun, Y. Liao, J. Yu, Adversary distillation for one-shot attacks on 3d target tracking, in: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2022, pp. 2453–2749.

[19] J. Tu, M. Ren, S. Manivasagam, M. Liang, B. Yang, R. Du, F. Cheng, R. Urtasun, Physically realizable adversarial examples for LiDAR object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 13716–13725.

[20] H.E. Oskouie, M.-S. Moin, S. Kasaei, Attack on scene flow using point clouds, in: 2024 IEEE 34th International Workshop on Machine Learning for Signal Processing (MLSP), IEEE, 2024, pp. 1–6.

[21] M. Cho, Y. Cao, Z. Zhou, Z.M. Mao, ADoPT: LiDAR Spoofing Attack Detection Based on Point-Level Temporal Consistency, in: 34th British Machine Vision Conference 2023, BMVC 2023, Aberdeen, UK, November 20-24, 2023, BMVA, 2023.

[22] A. Ranjan, J. Janai, A. Geiger, M.J. Black, Attacking optical flow, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 2404–2413.

[23] S. Schrodi, T. Saikia, T. Brox, Towards understanding adversarial robustness of optical flow networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 8916–8924.

[24] A. Kurakin, I. Goodfellow, S. Bengio, Adversarial machine learning at scale, 2016, arXiv preprint arXiv:1611.01236.

[25] N. Carlini, D. Wagner, Towards evaluating the robustness of neural networks, in: 2017 IEEE Symposium on Security and Privacy, IEEE, 2017, pp. 39–57.

[26] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, 2017, arXiv preprint arXiv:1706.06083.

[27] S. Agnihotri, S. Jung, M. Keuper, CosPGD: an efficient white-box adversarial attack for pixel-wise prediction tasks, in: Forty-First International Conference on Machine Learning, 2024.

[28] T. Koren, L. Talker, M. Dinerstein, R. Vitek, Consistent semantic attacks on optical flow, in: Proceedings of the Asian Conference on Computer Vision, 2022, pp. 1658–1674.

[29] J. Schmalfuss, L. Mehl, A. Bruhn, Distracting downpour: Adversarial weather attacks for motion estimation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 10106–10116.

[30] A.H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, O. Beijbom, Pointpillars: Fast encoders for object detection from point clouds, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 12697–12705.

[31] N. Chodosh, D. Ramanan, S. Lucey, Re-Evaluating LiDAR Scene Flow, in: 2024 IEEE/CVF Winter Conference on Applications of Computer Vision, WACV, 2024, pp. 5993–6003.