

# TeFF: Tracking-enhanced Forgetting-free Few-shot 3D LiDAR Semantic Segmentation

Junbao Zhou<sup>1,2</sup>, Jilin Mei<sup>1,†</sup>, Pengze Wu<sup>1,2</sup>, Liang Chen<sup>1</sup>, Fangzhou Zhao<sup>1</sup>, Xijun Zhao<sup>3,4</sup>, Yu Hu<sup>1,2,†</sup>

**Abstract**—In autonomous driving, 3D LiDAR plays a crucial role in understanding the vehicle’s surroundings. However, the newly emerged, unannotated objects presents few-shot learning problem for semantic segmentation. This paper addresses the limitations of current few-shot semantic segmentation by exploiting the temporal continuity of LiDAR data. Employing a tracking model to generate pseudo-ground-truths from a sequence of LiDAR frames, our method significantly augments the dataset, enhancing the model’s ability to learn on novel classes. However, this approach introduces a data imbalance biased to novel data that presents a new challenge of catastrophic forgetting. To mitigate this, we incorporate LoRA, a technique that reduces the number of trainable parameters, thereby preserving the model’s performance on base classes while improving its adaptability to novel classes. This work represents a significant step forward in few-shot 3D LiDAR semantic segmentation for autonomous driving. Our code is available at <https://github.com/BowmanChow/Track-no-forgetting>.

## I. INTRODUCTION

In autonomous driving, 3D LiDAR has been a pivotal sensor due to its proficiency in providing precise 3D position information of surrounding objects [1]. This precision is particularly important for semantic segmentation tasks. Semantic segmentation on 3D LiDAR usually leverages deep learning model trained on a large quantity of annotated data [2]–[19]. However, the autonomous driving scene [20] introduces more challenges to deep learning semantic segmentation due to its complexity. In a dynamic environment, the semantic segmentation model may be required to predict newly emerged objects, which is not annotated during training. Additionally, these newly emerged objects (i.e. novel objects) usually lack pixel-level annotations due to the difficulties in collecting and annotating 3D point cloud data. These challenges present the few-shot semantic segmentation problem, which becomes crucial for enhancing the capabilities of autonomous driving systems.

Taking safety into consideration, we extend the few-shot semantic segmentation problem to generalized few-shot semantic segmentation [21], [22]. Both settings involve a base training stage with abundant annotated data and a

\*This work was supported by National Natural Science Foundation of China under Grant No.U23B2034, No.62203424, and No.62176250; and in part by the Innovation Program of Institute of Computing Technology, Chinese Academy of Sciences under Grant No. 2024000112.

<sup>1</sup>Research Center for Intelligent Computing Systems, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100190, China.

<sup>2</sup>School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing, 100049, China.

<sup>3</sup>China North Artificial Intelligence & Innovation Research Institute.

<sup>4</sup>Collective Intelligence & Collaboration Laboratory (CIC).

<sup>†</sup>Correspondence: Jilin Mei, Yu Hu, {meijilin, huyu}@ict.ac.cn

novel data fine-tuning stage with only a few annotated novel classes. However, the generalized one requires the model to be evaluated on both base objects and novel objects while the former one only needs to predict novel objects. Obviously, the generalized few-shot semantic segmentation poses a bigger challenge that needs to be addressed rigorously.

Most of the existing research on generalized few-shot 3D LiDAR semantic segmentation [23] focuses on adapting the model to a few annotated novel data, while preserving the performance on base classes. However, by carefully investigating the LiDAR dataset in autonomous driving scene, we find that the LiDAR data has sequential characteristics from a temporal perspective, which opens a new opportunity for data augmentation. We employ tracking methods [24] to track LiDAR sequences with a few annotated ground truths, and then append the tracking results to extend the dataset. Those tracking results are considered as “pseudo ground truths”, which are combined with ground truths to fine-tune the model in the novel stage.

However, a notable problem is raised by the tracking-augmentation that the “pseudo ground truths” contain much more points belonging to novel objects than base objects. This data imbalance [25] leads to degradation of accuracy on base classes after novel data fine-tuning, which is known as catastrophic forgetting [26]. To mitigate this issue, we further introduce Low-rank Adaptation (LoRA) [27] in few-shot semantic segmentation, which is primarily used in fine-tuning large language models (LLM) [28], [29] on novel tasks. By comparing LLM fine-tuning and novel data fine-tuning in few-shot learning, we find the common characteristics of both tasks is fitting well on novel data while preserving the knowledge of base data. Therefore, by integrating LoRA, our method achieves the goal of **forgetting-free** and maintains a good balance of accuracy between base classes and novel classes.

Finally, we conduct extensive experiments on SemanticKITTI [20] and show that our method has great improvement on previous few-shot 3D LiDAR semantic segmentation methods and achieves the highest accuracy.

In conclusion, we make the following contributions:

- We discover the sequential characteristic of 3D LiDAR data in autonomous driving and leverage it by augmenting the novel data with tracking method [24].
- We introduce LoRA [27] to solve the catastrophic forgetting problem and achieve high accuracy on both base classes and novel classes.
- Experiments show that our method outperforms baseline methods and is effective for few-shot 3D LiDAR

semantic segmentation, with a noticeable improvement on novel classes.

## II. RELATED WORK

### A. 3D LiDAR Semantic Segmentation

3D LiDAR data is unordered and unstructured, which presents a great challenge for segmentation tasks. PointNet [2] is a milestone in addressing the unstructured problem by proposing a shared MLP network. It extracts features from the whole LiDAR scan directly and then aggregates all the unstructured features through MaxPooling. Unaware of local features, the performance of PointNet [2] is limited. Following PointNet [2], several methods [3], [4], [5], [6] further propose several convolution methods on point cloud to extract local features. Moreover, PointNet++ [7] proposes multi-scale sampling rather than convolution to extract multi-scale local features.

As for outdoor scenes, most point cloud segmentation methods transform the unstructured point cloud data into structured 2D data. SqueezeSeg [8], SqueezeSegv2 [9], RangeNet++ [10], SalsaNext [11] and RangeFormer [12] project the point cloud to a range view (frontal view) image, and utilize 2D convolution network to segment the projected 2D image. Other than range view, bird-eye-view (BEV) is another option to project the point cloud into structured 2D image. PolarNet [13], Salsanet [14] adopt the BEV projection to overcome the data sparsity.

Unlike range-view and BEV, voxelization is another method to convert point cloud into structured data while preserving 3D information. OccuSeg [15], SSCN [16], SegCloud [17] and SPVConv [18] apply 3D convolution networks on voxelized point cloud for LiDAR segmentation. Unlike voxelization, Cylinder3D [19] proposes cylinder partition of point cloud, which also preserve 3D information.

### B. Few-shot 3D LiDAR Semantic Segmentation

Few-shot semantic segmentation is a problem of making prediction after training on a few labeled novel data. Chen et al. [30] proposed a multi-view comparison component that exploits the redundant views of the support set, and extracts prototype features from each view. Zhao et al. [31] introduced EdgeConv and self-attention to design a multi-level feature learning network, that learn the geometric and semantic information between points. Lai et al. [32] explicitly point out that the background ambiguity problem is the main challenge in 3D point cloud semantic segmentation, thus the conventional loss function will lead to degradation of accuracy on few-shot learning.

Previously, we addressed the background ambiguity by introducing unbiased cross entropy loss and unbiased distillation loss [33] and we further utilized semantic vectors to enhance the capability of model on fitting novel data [23].

### C. Tracking Methods

3D Multi-Object Tracking (MOT) involves tracking objects in 3D LiDAR data. Currently, 3D MOT methods can be categorized into “Tracking-By Detection” (TBD) and “Joint

Detection and Tracking” (JDT). Weng et al. [34] firstly pioneered the TBD method, tracking by Linear Kalman Filter and 3D IOU, which is simple yet well-performed. SimpleTrack [35], Eagermot [36] and Camo-mot [37] further enhanced the TBD method. Feichtenhofer et al. [38] firstly proposed JDT method and later, Bergmann et al. [39], Zhang et al. [40] and Huang et al. [41] improved it. Recently, CenterPoint [42] proposed a novel tracking method by detecting objects’ center and associating them across frames. Currently, TBD methods are generally more precise than JDT methods.

However, 3D MOT methods are not compatible with our task, since in few-shot semantic segmentation, only a few novel objects are annotated and we do not have a well-performed detection model to detect novel objects. Therefore, we adopt video object segmentation (VOS) methods, which don’t need semantic information and track each objects by only given annotations in the 1st frame. Early VOS methods [43]–[47] performed feature matching between first frame and following frames, but are challenged by occluded or changing objects. Recently, memory-based VOS method [24], [48]–[52] have raised research interest and achieved high accuracy on most challenging VOS tasks.

## III. METHODOLOGY

### A. Formulation of Few-shot Learning

Let  $\mathcal{X}$  denotes the input space (i.e. LiDAR data space), each  $X \in \mathcal{X}$  denotes a 3D LiDAR scan. Without loss of generality, we assume  $|X| = N$ , where  $N$  is the number of points in  $X$ . The goal of LiDAR semantic segmentation is to assign a class from class space  $\mathcal{C} = \{c_1, c_2, \dots, c_k\}$  to each point in  $X$ . Therefore, the segmentation result  $Y$  belongs to output space  $\mathcal{C}^N$ . Given a training set  $\mathcal{T} = \mathcal{X} \times \mathcal{C}^N$ , the mapping procedure  $\mathcal{X} \mapsto \mathcal{C}^N$  is performed by a parameterized model  $f_\theta$ , which takes  $X$  as input and produce point-wise class probability, i.e.  $f_\theta : \mathcal{X} \mapsto \mathbb{R}^{|\mathcal{C}| \times N}$ . The output mask is further obtained by

$$Y = \{\arg \max_{c \in \mathcal{C}} f_\theta(c, x_p) | p = 1, \dots, N\} \quad (1)$$

where  $x_p$  denotes a point in a LiDAR scan with index  $p$ , and  $f_\theta(c, x_p)$  represents the predicted probability of class  $c$  at point  $x_p$ .

In few-shot learning setting, the training process is divided into base stage and novel stage. The class spaces in two training stages are  $\mathcal{C}_{base}$  and  $\mathcal{C}_{novel}$ , respectively, and the two class spaces are disjoint, i.e.  $\mathcal{C}_{base} \cap \mathcal{C}_{novel} = \emptyset$ . Note that the background or unlabeled class  $u$  is excluded in both class spaces. Therefore, the overall class space  $\mathcal{C} = \{u\} \cup \mathcal{C}_{base} \cup \mathcal{C}_{novel}$ , and the training set in base stage and novel stage are  $\mathcal{T}_{base} = \mathcal{X} \times \mathcal{C}_{base}^N$  and  $\mathcal{T}_{novel} = \mathcal{X} \times \mathcal{C}_{novel}^N$ , respectively.

In classical few-shot learning setting, only novel classes  $\mathcal{C}_{novel}$  are predicted after two stages of training. However, as for generalized few-shot learning setting, all the classes in  $\mathcal{C}$  are required to be tested after training. We explicitly point out that generalized few-shot learning setting is more

desirable in autonomous driving scene, because with various types of objects on the road, only recognizing novel objects is not sufficient for safety.

We primarily utilize transfer learning [53] to address few-shot semantic segmentation problem, which can be divided into three steps. In the first step, we train the base model  $f_{\theta_{base}}$  on abundant base data through loss function  $\mathcal{L}$ :

$$\theta_{base} = \arg \min_{\theta} \mathcal{L}(\mathcal{X}, \mathcal{C}_{base}^N) \quad (2)$$

Then, we use the base model  $\theta_{base}$  to initialize the model in the next step, and fine-tune the model with a few novel data:

$$\theta_{novel} = \arg \min_{\theta} \mathcal{L}(\mathcal{X}, \mathcal{C}_{novel}^N; \theta_{base}) \quad (3)$$

Note that the quantity of labelled data in  $\mathcal{T}_{novel}$  is so limited, presenting the **few-shot** problem. The final step is testing the prediction of all classes with the final model  $f_{\theta}$ , where the parameter is simply loaded from the novel model (i.e.  $\theta = \theta_{novel}$ ). The prediction of each LiDAR scan is obtained by  $\hat{Y} = f_{\theta}(X)$ .

### B. Base Model Training

In base training stage, we utilize the whole training set of SemanticKITTI [20], which contains abundant data annotated with classes  $c \in \{u\} \cup \mathcal{C}_{base}$ . Similar to previous semantic segmentation methods, we use weighted cross entropy loss and the Lovász softmax loss.

**Weighted Cross Entropy Loss.** SemanticKITTI is highly imbalanced annotated, for example, the points of class *road* significantly outnumber the points of other classes. Similar to previous segmentation work [11], we incorporate weighted cross entropy loss to overcome this biased distribution. With an input LiDAR scan  $X \in \mathcal{X}$  and its corresponding ground truth label  $Y \in \mathcal{C}^N$ , the conventional cross entropy loss at point  $x_p$  is calculated by:

$$\mathcal{L}_{CE}(x_p, y_p) = -\log p(y_p, x_p) \quad (4)$$

where  $p(y_p, x_p) = f_{\theta}(y_p, x_p)$  is the predicted probability of the ground truth class  $y_p$  at point  $x_p$ . The weighted cross entropy loss  $\mathcal{L}_{CE}^w$  is formulated by:

$$\mathcal{L}_{CE}^w = \frac{w_{y_p}}{\sum_{c \in \mathcal{C}} w_c} \mathcal{L}_{CE}(x_p, y_p) \quad (5)$$

$$w_c = \frac{1}{\sqrt{M_c}} \quad (6)$$

where  $M_c$  denotes the number of points belongs to class  $c$  in the whole training set.

**Lovász Softmax loss.** Similar to previous segmentation work [11], we also utilize Lovász softmax loss [54] to maximize the mIoU of our model. Lovász softmax loss is defined as:

$$\mathcal{L}_{LS} = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \overline{\Delta_{\mathcal{J}_c}}(m(c)), \quad (7)$$

$$m(c) = \begin{cases} 1 - f_{\theta}(c, x_p) & \text{if } c = y_p \\ f_{\theta}(c, x_p) & \text{otherwise} \end{cases} \quad (8)$$

where  $\mathcal{J}_c$  defines the Jaccard index, and  $\overline{\Delta_{\mathcal{J}_c}}$  is the Lovász extension of the Jaccard index.

The final loss function of the base training stage is :

$$\mathcal{L}_{base} = \mathcal{L}_{CE} + \mathcal{L}_{LS} \quad (9)$$

### C. Extending Novel Data with Tracking Method

In autonomous driving scene, the LiDAR data is collected over a continuous time period. Therefore, the LiDAR data is sequential from a temporal perspective. This feature of LiDAR data provides an opportunity for data augmentation via tracking method.

Taking the temporal continuity into consideration, we redefine the dataset as  $\mathcal{T} = \mathcal{X} \times \mathcal{C}^N = \{(X^t, Y^t) | t = 1, 2, \dots, T\}$ , where  $t$  denotes the timestamp of a LiDAR frame. A tracking model [24]  $\text{Track}(\cdot)$  firstly takes an annotated frame  $(X^t, Y^t)$  as input and extracts its features as  $F^t$ . Then,  $\text{Track}(\cdot)$  subsequently takes in the following frames  $\{X^{t+1}, X^{t+2}, \dots\}$  and produces the segmentation  $\{\hat{Y}^{t+1}, \hat{Y}^{t+2}, \dots\}$ . This procedure can be defined as:

$$\hat{Y}^{t+s} = \text{Track}(X^{t+s} | F^t) \quad (10)$$

$$= \text{Track}(X^{t+s} | X^t, Y^t) \quad (11)$$

Because the temporal continuity still holds in the reverse manner, the tracking model can also predict segmentation in a reverse LiDAR sequence. Therefore, we can obtain the segmentation  $\{\hat{Y}^{t-1}, \hat{Y}^{t-2}, \dots\}$  of a reverse LiDAR sequence:

$$\hat{Y}^{t-s} = \text{Track}(X^{t-s} | X^t, Y^t) \quad (12)$$

We hereby define the segmentation produced by the tracking model as **pseudo ground truth**. With the labelled ground truth  $(X^t, Y^t)$ , we combine them and construct a augmented dataset  $\hat{\mathcal{T}} = \mathcal{X} \times \mathcal{C}^N$ , where:

$$\hat{\mathcal{T}} = \{(X^t, Y^t), (X^{t+s}, \hat{Y}^{t+s}) | s = -T, \dots, -1, 1, \dots, T\} \quad (13)$$

and  $T$  denotes the max tracking number of frames.

Since annotation of novel classes is limited in few-shot learning, data augmentation is crucial to prevent over-fitting. The augmented dataset  $\hat{\mathcal{T}}$  provides more information of novel data and improves the performance in the novel fine-tuning stage.

### D. Novel Data fine-tuning

As is described in Sec. III-A, model  $f_{\theta}$  performs a mapping from input space to point-wise probability, i.e.  $f_{\theta} : \mathcal{X} \mapsto \mathbb{R}^{|\mathcal{C}| \times N}$ . This prediction process is accomplished by the combination of a backbone network  $\text{BN}(\cdot)$  and a classification head  $\text{CLS}(\cdot)$ . The prediction process of model  $f_{\theta}$  can be defined as:

$$\hat{Y}^t = f_{\theta}(X^t) \quad (14)$$

$$= \text{CLS}(\text{BN}(X^t)) \quad (15)$$

We adopt transfer learning [53] in few-shot semantic segmentation. In novel fine-tuning stage, we instantiate a new classification head  $\text{CLS}_{novel}(\cdot)$  to predict the novel classes. The output of  $\text{CLS}_{novel}(\cdot)$  is concatenated with the output

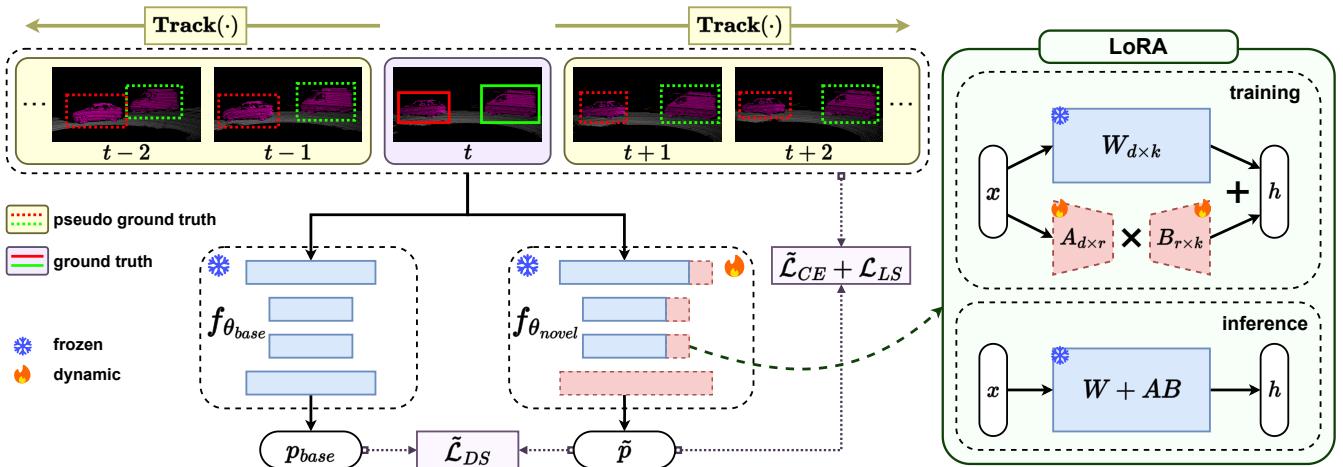


Fig. 1. **Demonstration of our TeFF.** In novel data fine-tuning stage, we firstly track each ground truth with tracking model  $\text{Track}(\cdot)$ , forwardly ( $t+T$ ) and backwardly ( $t-T$ ). The tracking results serve as pseudo ground truths and are combined with ground truths to supervise the novel model. We use unbias cross entropy  $\tilde{\mathcal{L}}_{CE}$ , unbias distillation  $\tilde{\mathcal{L}}_{DS}$  and Lovász softmax loss  $\mathcal{L}_{LS}$  to fine-tune the model. We further apply LoRA to novel model, which reduces the trainable parameters, thus achieving the goal of **forgetting-free**.

of the base classification head  $\text{CLS}_{base}(\cdot)$ . The parameters in  $\text{BN}(\cdot)$  and  $\text{CLS}_{base}(\cdot)$  are directly loaded from the base model.

**Mitigating Forgetting with Low Rank Adaptation.** The tracking-augmentation method is able to augment the data with limited annotation extensively. However, the tracking method predominantly focuses on novel classes, which results the ratio of novel classes points in the augmented data significantly higher than in the overall dataset. This imbalance biases the distribution of the augmented dataset and directs the model’s learning focus towards novel classes, leading to catastrophic forgetting. This phenomenon is where the model’s ability to recognize base classes deteriorates as it increasingly focuses on novel classes. This consequence is particularly problematic as our goal is the generalized few-shot learning problem (outlined in Sec. III-A). We aim to develop a model that maintains high accuracy across both base and novel classes. To mitigate the forgetting issue, we incorporate the Low Rank Adaptation (LoRA) [27] approach during the fine-tuning phase.

LoRA is a technique primarily used in large pre-trained models fine-tuning. In traditional fine-tuning, all of the parameters of a pre-trained model are updated during the training process on a new task, which is computationally expensive and time-consuming. Besides, updating the whole model probably leads to catastrophic forgetting, which harms the accuracy on original task after fine-tuning. The core idea of LoRA is to adapt a pre-trained model to a new task with minimal changes, enhancing the model’s accuracy on new task while preserving the model’s performance on original task. LoRA achieve this goal by introducing small, trainable weights rather than updating the model’s original weights directly. To be more specific, most of the weights can be written in form of matrices, with the denotation  $W \in \mathbb{R}^{d \times k}$ . During fine-tuning, LoRA constrains the update by decompo-

sition:  $W + \Delta W = W + BA$ , where  $\text{rank}(A) = \text{rank}(B) \ll \min(d, k)$ . In other words, the trainable parameters in  $A$  and  $B$  are far less than that in  $W$ . During fine-tuning,  $W$  is frozen and only  $A$  and  $B$  receive gradient updates. As for forward pass, the original output is added on the output of  $BA$ , i.e

$$h = (W + \Delta W)x = Wx + BAx \quad (16)$$

This process ensures that the model’s original capabilities are retained while it learns to recognize new classes.

Although not identical, few-shot learning is similar to the large model fine-tuning. As is shown in Fig. 1, we incorporate LoRA in the novel training stage and significantly reduce the number of trainable parameters. LoRA is only applied to  $\text{BN}(\cdot)$  while  $\text{CLS}_{base}(\cdot)$  and  $\text{CLS}_{novel}(\cdot)$  are kept dynamic. This setting effectively counteract the imbalance issue and mitigate the risk of catastrophic forgetting. This approach not only preserves the model’s performance on base tasks but also enhances its accuracy on novel tasks, aligning with our goal of generalized few-shot learning.

**Unbias Cross Entropy Loss.** Following our previous work [23], [33], we empirically choose unbias cross entropy loss to mitigate the gap between base training and novel data fine-tuning. The unbias cross entropy loss  $\tilde{\mathcal{L}}_{CE}$  is defined as follows:

$$\tilde{\mathcal{L}}_{CE} = -\log \tilde{p}(y_p, x_p) \quad (17)$$

where

$$\tilde{p}(c, x_p) = \begin{cases} f_\theta(c, x_p) & \text{if } c \in \mathcal{C}_{novel} \\ \sum_{c \in \{u\} \cup \mathcal{C}_{base}} f_\theta(c, x_p) & \text{otherwise} \end{cases} \quad (18)$$

**Unbias Distillation Loss.** In transfer learning setting, the base model  $f_{\theta_{base}}$  serves as a teacher model and supervise the student model (i.e. the novel model  $f_{\theta_{novel}}$ ) through distillation loss  $\mathcal{L}_{DS}$ . However, the traditional distillation loss does not take into account that novel objects are annotated

TABLE I  
COMPARISON WITH BASELINES ON SEMANTICKITTI VALIDATION  
SET.

Shot	Method	mIoU	mIoU <sub>base</sub>	mIoU <sub>novel</sub>
	Base Model	-	58.7	-
10	GFSS	49.1	56.8	20.3
	GFSS <sub>dyn</sub>	47.8	53.5	26.3
	LwF	48.0	53.3	28.4
	UBLoss	50.1	55.7	28.8
	SemVec	51.5	56.1	34.3
	<b>TeFF (Ours)</b>	<b>55.3</b>	<b>58.7</b>	<b>42.6</b>
5	GFSS	49.5	56.3	23.9
	GFSS <sub>dyn</sub>	48.1	53.6	27.5
	LwF	46.7	53.0	23.1
	UBLoss	49.6	56.4	23.8
	SemVec	49.3	55.0	27.6
	<b>TeFF (Ours)</b>	<b>53.9</b>	<b>58.1</b>	<b>37.3</b>
2	GFSS	48.3	55.2	22.4
	GFSS <sub>dyn</sub>	46.4	52.5	23.5
	LwF	46.8	52.3	26.1
	UBLoss	48.9	54.8	26.6
	SemVec	46.8	51.8	27.9
	<b>TeFF (Ours)</b>	<b>53.2</b>	<b>58.6</b>	<b>32.8</b>
1	GFSS	48.6	55.5	22.6
	GFSS <sub>dyn</sub>	43.9	48.2	27.9
	LwF	43.1	47.2	27.6
	UBLoss	48.5	53.8	28.8
	SemVec	40.5	44.2	26.5
	<b>TeFF (Ours)</b>	<b>52.4</b>	<b>58.0</b>	<b>31.4</b>

as background in the base training stage. Similar to previous few-shot semantic segmentation work [23], [33], we bridge this gap by using unbias distillation loss  $\tilde{\mathcal{L}}_{DS}$ , which is defined as:

$$\tilde{\mathcal{L}}_{DS} = -p_{base}(y_p, x_p) \log \tilde{p}(y_p, x_p) \quad (19)$$

where

$$\tilde{p}(c, x_p) = \begin{cases} f_\theta(c, x_p) & \text{if } c \in \mathcal{C}_{base} \\ \sum_{c \in \{u\} \cup \mathcal{C}_{novel}} f_\theta(c, x_p) & \text{otherwise} \end{cases} \quad (20)$$

#### IV. EXPERIMENTS

##### A. Dataset and Evaluation Metrics

**SemanticKITTI.** We primarily choose SemanticKITTI to demonstrate the effectiveness of our method. SemanticKITTI is a large scale LiDAR dataset features with over 43K 3D LiDAR scan, which are collected in driving scene and provided in sequences. In the semantic segmentation task, SemanticKITTI provides 20 annotated classes.

Identical to the official config of SemanticKITTI, we split the dataset into 3 subsets: sequences 00 - 07 and 09 - 10 are used for training, sequences 08 is for validation and sequences 11 - 21 are used for testing. As for few-shot learning setting, we set the **car**, **person**, **bicyclist**, and **motorcyclist** as the novel classes and the other 16 classes as base classes.

**Evaluation Metrics.** Similar to our previous work [23], [33], we evaluate our method with mIoU, and we further calculate mIoU<sub>base</sub> and mIoU<sub>novel</sub> for base classes and novel classes separately.

##### B. Baseline and Implementation Details

**Baselines.** We compare our method against several few-shot semantic segmentation methods used in 3D LiDAR data:

- **GFSS.** [22] Generalized few-shot semantic segmentation. After the base training stage, all the parameters in model's backbone are frozen and only the parameters in classification head receive gradient update.
- **GFSS<sub>dyn</sub>.** Share the same config with GFSS but during novel training stage, the parameters in the backbone are not frozen (i.e. dynamic) and also receive gradient update .
- **LwF.** [55] Learning without forgetting. During the novel fine-tuning stage, the predicted probabilities of base model are used to supervise the novel model through distillation loss.
- **UBLoss.** [33] Unbias cross entropy and distillation loss. It incorporate the background information and better mitigate the catastrophic forgetting problem.
- **SemVec.** [23] Integrating semantic vectors into few-shot semantic segmentation. During novel fine-tuning stage, semantic vectors are multiplied with the probabilities produced by classifiers, thus incorporating semantic information and enhancing the performance of few-shot learning.

**Model Settings.** We evaluate our method with SalsaNext [11], a 3D LiDAR segmentation network, as it is fast and still holds high accuracy on SemanticKITTI. As for tracking, most of the mainstream 3D LiDAR tracking methods require detection model [56]. This is not compatible with our few-shot setting as we don't have a model to predict on novel objects until novel fine-tuning stage. Therefore, we adopt a video tracking method, DeAOT [24], which does not require semantic information of novel classes and only needs the annotation of objects in the 1st frame. Note that DeAOT model requires 2D images as input, and to make it compatible with 3D LiDAR data, we project the 3D LiDAR into 2D range-view, with resolution 2048 × 64. The projected range-view frames are fed subsequently into DeAOT and produce tracking results. The predicted tracking results are reverse-projected back to 3D LiDAR and serve as **pseudo ground truth**, which will be used in finetuning the SalsaNext model.

**Training Details.** As is described in Sec. III-A, we adopt transfer learning, which contains two stages: base training and novel data fine-tuning. The base training stage utilize the whole training split while the novel classes (**car**, **person**, **bicyclist**, and **motorcyclist**) are labeled as background. During novel data fine-tuning, to align with few-shot learning setting, we randomly sample  $m$  scans for each novel class (i.e.  $m$ -shot) from the training split. Notably, considering our method use tracking model to extend data, we particularly ensure a minimum gap of 250 between each LiDAR scan, to avoid data redundancy. On both two training stages, we train the model for 160 epochs with batch size 14, which is sufficient for model to fully fit on the data.

Our proposed method TeFF and all the baselines share the same base training stage and start the novel fine-tuning

TABLE II  
COMPARISON WITH BASELINES ON SEMANTICKITTI TESTING SET.

Shot	Method	bicycle	motorcycle	truck	other-vehicle	road	parking	sidewalk	other-ground	building	fence	vegetation	trunk	terrain	pole	traffic-sign	car	person	bicyclist	motorcyclist	mIoU <sub>base</sub>	mIoU <sub>novel</sub>	mIoU
10	GFSS	30.4	26.1	27.3	21.7	90.1	57.1	73.5	27.2	84.9	53.2	77.6	60.5	63.0	49.7	55.2	80.4	0.0	1.3	0.0	53.2	20.4	46.3
	GFSS <sub>dyn</sub>	16.9	23.9	29.8	18.2	89.5	55.8	72.3	26.8	85.7	53.0	77.0	60.0	62.7	47.9	52.4	77.6	12.8	11.3	5.7	51.5	26.9	46.3
	LwF	19.4	24.3	32.9	18.9	89.3	54.0	70.9	24.8	85.8	52.7	77.0	59.4	61.7	45.4	50.7	78.0	13.9	12.5	5.6	51.1	27.5	46.2
	UBLoss	11.0	24.5	28.1	12.4	90.3	57.7	72.5	24.1	86.0	55.0	78.2	60.9	64.1	52.8	49.9	88.7	13.8	10.8	3.7	51.2	29.3	46.6
	SemVec	33.7	25.3	26.5	21.2	90.2	57.4	72.2	27.0	84.1	50.7	76.4	61.1	63.8	49.8	48.5	87.2	21.3	11.6	3.7	52.5	31.0	48.0
	<b>TeFF (Ours)</b>	19.9	29.9	26.5	20.6	90.2	59.2	72.9	28.3	85.6	55.1	79.0	62.4	64.3	53.0	56.8	89.5	24.8	18.8	6.6	<b>53.6</b>	<b>34.9</b>	<b>49.7</b>
5	GFSS	27.8	30.3	25.8	23.1	90.1	56.6	72.9	26.9	85.8	53.6	77.3	59.2	62.9	45.9	56.4	87.0	4.9	0.0	0.0	53.0	23.0	46.7
	GFSS <sub>dyn</sub>	38.1	28.8	14.6	18.4	89.5	57.0	69.9	25.9	85.5	54.6	77.1	59.3	61.0	46.2	52.1	86.4	15.2	1.9	1.4	51.9	26.2	46.5
	LwF	26.9	15.4	4.1	14.7	88.6	56.2	69.5	26.7	85.5	54.7	75.4	58.1	59.8	46.8	52.1	85.1	11.7	1.9	1.3	49.0	25.0	43.9
	UBLoss	23.8	28.2	22.6	18.0	89.8	57.5	71.3	26.2	84.9	55.1	77.8	62.0	62.2	52.2	51.9	87.0	6.4	0.0	0.7	52.2	23.5	46.2
	SemVec	36.4	28.1	24.2	23.3	89.5	53.1	70.5	27.6	84.2	48.3	76.0	61.1	62.6	38.2	54.7	88.2	11.4	2.4	1.9	51.9	26.0	46.4
	<b>TeFF (Ours)</b>	31.3	30.4	26.3	20.8	90.5	60.3	72.3	27.4	85.8	55.3	78.8	62.4	62.8	53	57	89.3	24.6	10.6	6.4	<b>54.3</b>	<b>32.7</b>	<b>49.8</b>
2	GFSS	27.4	24.0	28.6	21.8	90.3	57.0	72.6	24.5	85.9	53.0	77.3	56.7	61.8	48.0	56.5	87.0	0.0	2.4	0.0	52.4	22.4	46.0
	GFSS <sub>dyn</sub>	20.5	20.9	24.8	12.5	89.1	53.5	70.7	21.2	85.3	53.0	79.4	57.7	63.8	48.6	51.9	85.8	0.8	6.1	0.5	50.2	23.3	44.5
	LwF	16.8	26.1	22.2	12.5	88.2	51.1	70.7	20.9	86.1	54.7	79.1	56.8	63.8	49.5	51.6	84.3	0.9	7.6	0.5	50.0	23.3	44.4
	UBLoss	19.4	25.9	27.9	15.5	89.9	57.5	71.8	20.7	85.6	53.9	79.4	61.8	62.8	52.6	51.3	87.0	0.3	6.5	0.6	51.7	23.6	45.8
	SemVec	30.2	20.7	18.7	12.5	89.2	55.5	70.2	24.2	81.9	46.6	75.8	59.6	61.5	36.4	55.5	88.0	3.0	10.4	0.4	49.2	25.5	44.2
	<b>TeFF (Ours)</b>	26.3	29	28.1	21.9	90.1	60.5	72.1	26.7	85.5	54.8	79.1	62.7	63.2	53.2	54.8	88.8	14.2	11.8	6.1	<b>53.9</b>	<b>30.2</b>	<b>48.9</b>
1	GFSS	15.2	17.5	29.6	19.7	89.6	56.2	71.3	10.7	82.8	47	73.2	54.6	59.5	46.4	55.4	86.1	0	2.7	0	48.6	22.2	43.0
	GFSS <sub>dyn</sub>	0	17	17.8	12.4	89	52.1	71.3	12.9	84.5	47.8	74.2	51.8	62.4	43.4	39.9	85.4	0.7	11	0	45.1	24.3	40.7
	LwF	0	19.8	21.1	12.4	89	48.6	69.2	16	84.4	46.2	75.1	51.1	64.1	45	48.1	85.8	0.6	12	0.1	46.0	24.6	41.5
	UBLoss	3.5	25.5	26.9	17.2	89.3	54.2	70.3	8.4	83.3	50.7	76.9	55	63.9	49.2	45.2	85.2	0.4	12.3	0.2	48.0	24.5	43.0
	SemVec	6.1	15.2	3.2	21.2	88.6	54.1	69.8	12.8	80.8	40.2	71.9	59.2	62.2	31.1	51.4	85.2	3.3	7.3	0.0	44.5	24.0	40.2
	<b>TeFF (Ours)</b>	32.4	27.1	29	21.5	89.9	58.9	71.6	18.2	85.1	52.5	77	61.4	61.2	53.1	54.1	88.4	9.7	8.7	0	<b>52.9</b>	<b>26.7</b>	<b>47.4</b>

stage with the same base model. By adopting such setting, we ensure all the differences between each method are attributed solely to different novel fine-tuning strategies.

**TeFF Details.** We track each ground truth for 20 frames with tracking gap 15 (discussed in Sec. IV-D). As for LoRA, we apply LoRA on all the up-sample blocks and half of the ResNet blocks [11], while keep other layers frozen. The rank in LoRA is set to 1/4 of the hidden dimension for each layer.

### C. Quantitative Analysis

In Table I, we compare our method TeFF with previous few-shot semantic segmentation methods in 4 different settings, shot = 1, 2, 5 and 10. Our method achieves the highest score in all the 4 settings and establishes a new state-of-the-art in few-shot 3D LiDAR semantic segmentation. Notably, TeFF not only excels in adapting to novel classes, but also preserves a high score on base classes, effectively addressing the problem of catastrophic forgetting. This capability is especially important in generalized few-shot semantic segmentation for autonomous driving, where all the classes should be accurately predicted due to safety concerns. Our method, TeFF, leverages a tracking model to provide sufficient novel data for fine-tuning, and minimizes the catastrophic forgetting by introducing LoRA, which significantly reduces the trainable parameters.

Table II shows the IoU of all the classes on SemanticKITTI testing split. Our method also performs best on mIoU<sub>base</sub>, mIoU<sub>novel</sub> and mIoU. Besides, our method also achieves the highest score in most of the classes.

### D. Ablation Study

TABLE III  
ABLATION STUDY OF LORA CONFIG.

Method	mIoU	mIoU <sub>base</sub>	mIoU <sub>novel</sub>
Freezing	50.9	58.0	24.3
Dynamic	52.1	55.2	<b>40.6</b>
<b>LoRA (Ours)</b>	<b>53.2</b>	<b>58.6</b>	32.8

**Effectiveness of LoRA.** We compare LoRA with two fine-tuning strategies in Table III: (1) Freezing, except the classification head, all the parameters do not receive updates. (2) Dynamic, which tunes all the parameters of the model. LoRA reduces the trainable parameters while not freezes the whole model, thereby preserving good performance on base classes while also fitting well on novel data. Although it is outperformed by the Dynamic strategy on mIoU<sub>novel</sub>, it excels in maintaining good balance between base and novel classes, achieving the highest overall mIoU.

**Analysis on the Gap between Tracked Scans.** Although tracking method can provide sufficient novel data, it is not optimal to utilize every tracking result in fine-tuning. Firstly, the adjacent pseudo ground truths are similar, presenting a data redundancy problem, which probably leads to overfitting. Secondly, using too many samples in fine-tuning is computationally expensive and significantly increases the training time. Therefore, we introduce tracking gap, which means selecting a sample every certain scan in a tracking-generated

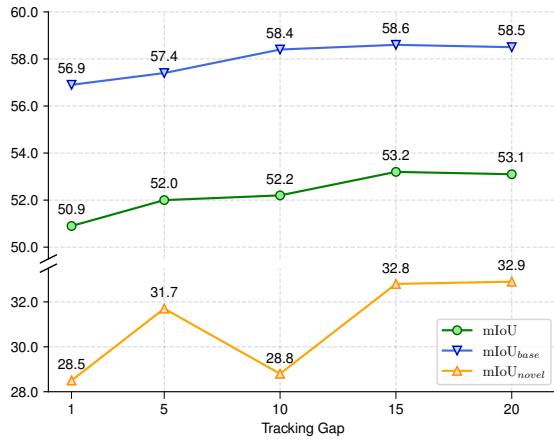


Fig. 2. **Ablation study of tracking gap on SemanticKITTI validation set.** It is tested with shot = 2 and the tracking frame number is 20 (10 forward and 10 backward).

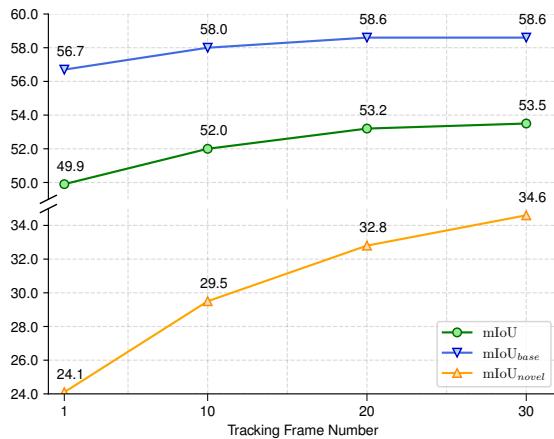


Fig. 3. **Ablation study of tracking frame numbers on SemanticKITTI validation set.** It is tested with shot = 2 and the tracking gap is 15.

sequence. However, if the tracking-generated sequence goes too long, the quality of tracking results tends to degrade. It means that there is a trade-off in tracking gap preventing it from being unlimitedly large. As shown in Fig. 2, the optimal tracking gap is 15, which performs best in the overall mIoU.

**Analysis on the Number of Tracked Scans.** As shown in Fig. 3, an increasing number of scans generally improves the overall mIoU. However, when the tracking frame number exceeds 20, the improvement tends to be minor (mIoU : 53.2 → 53.5). Considering that the tracking model requires much more GPU memory and becomes slow with more tracking frames, we set this value to 20 (10 forward and 10 backward), which is sufficient to demonstrate the effectiveness of our method.

## V. CONCLUSIONS

In this work, we address the few-shot 3D LiDAR semantic segmentation problem. By exploiting the sequential characteristic of 3D LiDAR data in autonomous driving, we leverage tracking method to augment the data with a few an-

notated ground truths. Those tracking results are considered as pseudo ground truths and combined with ground truths to fine-tune the model in novel stage. However, the tracking results are biased towards novel classes, which will cause catastrophic forgetting. By introducing LoRA, we solve the forgetting problem and achieve the highest mIoU on both base classes and novel classes.

## REFERENCES

- [1] Y. Li, L. Ma, Z. Zhong, F. Liu, M. A. Chapman, D. Cao, and J. Li, “Deep learning for lidar point clouds in autonomous driving: A review,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 8, pp. 3412–3432, 2021.
- [2] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “Pointnet: Deep learning on point sets for 3d classification and segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.
- [3] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, “Kpconv: Flexible and deformable convolution for point clouds,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 6411–6420.
- [4] W. Wu, Z. Qi, and L. Fuxin, “Pointconv: Deep convolutional networks on 3d point clouds,” in *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, 2019, pp. 9621–9630.
- [5] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, “Dynamic graph cnn for learning on point clouds,” *ACM Transactions on Graphics (tog)*, vol. 38, no. 5, pp. 1–12, 2019.
- [6] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, “Graph attention networks,” *arXiv preprint arXiv:1710.10903*, 2017.
- [7] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, “Pointnet++: Deep hierarchical feature learning on point sets in a metric space,” *Advances in neural information processing systems*, vol. 30, 2017.
- [8] B. Wu, A. Wan, X. Yue, and K. Keutzer, “Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud,” in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 1887–1893.
- [9] B. Wu, X. Zhou, S. Zhao, X. Yue, and K. Keutzer, “Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud,” in *2019 international conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 4376–4382.
- [10] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, “Rangenet++: Fast and accurate lidar semantic segmentation,” in *2019 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2019, pp. 4213–4220.
- [11] T. Cortinhal, G. Tzelepis, and E. Erdal Aksoy, “Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds,” in *Advances in Visual Computing: 15th International Symposium, ISVC 2020, San Diego, CA, USA, October 5–7, 2020, Proceedings, Part II*. Springer, 2020, pp. 207–222.
- [12] L. Kong, Y. Liu, R. Chen, Y. Ma, X. Zhu, Y. Li, Y. Hou, Y. Qiao, and Z. Liu, “Rethinking range view representation for lidar segmentation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 228–240.
- [13] Y. Zhang, Z. Zhou, P. David, X. Yue, Z. Xi, B. Gong, and H. Foroosh, “Polarnet: An improved grid representation for online lidar point clouds semantic segmentation,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9601–9610.
- [14] E. E. Aksoy, S. Baci, and S. Cavdar, “Salsanet: Fast road and vehicle segmentation in lidar point clouds for autonomous driving,” in *2020 IEEE intelligent vehicles symposium (IV)*. IEEE, 2020, pp. 926–932.
- [15] L. Han, T. Zheng, L. Xu, and L. Fang, “Occuseg: Occupancy-aware 3d instance segmentation,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2940–2949.
- [16] B. Graham, M. Engelcke, and L. Van Der Maaten, “3d semantic segmentation with submanifold sparse convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9224–9232.
- [17] L. Tchapmi, C. Choy, I. Armeni, J. Gwak, and S. Savarese, “Segcloud: Semantic segmentation of 3d point clouds,” in *2017 international conference on 3D vision (3DV)*. IEEE, 2017, pp. 537–547.

- [18] H. Tang, Z. Liu, S. Zhao, Y. Lin, J. Lin, H. Wang, and S. Han, "Searching efficient 3d architectures with sparse point-voxel convolution," in *European conference on computer vision*. Springer, 2020, pp. 685–702.
- [19] H. Zhou, X. Zhu, X. Song, Y. Ma, Z. Wang, H. Li, and D. Lin, "Cylinder3d: An effective 3d framework for driving-scene lidar semantic segmentation," *arXiv preprint arXiv:2008.01550*, 2020.
- [20] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "Semantickitti: A dataset for semantic scene understanding of lidar sequences," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9297–9307.
- [21] Z. Tian, X. Lai, L. Jiang, S. Liu, M. Shu, H. Zhao, and J. Jia, "Generalized few-shot semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11 563–11 572.
- [22] J. Myers-Dean, Y. Zhao, B. Price, S. Cohen, and D. Gurari, "Generalized few-shot semantic segmentation: All you need is fine-tuning," *arXiv preprint arXiv:2112.10982*, 2021.
- [23] P. Wu, J. Mei, X. Zhao, and Y. Hu, "Generalized few-shot semantic segmentation for lidar point clouds," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 7622–7628.
- [24] Z. Yang and Y. Yang, "Decoupling features in hierarchical propagation for video object segmentation," *Advances in Neural Information Processing Systems*, vol. 35, pp. 36 324–36 336, 2022.
- [25] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *Journal of Big Data*, vol. 6, no. 1, pp. 1–54, 2019.
- [26] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.
- [27] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.
- [28] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.
- [29] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.
- [30] X. Chen, C. Zhang, G. Lin, and J. Han, "Compositional prototype network with multi-view comparison for few-shot point cloud semantic segmentation," *ArXiv*, vol. abs/2012.14255, 2020.
- [31] N. Zhao, T.-S. Chua, and G. H. Lee, "Few-shot 3d point cloud semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8873–8882.
- [32] L. Lai, J. Chen, C. Zhang, Z. Zhang, G. Lin, and Q. Wu, "Tackling background ambiguities in multi-class few-shot point cloud semantic segmentation," *Knowledge-Based Systems*, 2022.
- [33] J. Mei, J. Zhou, and Y. Hu, "Few-shot 3d lidar semantic segmentation for autonomous driving," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 9324–9330.
- [34] X. Weng, J. Wang, D. Held, and K. Kitani, "3d multi-object tracking: A baseline and new evaluation metrics," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 10 359–10 366.
- [35] Z. Pang, Z. Li, and N. Wang, "Simpletrack: Understanding and rethinking 3d multi-object tracking," in *European Conference on Computer Vision*. Springer, 2022, pp. 680–696.
- [36] A. Kim, A. Ošep, and L. Leal-Taixé, "Eagermot: 3d multi-object tracking via sensor fusion," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 11 315–11 321.
- [37] L. Wang, X. Zhang, W. Qin, X. Li, J. Gao, L. Yang, Z. Li, J. Li, L. Zhu, H. Wang *et al.*, "Camo-mot: Combined appearance-motion optimization for 3d multi-object tracking with camera-lidar fusion," *IEEE Transactions on Intelligent Transportation Systems*, 2023.
- [38] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Detect to track and track to detect," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 3038–3046.
- [39] P. Bergmann, T. Meinhardt, and L. Leal-Taixé, "Tracking without bells and whistles," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 941–951.
- [40] J. Zhang, S. Zhou, X. Chang, F. Wan, J. Wang, Y. Wu, and D. Huang, "Multiple object tracking by flowing and fusing," *arXiv preprint arXiv:2001.11180*, 2020.
- [41] K. Huang and Q. Hao, "Joint multi-object detection and tracking with camera-lidar fusion for autonomous driving," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 6983–6989.
- [42] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3d object detection and tracking," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 11 784–11 793.
- [43] Y. Chen, J. Pont-Tuset, A. Montes, and L. Van Gool, "Blazingly fast video object segmentation with pixel-wise metric learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1189–1198.
- [44] Y.-T. Hu, J.-B. Huang, and A. G. Schwing, "Videomatch: Matching based video object segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 54–70.
- [45] L. Yang, Y. Wang, X. Xiong, J. Yang, and A. K. Katsaggelos, "Efficient video object segmentation via network modulation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6499–6507.
- [46] Z. Yang, Y. Wei, and Y. Yang, "Collaborative video object segmentation by foreground-background integration," in *European Conference on Computer Vision*. Springer, 2020, pp. 332–348.
- [47] P. Voigtlaender, Y. Chai, F. Schroff, H. Adam, B. Leibe, and L.-C. Chen, "Feelvos: Fast end-to-end embedding learning for video object segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 9481–9490.
- [48] S. W. Oh, J.-Y. Lee, N. Xu, and S. J. Kim, "Video object segmentation using space-time memory networks," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9226–9235.
- [49] H. K. Cheng and A. G. Schwing, "Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model," in *European Conference on Computer Vision*. Springer, 2022, pp. 640–658.
- [50] H. Xie, H. Yao, S. Zhou, S. Zhang, and W. Sun, "Efficient regional memory network for video object segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 1286–1295.
- [51] H. K. Cheng, Y.-W. Tai, and C.-K. Tang, "Rethinking space-time networks with improved memory coverage for efficient video object segmentation," *Advances in Neural Information Processing Systems*, vol. 34, pp. 11 781–11 794, 2021.
- [52] Z. Yang, Y. Wei, and Y. Yang, "Associating objects with transformers for video object segmentation," *Advances in Neural Information Processing Systems*, vol. 34, pp. 2491–2502, 2021.
- [53] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2020.
- [54] M. Berman, A. R. Triki, and M. B. Blaschko, "The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4413–4421.
- [55] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.
- [56] X. Li, T. Xie, D. Liu, J. Gao, K. Dai, Z. Jiang, L. Zhao, and K. Wang, "Poly-mot: A polyhedral framework for 3d multi-object tracking," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 9391–9398.