In Defense of Knowledge Distillation for Task Incremental Learning and its Application in 3D Object Detection

Peng YUN[1] Yuxuan LIU[1] and Ming LIU[1]

***Abstract*—Making robots learn skills incrementally is an efficient way to design real intelligent agents. To achieve this, researchers adopt knowledge distillation to transfer old-task knowledge from old models to new ones. However, when the length of the task sequence increases, the effectiveness of knowledge distillation to prevent models from forgetting old-task knowledge degrades, which we call the *long-sequence effectiveness degradation (LED)* problem. In this paper, we analyze the *LED* problem in the task-incremental-learning setting, and attribute it to the inevitable data distribution differences among tasks. To address this problem, we propose to correct the knowledge distillation for task incremental learning with a Bayesian approach. It additionally maximizes the posterior probability related to the data distributions of all seen tasks. To demonstrate its effectiveness, we further apply our proposed corrected knowledge distillation to 3D object detection. The comparison between the results of increment-at-once and increment-in-sequence experiments shows that our proposed method solves the *LED* problem. Besides, it reaches the upper-bound performance in the task-incremental-learning experiments on the *KITTI* dataset. The code and supplementary materials are available at https://sites.google.com/view/c-kd/.**

***Index Terms*—Probability and Statistical Methods; Incremental Learning; Computer Vision for Transportation.**

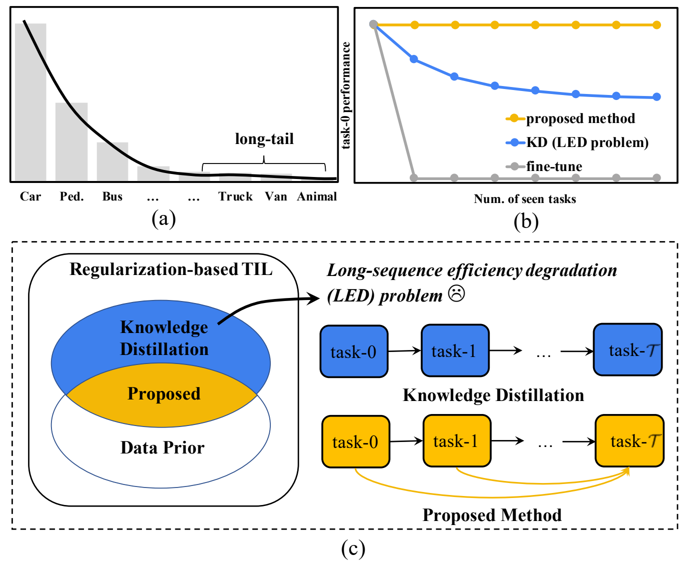

Fig. 1: (a) Histogram of a long-tail distribution. (b) Performance curve of task-0 when the length of the task sequence increases. The fine-tune curve illustrates the catastrophic forgetting. The *KD* curve illustrates the *LED* problem of knowledge distillation. (c) Our proposed method solves the *LED* problem with a Bayesian approach by adding constraints between the new task and all seen tasks.

I. INTRODUCTION

LEARNING skills on top of previous knowledge is a marked feature of intelligent agents. When human beings acquire new knowledge, old skills get preserved, which intensifies our adapting ability to survive in the changing world. Robots run in the same changing world where the data distributions exhibit the long-tail property (Figure 1 (a)) [1]. The massive edge-case data points, lying at the tail of the data distribution, are always unexpected in advance. Take autonomous driving as an instance. Researchers have access to common data in driving scenes through public datasets, but must spend huge effort to enumerate and collect edge-case data points, like different types of trucks and various animals. Many tragic autonomous car accidents can be attributed to unexpected edge cases.

Designing optimal incremental learning algorithms to make robots learn skills incrementally is challenging. It has been recently proved that the optimal incremental learning algorithm is theoretically *NP-HARD* and requires the perfect memory condition [2]. Researchers have made efforts to prevent parametric models from forgetting old knowledge when learning new tasks, i.e., to overcome catastrophic forgetting. Knowledge distillation [3] is one of the methods to prevent parametric models from the catastrophic forgetting. By adding a regularization penalty, it transfers the knowledge from the old model to the new model recursively. It has shown great success in the image classification [4], [5] and image-based 2D object detection tasks [6]–[8].

Knowledge distillation takes effect in reducing forgetting, however, transferring knowledge in the recursion way accumulates to build-up errors. When the length of the task sequence increases, the effectiveness of knowledge distillation drops significantly (Figure 1 (b)), which we call the *long-sequence effectiveness degradation (LED)* problem. Since the unexpected data haunts and frequently appears, researchers have to improve the models on hand repeatedly. In consequence, the *LED* problem hinders the practical usage of knowledge distillation in incremental learning.

In this paper, we aim to solve the *LED* problem of knowledge distillation for *task incremental learning (TIL)*, which

Manuscript received October 15, 2020; revised January 11, 2021; accepted February 5, 2021. This paper was recommended for publication by Editor Lucia Pallottino upon evaluation of the reviewers' comments. The authors would like to thank the editors and the anonymous reviewers for their efforts and comments. This work was supported by the National Natural Science Foundation of China, under grant No. U1713211, Collaborative Research Fund by Research Grants Council Hong Kong, under Project No. C4063-18G, and HKUST-SJTU Joint Research Collaboration Fund, under project SJTU20EG03, awarded to Prof. Ming Liu.

[1]Peng YUN, Yuxuan LIU, and Ming LIU are with the Department of Computer Science Engineering, The Hong Kong University of Science and Technology, Hong Kong SAR, China (e-mail: {pyun,yliuhb,eelium}@ust.hk).

Digital Object Identifier (DOI): see the top of this page.

is the incremental learning scenario to overcome the long-tail distributions in the real world. We first analyze the reason for *LED* problem and attribute it to the inevitable data distribution differences in *TIL*. To address this problem, we propose to correct knowledge distillation for *TIL* with a Bayesian approach. Compared with the original knowledge distillation, our proposed method additionally maximizes the posterior probability related to the data distribution of all seen tasks (Figure 1 (c)). Finally, we demonstrate the effectiveness of the proposed corrected knowledge distillation with an application of 3D object detection. We consider two incremental learning settings: incrementing tasks at once and incrementing tasks in sequence. The experimental results demonstrate the manifestation of *LED* problem and the effectiveness of our proposed method.

II. RELATED WORK

In this section, we first review the regularization-based incremental learning methods and describe the relationship between our proposed method and the existing works, and then review the literature of LiDAR-based 3D object detection.

A. Regularization-based incremental learning

The regularization-based methods alleviate old-task forgetting by adding regularization terms in the objective function. Researchers resolve the catastrophic forgetting with this idea in two general ways: knowledge distillation and reserving priors.

a) Knowledge distillation: Knowledge-distillation-based methods [4]–[8] retain the knowledge of previous tasks by restricting the conditional distribution computed with the updated parameters $p(\hat{\mathcal{Y}}|\mathcal{X}, \boldsymbol{\theta})$ close to that computed with the optimal parameter of previous tasks $p(\hat{\mathcal{Y}}|\mathcal{X}, \boldsymbol{\theta}_0)$. A regularization term, proportional to the distance between these two conditional distributions, is added to the original loss function.

In classification, the distance is commonly measured by the *Kullback-Leibler (KL)* divergence [4], [5]. It can be seamlessly extended from classification to regression tasks by replacing the *KL* divergence with a smoothed-L1-norm or L2-norm. Shmelkov *et al.* [6] first proposed an incremental learning object detector, called *ILOD*, and extended the knowledge distillation to the image-based 2D object detection problem. In recent years, there are follow-up works of *ILOD* [6], including *RILOD* [7] and *Faster-ILOD* [8].

Rannen *et al.* [5] first reported the shortcomings of the knowledge-distillation-based incremental learning methods, and designed *EBLL* to deal with the data distribution differences for image classification tasks. They proposed to project features on a low dimensional manifold with an undercomplete auto-encoder and impeded the new feature deviating from previous task optimal ones. However, it requires additional computation in auto-encoder training and needs to collect low-dimensional features in each optimization step. In contrast, we deal with the inevitable data distribution differences in *TIL* from a probability perspective by maximizing the posterior probability related to the data distributions of all seen tasks. Our proposed method does not cause an additional computational burden when learning new tasks.

b) Reserving priors: Kirkpatrick *et al.* [9] formulated the statistical risk in incremental learning with the posterior probability $p(\boldsymbol{\theta}|\mathcal{D})$ to find the most probable weights given the data $\mathcal{D} = \mathcal{D}_A \cup \mathcal{D}_B$, where $\mathcal{D}_A$ and $\mathcal{D}_B$ denote old-task data and new-task data respectively. They optimized the parameters $\boldsymbol{\theta}$ by maximizing the logarithm posterior probability:

$$\log p(\boldsymbol{\theta}|\mathcal{D}) = \log p(\mathcal{D}_B|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta}|\mathcal{D}_A) - \log p(\mathcal{D}_B|\mathcal{D}_A), \tag{1}$$

where the term $\log p(\boldsymbol{\theta}|\mathcal{D}_A)$ was modeled with Laplacian approximation with the mean given by the optimal parameters $\boldsymbol{\theta}_A^*$ and the covariance matrix approximated by a diagonal *Fisher information matrix (FIM)*. The *FIM* was computed and stored right after training the old task and worked as the prior information of the old tasks. There are multiple follow-up research on the computation of the weight importance measurements [10]–[14]. Zenke *et al.* [10] proposed an online method for estimating the weight importance. Instead of computing the *FIM* in a separate stage after training, they estimate the importance of each parameter calculated with gradients and parameter changes when training the neural network. Aljundi *et al.* [11] redefined the importance weights to an unsupervised setting. Instead of calculating the gradients of the loss function, they adopt the gradients of the squared L2-Norm of the network output, which can be considered as a heuristic approximation of the *FIM*. We adopt the method of evaluating the posterior probability on the seen task data distributions in *EWC* [9] and also consider its alternative *MAS* [11] in our experiments.

In recent days, Liu *et al.* [15] reported that *EWC* failed in the image-based 2D object detection problem, and proposed *IncDet* to facilitate the use of *EWC* in incremental 2D object detection by utilizing pseudo bounding box annotations. Their proposed pseudo-annotation technique exploits the old-task optimal model to generate fake labels offline in order to remedy the lack of old-task class annotations in the new-task data. However, this is a suboptimal solution for the parametric models which require online data augmentation, like randomly translating or rotating objects in LiDAR-based 3D object detection. The incorrect pseudo annotations induce a mass of noise in the training process. We compare our proposed method with *IncDet* [15] in Section V.

B. LiDAR-based 3D object detection

The geometry information captured by LiDARs can be used for perception, and the accurate spatial information is helpful for precise 3D object detection. The challenge for LiDAR-based 3D object detection is that the point clouds captured from LiDARs suffer from the sparsity and are represented as unordered vector lists, which is not suitable for convolution operations. Researchers who intend to percept key objects from LiDAR scans have to deal with the problem: How to extract features from point clouds of LiDARs? The current solution can be categorized into two general categories: *Quantization+Convolution NeuralNetwork (Q+CNN)* based methods, which convert the input point clouds into convolution-friendly representations and then apply *CNNs* to extract features [16]–[23], and PointNet-based methods, which directly extract features from the input point clouds based on PointNet [24] and its variants [25]–[27].

In this paper, we focus on the 3D *region proposal networks (RPNs)* within the *Q+CNN*-based methods. The quantization process, which converting input point clouds into convolution-friendly representations, include projecting the point cloud into bird's-eye-view images [16]–[19] or voxelizing the point cloud into a grid [20]–[23]. In quantization, the 3D spatial information is encoded into bird's-eye-view images or grids with hand-crafted features, like point density, distance, occupancy, etc. Li *et al.* [20] encoded the 3D spatial information of point clouds into grids by hand-crafted features and proposed an encoder-decoder network with 3D dense convolutions by extending a 2D fully convolutional network. Zhou *et al.* [21] proposed an end-to-end network for 3D object detection, called *VoxelNet*, where the voxel-wise features are learned from raw point clouds instead of hand-crafted by researchers. After quantization, high-level features can be extracted with *CNNs* and further used for 3D bounding box estimation.

The *RPN* was first proposed in [28] for image-based 2D object detection task. In [28], an *RPN* takes an image as input and outputs a set of rectangular object proposals, each with an objectness score. The similar idea of *RPN* is extended to LiDAR-based 3D object detection tasks in [18]–[22]. In 3D object detection tasks, the *RPN* finally estimates a classification map and a regression map based on a predefined anchor map. In specific terms, the anchor map contains a set of anchors locating on each discrete location in the 3D space. The classification map predicts the semantic class of each anchor, and the regression map estimates the refined shape and location of 3D bounding box for each anchor.

In recent years, Yan *et al.* [22] proposed *SECOND* based on *VoxelNet* [21] and adopted spatially sparse convolution to deal with the sparsity of point clouds. As a result, the running-time performance of *SECOND* was improved by $4\times$ (230 ms to 50 ms per frame). We adopt *SECOND* as the basic 3D object detector for the use case in 3D object detection. For more details about the the network architecture, please refer to the original paper [22].

III. METHODOLOGY

In this section, we first provide the formal definition of *TIL* and then analyze the reason for the *LED* problem. Based on the analysis, we further describe our Bayesian approach to correct knowledge distillation for *TIL*.

A. Problem definition of TIL

We define the *TIL* following the definition in [29] and consider training a parametric model on a sequence of tasks. Each task task-t consists of a task-specific class set $C^{(t)}$ and a task-specific data distribution $(\mathcal{X}^{(t)}, \mathcal{Y}^{(t)}) \sim P^{(t)}$. Different tasks have different class sets and data distributions, i.e., $C^{(i)} \neq C^{(j)}$ and $P^{(i)} \neq P^{(j)}$, if $i \neq j$. The goal of *TIL* is to control the statistical risk of all seen tasks given limited or no access to data $(\mathcal{X}^{(t)}, \mathcal{Y}^{(t)})$ from previous tasks $t \leq \mathcal{T}$:

$$\sum_{t=1}^{\mathcal{T}} \mathbb{E}_{(\mathcal{X}^{(t)},\mathcal{Y}^{(t)})\sim P^{(t)}}[\ell(f_t(\mathcal{X}^{(t)}, \boldsymbol{\theta}), \mathcal{Y}^{(t)})], \tag{2}$$

where $f_t(\cdot, \boldsymbol{\theta})$ is the parametric model of task t, ℓ is the loss function, and $\mathcal{T}$ is the number of tasks seen so far. For the current task task-$\mathcal{T}$, the statistical risk can be approximated by the empirical risk:

$$\mathcal{L}(\boldsymbol{\theta}, D^{\mathcal{T}}) = \frac{1}{N_{\mathcal{T}}} \sum_{i=1}^{N_{\mathcal{T}}} \ell(f_t(\mathcal{X}_i^{(t)}, \boldsymbol{\theta}), \mathcal{Y}_i^{(t)}), \tag{3}$$

where the dataset of task-t is sampled from its task-specific data distribution, $\{(\mathcal{X}_i^{(t)}, \mathcal{Y}_i^{(t)})\} = \mathcal{D}^{(t)} \sim P^{(t)}$, and $N_{\mathcal{T}}$ is the capacity of $\mathcal{D}^{(t)}$.

The major challenge of *TIL* is that the data is no longer available for previous tasks when training on new tasks, which hinders the evaluation of statistical risk for the new parameter values on previous tasks.

B. Analysis of the LED problem

Knowledge-distillation-based methods prevent parametric models from forgetting old-task knowledge by restricting the conditional distribution $p(\hat{\mathcal{Y}}|\mathcal{X}, \boldsymbol{\theta})$ close to $p(\hat{\mathcal{Y}}|\mathcal{X}, \boldsymbol{\theta}^*_{1..\mathcal{T}-1})$ with the objective function:

$$\begin{aligned}&\mathcal{L}_{KD}(\boldsymbol{\theta}, \boldsymbol{\theta}^*_{1..\mathcal{T}-1}, D^{\mathcal{T}}) = \mathcal{L}(\boldsymbol{\theta}, D^{\mathcal{T}}) \\ &+ \lambda \mathbb{E}_{\mathcal{X}_i^{\mathcal{T}} \sim D^{\mathcal{T}}} \mathrm{KL}(p(\hat{\mathcal{Y}}|\mathcal{X}_i^{\mathcal{T}}, \boldsymbol{\theta}) || p(\hat{\mathcal{Y}}|\mathcal{X}_i^{\mathcal{T}}, \boldsymbol{\theta}^*_{1..\mathcal{T}-1})),\end{aligned} \tag{4}$$

where task-$\mathcal{T}$ is the current task, the $\boldsymbol{\theta}^*_{1..\mathcal{T}-1}$ is the optimal parameters of the old tasks, and the KL denotes the Kullback-Leibler divergence, λ is the weighting factor of the regularization term.

To find the reason for the *LED* problem, we first consider a two-task case where the task sequence contains task-A and task-B. The statistical risk of *TIL* is

$$\begin{aligned}&\mathcal{L}(\boldsymbol{\theta}, \mathcal{D}^A) + \mathcal{L}(\boldsymbol{\theta}, \mathcal{D}^B) \\ =&\mathbb{E}_{(\mathcal{X}^A,\mathcal{Y}^A)\sim P^A}[\ell(f_t(\mathcal{X}^A, \boldsymbol{\theta}), \mathcal{Y}^A)] \\ &+ \mathbb{E}_{(\mathcal{X}^B,\mathcal{Y}^B)\sim P^B}[\ell(f_t(\mathcal{X}^B, \boldsymbol{\theta}), \mathcal{Y}^B)].\end{aligned} \tag{5}$$

The objective function of knowledge distillation is

$$\begin{aligned}&\mathcal{L}_{KD}(\boldsymbol{\theta}, \boldsymbol{\theta}^*_A, \mathcal{D}^B) \\ =&\mathbb{E}_{\mathcal{X}_i^B \sim D^B} KL(p(\hat{\mathcal{Y}}|\mathcal{X}_i^B, \boldsymbol{\theta}) || p(\hat{\mathcal{Y}}|\mathcal{X}_i^B, \boldsymbol{\theta}^*_A)) \\ &+ \mathbb{E}_{(\mathcal{X}^B,\mathcal{Y}^B)\sim P^B}[\ell(f_t(\mathcal{X}^B, \boldsymbol{\theta}), \mathcal{Y}^B)],\end{aligned} \tag{6}$$

where we drop the hyperparameter λ for a better comparison. It requires two conditions to make optimizing equation (6) equivalent to optimizing equation (5): (1) $\boldsymbol{\theta}^*_A$ is optimal for ℓ conditioned on the data distribution P^A; (2) the data distribution P^A is highly related to P^B. In practice, when incrementally training a parametric model on the task-B, we always start with the parameters having converged on the task-A, which is good enough for ℓ conditioned on P^A.

The second condition does not hold in *TIL*. According to the problem definition of *TIL*, different tasks have different data distribution $P^{(i)} \neq P^{(j)}$, and the data distributions can be significantly different among tasks in practice. The tasks can be composed of data points sampled from different generative distributions, as shown in Figure 2 (left). Moreover, data points

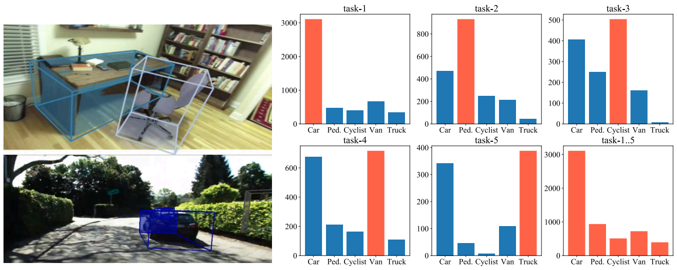

Fig. 2: Top left: an indoor scene from the *SUN-RGBD* dataset [30]. Bottom left: an outdoor scene from the *KITTI* dataset [31]. Right: histograms of the data points sampled from one joint generative distribution in a class-biased way. The task-specific classes are highlighted in red.

of different tasks can also be generated from a joint generative distribution but with different sampling strategies. This is the situation where the parametric model progressively learns the tail classes on a given dataset. To illustrate this point, we simulate this situation based on the *KITTI* 3D detection dataset [31], and plot the histograms of each separate tasks (task-1 to task-5) and that of the whole dataset (task-1..5) in Figure 2 (right).

The inevitable data distribution differences in *TIL* make optimizing the knowledge-distillation loss not equivalent to optimizing the statistic risk for all seen tasks. It results in degradation of old-task performance when adding a new task, and the degradation is relative to the extend of the data distribution differences between the new task and the previously seen tasks. Indeed, it has been shown empirically in [5], [32] that the use of significantly different data distributions leads to a significant decrease in performance for the knowledge-distillation-based method *LwF* [4].

The degradation will be accumulated to build-up errors, so that it will eventually cause the *LED* problem when the length of the task sequence increases. In Figure 3 (a), the intersection of the low error regions of task-A and task-B denotes the low error region of *TIL*. It overlaps with the low error region of $\mathcal{L}_{KD}$ (white circle), but they do not completely coincide. When applying knowledge distillation to the third task, the low error region of $\mathcal{L}_{KD}$ will drift towards the task-C (Figure 3 (b)). In practice, it can be observed that the parametric model tends to over-fit the new task and the performance on task-A and task-B degrades.

C. A Bayesian solution to correct knowledge distillation

Since the data distribution differences among tasks lead to the *LED* problem in *TIL*, we intend to correct the objective function of knowledge-distillation (equation (4)) by adding a constraint related to the data distributions of all previously seen tasks. It can be achieved by maximizing the logarithm posterior probability $\log p(\boldsymbol{\theta}| \cup_{t=1}^{\mathcal{T}-1} \mathcal{D}^{(t)})$. Therefore, the corrected objective function of knowledge distillation is

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathcal{L}_{KD}(\boldsymbol{\theta}, \boldsymbol{\theta}^*_{1..\mathcal{T}-1}, \mathcal{D}^{\mathcal{T}}) - \beta \log p(\boldsymbol{\theta}| \cup_{t=1}^{\mathcal{T}-1} \mathcal{D}^{t}), \tag{7}$$

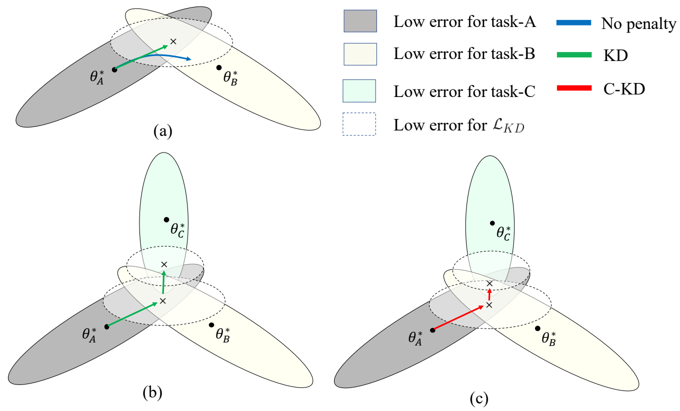

Fig. 3: Schematic illustration of knowledge distillation and our proposed corrected knowledge distillation (best viewed in color). (a) knowledge distillation preserves the task-A knowledge by constructing an alternative objective goal which restricts $p(\hat{\mathcal{Y}}|\mathcal{X};\boldsymbol{\theta})$ close to $p(\mathcal{Y}|\mathcal{X};\boldsymbol{\theta}_0)$; the low error region of $\mathcal{L}_{KD}$ is illustrated with a white circle. (b) When recursively applying $\mathcal{L}_{KD}$ to the third task, the constructed alternative objective function will drift towards the new task. (c) $\mathcal{L}_{C-KD}$ optimizes the parameters to the direction maximizing the posterior probability related to **all** seen tasks, which prevents the optimization process from over-fitting the new task.

where β is the weighting factor of the logarithm posterior term.

The evaluation of the term $\log p(\boldsymbol{\theta}| \cup_{t=1}^{\mathcal{T}-1} \mathcal{D}^{(t)})$ is challenging, since the data of previous tasks is intractable when training the task-$\mathcal{T}$. Here we adopt the Laplacian approximation in *EWC* [9] to evaluate this term. Its mechanism is to restore the prior information of old tasks, like *FIM* or its alternatives, and then use the priors to evaluate this logarithm posterior term for correcting the drift towards the new task. It can be written as

$$-\log p(\boldsymbol{\theta}| \cup_{t=1}^{\mathcal{T}-1} \mathcal{D}^{(t)}) \approx \frac{1}{2}\sum_{t=1}^{\mathcal{T}-1}\sum_{i} \beta_t \mathbb{F}_i^t ||\boldsymbol{\theta}_i - \boldsymbol{\theta}^*_{1..\mathcal{T}-1,i}||_2^2,$$
$$\mathbb{F}^t = \frac{1}{|S|}\sum_{\tilde{\mathcal{D}}_t \sim \mathcal{D}_t} [\frac{\partial}{\partial \boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \mathcal{D}^t)^T \frac{\partial}{\partial \boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \mathcal{D}^t)], \tag{8}$$

where β_t is the hyperparameter balancing the weight of each task-t, $\mathbb{F}_i^t$ is value of the i-th parameter in the diagonal of the *FIM* computed on task-t, and $|S|$ denotes the number of times $\tilde{\mathcal{D}}_t$ is sampled from $\mathcal{D}_t$. We provide an integrated derivation in our Supplementary Materials.

As a result, the corrected objective function of knowledge distillation for *TIL* is

$$\mathcal{L}_{C-KD} = \mathcal{L}_{KD} + \frac{1}{2}\sum_{t=1}^{\mathcal{T}-1}\sum_{i} \beta_t \mathbb{F}_i^t ||\boldsymbol{\theta}_i - \boldsymbol{\theta}^*_{1..\mathcal{T}-1,i}||_2^2. \tag{9}$$

Figure 3 (c) illustrates the effects of the corrected objective function. Optimizing the corrected objective function $\mathcal{L}_{C-KD}$ will lead the parameters to the direction maximizing posterior distribution related to **all** seen tasks, and prevent the optimization process from over-fitting the new task.

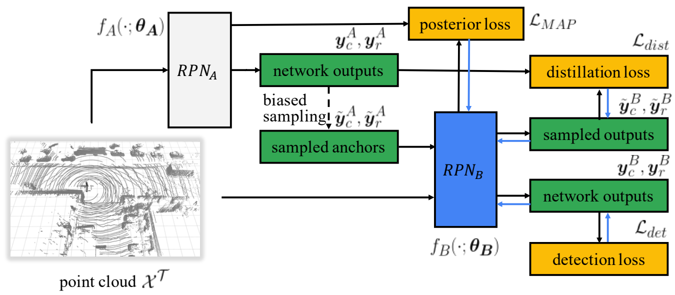

Fig. 4: Overview of the dual network learning framework. RPN_A is the well-trained optimal parametric model $f_A(\cdot, \boldsymbol{\theta}_A^*)$ for previous tasks, the parameters of which are frozen. RPN_B is the new-task parametric model $f_B(\cdot, \boldsymbol{\theta}_B)$, the parameters of which are initialized from $\boldsymbol{\theta}_A^*$. The blue arrows represent the backward propagation paths of optimization.

	trainable param.	initial lr	training steps per task	anno.	loss
fine-tuning	$\boldsymbol{\phi} \cup \boldsymbol{\varphi} \backslash \boldsymbol{\varphi}_A$	0.1γ	50 epochs	new	$\mathcal{L}_{det}$
joint training	$\boldsymbol{\theta}$	γ	50 epochs	all	$\mathcal{L}_{det}$
kd	$\boldsymbol{\theta}$	γ	50 epochs	new	$\mathcal{L}_{det}$ + $\mathcal{L}_{dist}$
ewc/mas	$\boldsymbol{\theta}$	γ	50 epochs	new	$\mathcal{L}_{det}$ + $\mathcal{L}_{MAP}$
incdet	$\boldsymbol{\theta}$	γ	50 epochs	new & pseudo	$\mathcal{L}_{det}$ + $\mathcal{L}_{MAP}$
c-kd	$\boldsymbol{\theta}$	γ	50 epochs	new	$\mathcal{L}_{det}$ + $\mathcal{L}_{dist}$ + $\mathcal{L}_{MAP}$

TABLE I: Comparison among different *TIL* schemes in our experiments.

IV. An Application in 3D Object Detection

In this section, we demonstrate a use case of our corrected knowledge distillation by applying it to a 3D object detection *Region Proposal Network (RPN)* under the *TIL* setting. We have reviewed the background of 3D object detection *RPNs* in Section II, and now we will state it in a more formal way. A 3D object detection *RPN* is a parametric model $f_{\boldsymbol{\theta}} = h_{\boldsymbol{\varphi}} \circ g_{\boldsymbol{\phi}}$, where $g_{\boldsymbol{\phi}}$ represents the feature extraction submodel, projecting the input into the embedding space $\mathbb{R}^F$, where F is the dimension of the embedding space, and $h_{\boldsymbol{\varphi}}$ represents the header of *RPN*. The $h_{\boldsymbol{\varphi}}$ consists of two parts: $h_{\boldsymbol{\varphi}_{cls}} : \mathbb{R}^F \rightarrow \mathbb{R}^{M \times A \times C}$ and $h_{\boldsymbol{\varphi}_{reg}} : \mathbb{R}^F \rightarrow \mathbb{R}^{M \times A \times S}$ conditioned on the anchor map $\boldsymbol{a} \in \mathbb{R}^{M \times A \times S}$ where M is the total number of locations, A is number of anchors on each location, C is the total number of classes, and S is the dimension of the parameterized anchor vector [1]. For simplicity, we denote $f_{cls} = h_{\boldsymbol{\varphi}_{cls}} \circ g_{\boldsymbol{\phi}}$ and $f_{reg} = h_{\boldsymbol{\varphi}_{reg}} \circ g_{\boldsymbol{\phi}}$ in the rest of the paper.

In 3D object detection, we write the corrected objective function (equation (9)) into the following form:

$$\begin{aligned}\mathcal{L}_{C-KD} = \mathcal{L}_{det}(\boldsymbol{\theta}, \mathcal{D}^{\mathcal{T}}) + \mathcal{L}_{dist}(\boldsymbol{\theta}, \boldsymbol{\theta}^*_{1..\mathcal{T}-1}, \mathcal{D}^{\mathcal{T}}) \\ + \mathcal{L}_{MAP}(\boldsymbol{\theta}, \mathbb{F}^1, ..., \mathbb{F}^{\mathcal{T}-1}, \boldsymbol{\theta}^*_{1..\mathcal{T}-1}, \mathcal{D}^{\mathcal{T}}),\end{aligned} \tag{10}$$

where $\mathcal{L}_{det}(\boldsymbol{\theta}|\mathcal{D}^{\mathcal{T}})$ denotes the likelihood of $\mathcal{D}^{\mathcal{T}}$ conditional on $\boldsymbol{\theta}$, $\mathcal{L}_{dist}(\boldsymbol{\theta}|\boldsymbol{\theta}^*_{1..\mathcal{T}-1}, \mathcal{D}^{\mathcal{T}})$ denotes the knowledge-distillation regularizer, and the last term denotes the logarithm posterior term.

To evaluate the three terms in equation (10), we adopt the dual network learning framework in [6]. We consider the step of incrementally training the task-$\mathcal{T}$, $\mathcal{T} > 1$, and denote the well-trained optimal parametric model for the previous tasks task-t, $t < \mathcal{T}$ as $f_A(\cdot;\boldsymbol{\theta}_A^*)$, while represent the current task-$\mathcal{T}$ as task-B and its parametric model as $f_B(\cdot;\boldsymbol{\theta}_B)$. Figure 4 demonstrates the dual network learning framework. We forward point clouds $\mathcal{X}^{\mathcal{T}}$ to $f_B(\cdot;\boldsymbol{\theta}_B)$ and evaluate the detection loss $\mathcal{L}_{det}$. In the RPN_A branch, we forward point clouds $\mathcal{X}^{\mathcal{T}}$ to $f_A(\cdot;\boldsymbol{\theta}_A^*)$, which is the optimal model for task-t, $t < \mathcal{T}$. We sample a foreground subset $\tilde{\boldsymbol{y}}_c^A, \tilde{\boldsymbol{y}}_r^A$ from the RPN_A outputs $\boldsymbol{y}_c^A, \boldsymbol{y}_r^A$ with a biased sampling strategy $\boldsymbol{y}_c^A = f_{A,cls}(\mathcal{X}^{\mathcal{T}}, \boldsymbol{\theta}_A^*), \boldsymbol{y}_r^A = f_{A,reg}(\mathcal{X}^{\mathcal{T}}, \boldsymbol{\theta}_A^*)$, and then find their corresponding estimations $\tilde{\boldsymbol{y}}_c^B, \tilde{\boldsymbol{y}}_r^B$ from the RPN_B outputs. As in *ILOD* [6], we compute the knowledge-distillation regularization term $\mathcal{L}_{dist}$ with

$$\mathcal{L}_{dist} = \ell_c(\tilde{\boldsymbol{y}}_c^A, \tilde{\boldsymbol{y}}_c^B) + \alpha \ell_r(\tilde{\boldsymbol{y}}_r^A, \tilde{\boldsymbol{y}}_r^B), \tag{11}$$

where ℓ_c and ℓ_r represent the distance measurement function for classification and bounding box regression, and α balances the weights of these two terms. The logarithm posterior term is evaluated with $\boldsymbol{\theta}_A^*$ and $\boldsymbol{\theta}_B$ as well as the data prior *FIMs*:

$$\mathcal{L}_{MAP} = \frac{1}{2} \sum_{t=1}^{\mathcal{T}-1} \sum_i \beta_t \mathbb{F}_t^i ||\boldsymbol{\theta}^*_{A,i} - \boldsymbol{\theta}_{B,i}||_2^2. \tag{12}$$

We can compute the *FIM* with equation (8) in a supervised method. There is also a heuristic computation method *MAS* [11], which provides an unsupervised way to approximate the *FIM* with

$$\begin{aligned}\mathbb{F}^t = & \frac{1}{|S|} \sum_{\tilde{\mathcal{D}}_t \sim \mathcal{D}_t} [(\frac{\partial}{\partial \boldsymbol{\theta}} ||f_{1..t}(\mathcal{X}^{\mathcal{T}}, \boldsymbol{\theta}^*_{1..t})||_2^2)^T \\ & \times \frac{\partial}{\partial \boldsymbol{\theta}} ||f_{1..t}(\mathcal{X}^{\mathcal{T}}, \boldsymbol{\theta}^*_{1..t})||_2^2],\end{aligned} \tag{13}$$

where the notation is the same as before. We consider both of these two computation methods in our experiments.

V. Experiments

In this section, we report the experimental results to demonstrate the effectiveness of our proposed corrected knowledge distillation for *TIL*. We first describe our implementation details and then demonstrate the performance of our proposed method in overcoming the catastrophic forgetting. Finally, we show its ability to solve the *LED* problem by comparing the results in the increment-at-once and the increment-in-sequence experiments.

A. Implementation details

We adopt *SECOND*[2] as our basic 3D object detector. It is an *RPN* for LiDAR-based 3D object detection. To simplify the

[1]For instance, if we discretize the 3D spaces into a 3D grid with the shape $[200, 200, 8]$, and put ten anchors on each location with representing each anchor with a vector of length seven, then $a \in \mathbb{R}^{200\times200\times8\times10\times7}$, where $M = 200 \times 200 \times 8$, $A = 10$, $S = 7$.

[2]https://github.com/traveller59/second.pytorch

description, we continue to use the notation in Section IV and denote the old task(s) as task-A and the new task(s) as task-B. We train task-A from scratch and consider the following *TIL* training schemes:

- *fine-tuning*: We freeze the old-task part of the header parameters $\boldsymbol{\varphi}_A \subseteq \boldsymbol{\varphi}$. The parameters of feature extractor $\boldsymbol{\phi}$ and the new-task part $\boldsymbol{\varphi} \backslash \boldsymbol{\varphi}_A$ of the header are trainable with the detection loss $\mathcal{L}_{det}$. We set the initial learning rate to 0.1γ to avoid drifting greatly from $\boldsymbol{\theta}_A^*$.
- *joint training*: We merge all the training data of seen tasks and use the annotations of all classes in training. Theoretically, this provides the upper-bound performance for old-task performance in *TIL*.
- *kd*: It is the implementation of knowledge distillation as *ILOD* [6]. We additionally consider its two variants: *kd (unbiased)* with the unbiased sampling strategy and *kd (threshold)* with the threshold sampling strategy as *Faster-ILOD* [8]. For more details about the sampling strategies, please refer to our Supplementary Materials.
- *ewc/mas*: It is the implementation of the data-prior-based methods *EWC* [9] and *MAS* [11].
- *incdet*: It is the implementation of *IncDet* [15].
- *c-kd*: It is the implementation of our proposed corrected knowledge distillation. We consider two cases: *c-kd (ewc)* and *c-kd (mas)*. *c-kd (ewc)* computes the *FIM* with equation (8), while *c-kd (mas)* approximates the *FIM* with equation (13).

We also list the differences among the *TIL* schemes in Table I. For more details about our implementation and hyperparameters, please refer to our Supplementary Materials.

Data: We use the dataset of *KITTI* 3D object detection benchmark [31], and consider two more classes ”Van” and ”Truck” to construct five tasks: task-1(Car), task-2(Pedestrian), task-3(Cyclist), task-4(Van), and task-5(Truck). The class within the brackets is the task-specific class. Each task is composed of its training set $D_{train}^{(t)} = \{(\mathcal{X}_{train,i}^{(t)}, \mathcal{Y}_{train,i}^{(t)})\}$ and a testing set $D_{test}^{(t)} = \{\mathcal{X}_{test,i}^{(t)}\}$. Every $\mathcal{X}_{train,i}^{(t)}$ and $\mathcal{X}_{test,i}^{(t)}$ contains at least one instance of the task-specific class. In consequence, all the tasks have different data distributions, annotation distributions and different classes, which coincides with the *TIL* definition. The statistical information about these tasks is available in our Supplementary Materials. In our experiments, the task-$(1..K)$ represents the merged tasks from task-1 to task-K. We merge tasks by gathering their training datasets and testing datasets. The task-specific class set of task-$(1..K)$ is the union of the tasks from task-1 to task-K, i.e., $C^{(1..K)} = \cup_{i=1...K} C^{(i)}$.

Evaluation metrics: We use the *3D average precision (AP)* to evaluate the detection results. The *intersection-over-union (IoU)* thresholds are 0.5 for Car, Van and Truck, and 0.25 for Pedestrian and Cyclist. There are three difficulty levels: easy ($\leq$ 20 m), moderate ($\leq$ 35 m) and hard ($\leq$ 50 m) according to the distances between the object and the ego vehicle as well as the occlusion, as in [31]. We compute the *mean 3D average precision (mAP)* to compare different cases:

$$mAP^{(t)} = \sum_{c \in C^{(t)}} \frac{N_c^{(t)}}{N^{(t)}} \{ \frac{1}{3} [AP_{c,easy}^{(t)} + AP_{c,mod.}^{(t)} + AP_{c,hard}^{(t)}] \}, \quad (14)$$

where $C^{(t)}$ denotes the class set of task-t, and the *mAP* is the weighted average of the *APs* in the three difficulty levels of task-t.

method	old(+)	forget(-)	new(+)	all(+)
A(1..2)	81.9	-	-	-
+B(3..5) *fine-tuning*	0.0	81.9	40.3	10.7
+B(3..5) *ewc*	0.0	81.9	39.8	10.5
+B(3..5) *mas*	0.0	81.9	39.1	10.7
+B(3..5) *kd (unbiased)*	0.0	81.9	34.7	9.2
+B(3..5) *kd (threshold)*	77.6	4.3	37.7	63.2
+B(3..5) *kd*	81.1	0.8	40.1	67.1
+B(3..5) *incdet*	80.8	1.1	30.3	64.0
+B(3..5) *c-kd (ewc)*	81.1	0.8	39.3	65.2
+B(3..5) *c-kd (mas)*	**81.9**	**0.0**	**40.6**	**67.6**
+B(3..5) *joint training*	81.5	0.4	24.1	63.1
A(1..5)	81.5	0.4	28.8	64.8

TABLE II: Evaluation results on testing set of the increment-at-once *TIL* experiment based on the *KITTI* dataset. The positive metrics ”old”, ”new” and ”all” columns represent the *mAP* computed on task-(1..2), task-(3..5) and task-(1..5); the negative metric ”forget” column represents the performance degradation relative to A(1..2).

B. Increment at once

In this experiment, we explore the *TIL* scenario to increment multiple tasks at once. We first train the 3D detector on task-(1..2) from scratch, an then incrementally train it on task-(3..5). The evaluation results are shown in Table II.

For old tasks, *fine-tuning* forgets all the old-task knowledge, which shows the manifestation of the *catastrophic forgetting*. The prior-based methods *ewc* and *mas* cannot prevent the detector from forgetting the old-task knowledge in detection tasks, which coincides with the findings in [15]. The knowledge-distillation-based method *kd* prevents the 3D detector from forgetting and performs better than its unbiased-sampling and threshold-sampling variants. Our corrected knowledge distillation methods *c-kd(ewc)* and *c-kd(mas)* perform better or comparable than the original case *kd*. It shows the effectiveness of knowledge-distillation-based methods in overcoming the the catastrophic forgetting.

For new tasks, all the *TIL* schemes trained with only new annotations (*fine-tuning*, *kd*, *ewc/mas*, and *c-kd*) result in much better performance than the cases trained with all or pseudo annotations (*joint training*, and *incdet*). We attribute this to the class imbalance of the dataset. In Figure 2, we compare the class histogram of the task-(3..5) and that of the task-(1..5). It demonstrates that the class imbalance is worse in the task-(1..5), which is used in *joint training* in this experiment. The class-imbalance situation of *incdet* is similar to the task-(1..5) according to the mechanism of pseudo annotations in *IncDet* [15].

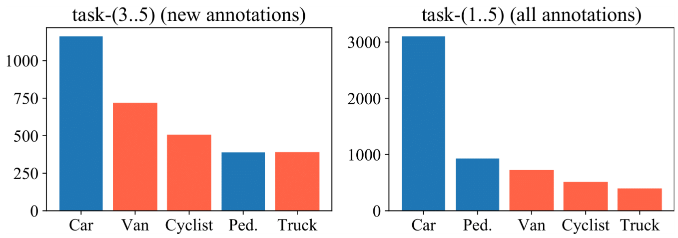

Fig. 5: Histograms of the training sets of task-(3..5) and task-(1..5). The y-axis represents the number of data samples. In the x-axis, "Ped." is for "Pedestrian". We highlight the task-specific classes of task-(3..5) with red to demonstrate that the class imbalance is worse in task-(1..5).

method	old(+)	forget(-)	new(+)	all(+)
A(1..2)	81.9	-	-	-
+B(3)(4)(5) *fine-tuning*	0.0	81.9	0.0	0.0
+B(3)(4)(5) *ewc*	0.0	81.9	0.9	0.2
+B(3)(4)(5) *mas*	0.0	81.9	0.8	0.1
+B(3)(4)(5) *kd (unbiased)*	0.0	81.9	0.3	0.1
+B(3)(4)(5) *kd (threshold)*	66.8	15.1	27.9	53.3
+B(3)(4)(5) *kd*	67.8	14.1	33.5	55.4
+B(3)(4)(5) *incdet*	75.3	6.6	25.9	58.7
+B(3)(4)(5) *c-kd (ewc)*	79.3	2.6	37.8	65.2
+B(3)(4)(5) *c-kd (mas)*	**81.8**	**0.1**	**39.2**	**66.8**
+B(3)(4)(5) *joint training*	83.8	-1.9	29.4	65.3
A(1..5)	81.5	0.4	28.8	64.8

TABLE III: Evaluation results on testing set of the increment-in-sequence *TIL* experiment based on the *KITTI* dataset.

C. Increment in sequence

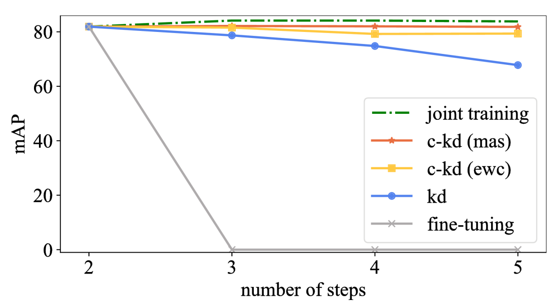

Fig. 6: Increment-in-sequence *mAP* curves of task-(1..2). The x-axis denotes the time step of evaluating the *mAP* on task-(1..2). For example, if the number of steps is three, its *mAP* value is the evaluation result after incrementally training on task-3.

We explore an alternative scenario of *TIL*: incrementing multiple tasks in sequence. To compare with the increment-at-once experiment in Section V-B, we first train the 3D object detector on task-(1..2) as before, and then incrementally train the detector on task-3, task-4, and task-5 in sequence. The evaluation results are shown in Table III [3].

[3] **+B(3)(4)(5)** ***joint training*** improves the performance of old-task a little, since its training iterations is three times as many as **+B(3..5)** ***joint training*** with all annotations.

We can observe the manifestation of *LED* problem by comparing **+B(3)(4)(5)** ***kd*** in Table III with **+B(3..5)** ***kd*** in Table II: 0.8 *mAP* $\leftrightarrow$ 14.1 *mAP* in the forget metric. In contrast, *c-kd (ewc)* and *c-kd (mas)* perform much better and more consistent: 0.8 *mAP* $\leftrightarrow$ 2.6 *mAP* as well as 0.0 *mAP* $\leftrightarrow$ 0.1 *mAP* in the forget metric. It demonstrates that our proposed corrected knowledge distillation method takes effect in solving the *LED* problem as we expected. We also plot the *mAP* curves of the old task during the whole *TIL* process in Figure 6 for a better comparison.

We also conducted the incremental-in-sequence based on the *NuScenes* dataset [33]. The same conclusion still holds, and please refer to our Supplementary Materials for more detail.

VI. CONCLUSION

In this paper, we attribute the *LED* problem to the inevitable data distribution differences in *TIL*. To solve this problem, we propose to correct the original knowledge distillation for *TIL* by additionally maximizing the posterior probability related to all previously seen tasks. We show its usefulness with an application in 3D object detection. The experimental results demonstrate its effectiveness. Our proposed method reaches the upper-bound performance, which is provided by joint training with all old data, in the *TIL* experiments based on the *KITTI* dataset.

As a result, the existing knowledge-distillation-based *TIL* methods will benefit from the proposed corrected knowledge distillation and prevent parametric models from forgetting knowledge even in the face of a long task sequence.

REFERENCES

[1] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.
[2] J. Knoblauch, H. Husain, and T. Diethe, "Optimal continual learning has perfect memory and is np-hard," *arXiv preprint arXiv:2006.05188*, 2020.
[3] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.
[4] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 12, pp. 2935–2947, 2018.
[5] A. Rannen, R. Aljundi, M. B. Blaschko, and T. Tuytelaars, "Encoder based lifelong learning," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 1329–1337.
[6] K. Shmelkov, C. Schmid, and K. Alahari, "Incremental learning of object detectors without catastrophic forgetting," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 3420–3429.
[7] D. Li, S. Tasci, S. Ghosh, J. Zhu, J. T. Zhang, and L. Heck, "Rilod: near real-time incremental learning for object detection at the edge," in *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*, 2019, pp. 113–126.
[8] C. Peng, K. Zhao, and B. Lovell, "Faster ilod: Incremental learning for object detectors based on faster rcnn," *arXiv preprint arXiv:2003.03901*, 2020.
[9] 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.
[10] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," *Proceedings of machine learning research*, vol. 70, p. 3987, 2017.
[11] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars, "Memory aware synapses: Learning what (not) to forget," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 139–154.

[12] S. W. Lee, J. H. Kim, J. Jun, J. W. Ha, and B. T. Zhang, "Overcoming catastrophic forgetting by incremental moment matching," in *Advances in neural information processing systems*, 2017, pp. 4652–4662.

[13] A. Chaudhry, P. K. Dokania, T. Ajanthan, and P. H. S. Torr, "Riemannian walk for incremental learning: Understanding forgetting and intransigence," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[14] X. Liu, M. Masana, L. Herranz, J. Van de Weijer, A. M. Lopez, and A. D. Bagdanov, "Rotate your networks: Better weight consolidation and less catastrophic forgetting," in *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, 2018, pp. 2262–2268.

[15] L. Liu, Z. Kuang, Y. Chen, J. Xue, W. Yang, and W. Zhang, "Incdet: In defense of elastic weight consolidation for incremental object detection," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2020.

[16] J. Beltrán, C. Guindel, F. M. Moreno, D. Cruzado, F. Garcia, and A. De La Escalera, "Birdnet: A 3d object detection framework from lidar information," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Nov 2018, pp. 3517–3523.

[17] M. Simon, S. Milz, K. Amende, and H. M. Gross, "Complex-yolo: An euler-region-proposal for real-time 3d object detection on point clouds," in *European Conference on Computer Vision*. Springer, 2018, pp. 197–209.

[18] B. Yang, W. Luo, and R. Urtasun, "Pixor: Real-time 3d object detection from point clouds," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 7652–7660.

[19] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 12 689–12 697.

[20] B. Li, "3d fully convolutional network for vehicle detection in point cloud," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 1513–1518.

[21] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4490–4499.

[22] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018.

[23] B. Zhu, Z. Jiang, X. Zhou, Z. Li, and G. Yu, "Class-balanced Grouping and Sampling for Point Cloud 3D Object Detection," *arXiv e-prints*, p. arXiv:1908.09492, Aug 2019.

[24] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 77–85.

[25] R. Q. Charles, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in neural information processing systems*, 2017, pp. 5099–5108.

[26] J. Li, B. M. Chen, and G. H. Lee, "So-net: Self-organizing network for point cloud analysis," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9397–9406.

[27] H. Su, V. Jampani, D. Sun, S. Maji, E. Kalogerakis, M. Yang, and J. Kautz, "Splatnet: Sparse lattice networks for point cloud processing," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2530–2539.

[28] S. Q. Ren, K. M. He, G. Ross, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[29] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "Continual learning: A comparative study on how to defy forgetting in classification tasks," *arXiv preprint arXiv:1909.08383*, vol. 2, no. 6, 2019.

[30] S. Song, S. P. Lichtenberg, and J. Xiao, "Sun rgb-d: A rgb-d scene understanding benchmark suite," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 567–576.

[31] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3354–3361.

[32] R. Aljundi, P. Chakravarty, and T. Tuytelaars, "Expert gate: Lifelong learning with a network of experts," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 7120–7129.

[33] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020, pp. 11 618–11 628.