Confidence Calibration for Incremental Learning

DONGMIN KANG, YEONSIK JO©, YEONGWOO NAM, AND JONGHYUN CHOI©, (Member, IEEE)
Gwangju Institute of Science and Technology (GIST), Gwangju 61005, South Korea

Corresponding author: Jonghyun Choi (jhc@gist.ac.kr)

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2019-0-01351, Development of Ultra Low-Power Mobile Deep Learning Semiconductor With Compression/Decompression of Activation/Kernel Data).

ABSTRACT Class incremental learning is an online learning paradigm wherein the classes to be recognized are gradually increased with limited memory, storing only a partial set of examples of past tasks. At a task transition, we observe an unintentional imbalance of confidence or likelihood between the classes of the past and the new task. We argue that the imbalance aggravates a catastrophic forgetting for class incremental learning. We propose a simple yet effective learning objective to balance the confidence of classes of old tasks and new task in the class incremental learning setup. In addition, we compare various sample memory configuring strategies and propose a novel sample memory management policy to alleviate the forgetting further. The proposed method outperforms the state of the arts in many evaluation metrics including accuracy and forgetting $F$ by a large margin (up to 5.71% in $A_{10}$ and 17.1% in $F_{10}$) in extensive empirical validations on multiple visual recognition datasets such as CIFAR100, TinyImageNet and a subset of the ImageNet.

INDEX TERMS Incremental learning, continual learning, confidence calibration.

I. INTRODUCTION

We consider the problem of ‘incremental learning (IL)’, which is a continual learning paradigm that the model is expected to learn a set of tasks and infer the sample’s label without knowing its originating task. Specifically, we consider a classification problem where new sets of classes are added incrementally with a budgeted memory storing the training examples of past tasks, referred to as ‘class-incremental learning’ [1]. Same as other continual learning set-ups, the IL algorithms suffer from the stability-plasticity dilemma [2]. This dilemma is often addressed by two contradicting issues separately; catastrophic forgetting of the old knowledge and intransigence on learning new knowledge [3]. By the success of deep neural networks in machine learning [4]–[6], many IL algorithms are based on deep neural networks [3], [7]–[9]. Unfortunately, training a neural network is known to suffer from the catastrophic forgetting not even in the incremental scenario [10], [11] and the forgetting problem worsens in the class incremental learning scenarios.

The limited memory resource in the IL set-up, which we name as a ‘sample memory’ (also called as a ‘representative memory’ [9]), is used to alleviate forgetting [12]. But the budget limitation causes two critical problems for supervised learning: 1) class imbalance between the classes in the previous tasks and the new task and 2) lack of optimal sampling strategy. The class imbalance problem leads a model to overfitting to the new task, and thus, makes examples in the past tasks prone to be inferred as a new task’s example. The sampling policy is less explored in the literature.

Investigating the class imbalance problem and the sub-optimal sampling strategy, we observe two phenomena; 1) the predicted confidence or likelihood of an example at the task transition has an artificial offset in the new tasks due to the sample memory, and 2) the most of sampling strategy is marginally better than the uniform random sampling. Based on the observations, we propose 1) a novel method to calibrate the inferred confidence to alleviate forgetting of the IL models by balancing the likelihoods of past and the new task’s classes and 2) a novel sample memory management policy using a proposed scoring function based on cumulative likelihood over the epochs. In empirical validations, we show that the proposed method outperforms state-of-the-art IL algorithms in various metrics including average accuracy ($A_{avg}$) at final task transition, the accuracy of the final model ($A_{10}$) and forgetting ($F_{10}$) by a non-trivial margin (up to 5.71% in $A_{10}$ and 17.1% in $F_{10}$).

A. RELATED WORK

1) LEARNING SEQUENCE OF TASKS

A machine learning paradigm that a model is trained on a sequence of tasks has been referred to as various names such
as \{continuous, continual, sequential, incremental and lifelong\} learning. As these terms are not precisely defined, each work has its own learning scenarios and constraints. But all these paradigms face a common problem; stability-plasticity dilemma \cite{13} which refers to contradicting difficulties for a model to not only learn a new task but also keep the past knowledge since the statistics of the samples continuously change. Especially, forgetting past knowledge, referred to as catastrophic interference or forgetting \cite{10}, \cite{11}, is exacerbated by the usage of deep neural networks. Among the strategies summarized in \cite{1}, we are particularly interested in continual learning with a memory of previous tasks and no task identity given at the inference, which is referred to as ‘incremental learning’ in \cite{12}. Please refer to \cite{14} for a comprehensive review.

**Incremental learning (IL)** is a sequential learning paradigm with limited memory resources \cite{7}, \cite{9}, \cite{12}. There are many proposals about memory managing strategies \cite{3}, \cite{7}, \cite{9}, a better regularizing strategy \cite{8}, \cite{15}–\cite{19}, and using an external memory with unlabeled data \cite{20} and a method robust to the order of given tasks with task identity given \cite{21}. We will review the ones addressing the forgetting and the sample memory management policy in detail.

**Catastrophic forgetting or interference** has been well-studied but is still an open problem in neural networks \cite{16}, \cite{18}, \cite{19}, \cite{22}–\cite{27}. The catastrophic forgetting is aggravated in the incremental learning set-up \cite{10}, \cite{11}. There are two problem set-ups in the incremental learning. We briefly reviewed how previous works alleviate the catastrophic forgetting in both set-ups as follows.

The first set-up is the class incremental learning that the task is unknown during inference. In this set-up, recent works \cite{3}, \cite{7}, \cite{9} use previous tasks’ samples that contain the knowledge of previous tasks. iCaRL \cite{7} and end-to-end incremental learning (E2E) \cite{9} use knowledge distillation \cite{28} to preserve the knowledge of previous tasks. Chaudhry et al. \cite{3} uses elastic weight consolidation (EWC) \cite{8}. Our proposed method falls into this problem set-up.

The other set-up is that the task identity is given during inference \cite{8}, \cite{15}, \cite{29}–\cite{32}. Kirkpatrick et al. propose to use a Fisher information matrix (FIM) to weigh $l_2$ distances between the parameters learned in the past and the new task for knowledge transfer \cite{8}. Inspired by \cite{8}, several methods based on parameter regularization were presented (e.g., SI \cite{15}, MAS \cite{31}, HAT \cite{32}, GEM \cite{29}, VCL \cite{30}).

### 2) SAMPLE MEMORY MANAGEMENT

As sample memory management policy affects accuracy, there are a few strategies proposed in the literature \cite{3}, \cite{7}, \cite{9}. Interestingly, many proposals show marginal accuracy improvement over uniform sampling despite the computational complexity \cite{3}, \cite{7}, \cite{9}. These methods include Herding selection \cite{33} used in \cite{7}, sampling proportional to a histogram of each sample’s distance to the class mean \cite{9}, using a distance to decision boundary, and entropy of output softmax distribution \cite{3}. Recently, generative models are employed to generate past task samples \cite{34}–\cite{37} instead of sampling. The sample generation strategy for the IL is an active research topic and show promising results in relatively straightforward experimental validation (e.g., on MNIST and SVHN). But on these datasets, sampling from the uniform distribution already achieves saturated accuracy \cite{3} and there is no promising results reported in challenging datasets yet. In addition, due to the difference in number of samples belonging to the past and the new tasks, we argue that the class imbalance problem exacerbates the overfitting on the new tasks, which leads to accuracy drop. The class imbalance problem, however, is relatively less explored in the IL context \cite{9}, \cite{38}, \cite{39}. Ditzler et al. propose a dynamically-weighted consult and vote (DW-CAV) algorithm for an ensemble of classifiers for better plasticity \cite{39}. Pang et al. use the weighted-LPSVM for the class imbalance for IL \cite{38}.

### 3) CONFIDENCE CALIBRATION IN NEURAL NETWORK

Guo et al. show that a high capacity neural network (e.g., ResNet) exhibits overconfidence which leads to accuracy drop in the ordinary learning set-up and propose multiple logit calibration methods \cite{40}. We observe overconfidence in the classes of the new task in the IL setup and propose to calibrate the confidence. Pereyra et al. use a maximum entropy prior in logits as a regularizer for overly confident output distribution \cite{41}. Kuleshov and Ermon propose a calibration method in the online learning set-up particularly when the inputs can come from a potentially adversarial source \cite{42}, whose objective is clearly different from our proposal addressing the artificial overconfidence on the set of new classes.

### 4) EVALUATION METRICS FOR INCREMENTAL LEARNING

Average accuracy of incrementally trained classification model at all task transitions is the most widely used evaluation metric \cite{9}, \cite{20}. The average may not include the accuracy of the model at the first task as the model on the first task is not yet incrementally trained \cite{9}, \cite{20}. The accuracy at the last task has also been used \cite{3}. There are a number of measures for evaluating the amount of forgetting in the IL models including Backward transfer, Forward transfer, Remembering, measures for the efficiency of IL models \cite{3}, \cite{29}, \cite{43}. The Backward transfer (BWT) captures the influence of learning new task on the previous task, which can be translated as a measure of forgetting \cite{29}. The BWT averages accuracy drops at all task transitions. As the BWT can be negative, \cite{44} further propose variants of BWT; Positive Backward Transfer ($\max(BWT, 0)$) and Remembering ($1 - \min(BWT, 0)$). Chaudhry et al. proposed a measure for forgetting ($F$) which computes the accuracy drops between the best of all tasks and the accuracy at the current task \cite{3}. There are also some measures for the model’s ability to learn new knowledge; Forward Transfer (FWT) \cite{29}, \cite{43}
Each task is a set of classes to be recognized. On a set truth label and which can be translated as forgetting of the past knowledge the model less generalizable for the classes in the past tasks, from the past tasks and the samples of the new task. It makes of the new task. Despite the forgetting prevention mecha-

tions including fine-tuning or knowledge distillation [28], the model tends to overfit to the set of the small samples from the past tasks and the samples of the new task. It makes the model less generalizable for the classes in the past tasks, which can be translated as forgetting of the past knowledge but rather promotes the model to classify to the classes in the new task by the increased confidence of the classes in the new task in the inferred likelihood \( \hat{y} \). The artificially high confidence in inference for the classes in the new task results in low classification accuracy especially in the past tasks and eventually leads to low overall accuracy. Figure 1-(a) compares the average of logits on old and new classes of all test samples at the last task (Baseline) and figure 1-(b) shows the corresponding confusion matrices. We observe that the average of logits of new classes is much higher than that of old classes (Baseline) (Figure 1-(a)) and this imbalance causes artificially higher confusion of the old classes to the new classes (Figure 1-(b)).

1) LOSS FUNCTION TO COMPENSATE FOR THE OVERCONFIDENCE

To compensate the unexpected logit differences between the past tasks and the new task, we propose a calibrated loss that adds the difference of the logits of the classes of the past tasks (\( \mathcal{T}_p^k \)) and the classes of the new task (\( t_k \)) to the logits of new classes. We call this addition as ‘Logit Balancing’. It differs from [45] that requires held-out set to optimize the bias correction layer because we calibrate the logits during without requiring held-out set in training.

Formally, the classification loss \( L_C(D; \theta^k) \) with the logit balancing is written as a form of modified cross entropy loss:

\[
L_C(D; \theta^k) = -\sum_i \log \sigma'(z)_{iy_i}
\]

\[
= -\sum_i \log \left( \frac{\gamma \cdot \mathbf{1}(y_i \in t^k) e^{z_{iy_i}}}{\sum_{j \in \mathcal{T}_p^k} e^{z_{ij}} + \gamma \sum_{j \in t^k \neq y_i} e^{z_{ij}}} \right).
\]

where \( \sigma'(\cdot) \) is our modified softmax function with the logit balancing and the \( \gamma \) is the balancing parameter. We compute and update the \( \gamma \) in each epoch of training by the difference between the average logit of the classes in \( \mathcal{T}_p^k \) and the classes in \( t_k \) as follows:

\[
\gamma = e^{\alpha - \beta}, \quad \text{where}
\]

\[
\alpha = \frac{1}{|\mathcal{D}_k|} \sum_i \left( \frac{1}{|t_k| - 1} \sum_{j \neq y_i \in t_k} z_{ij} \right),
\]

\[
\beta = \frac{1}{|\mathcal{D}_k|} \sum_i \left( \frac{1}{|\mathcal{T}_p^k| - 1} \sum_{j \neq y_i \in \mathcal{T}_p^k} z_{ij} \right).
\]

The \( \alpha \) and \( \beta \) are averaged logits of current task and the previous tasks, respectively, except the logit of the ground truth class which is not part of abnormal logit. \( \mathcal{D}_k = \{(x_i, y_i) | (x_i, y_i) \in t_k \} \) denotes a training set at \( k \)-th task, we train the model together with sample memory at \( k \)-th task \( \mathcal{M}_k \), \( |\mathcal{D}_k| \) is its cardinality. In computing the \( \alpha \) and the \( \beta \), we average the logits of old and new task respectively to compute ‘overall’ logit difference between old and new task. In particular for the difference statistic, we use \( \alpha - \beta \) over a ratio \( \alpha/\beta \) for numerical stability in learning since \( \beta \) is often close to zero. When \( \gamma > 1 \), the \( \gamma \) multiplied logit increases the loss of new classes thus the gradient magnitude of the new classes increases. The larger gradient penalizes the over-confidence by back-propagation and reduces artificially high confidence in the new classes. When \( \gamma < 1 \), in contrast, the \( \gamma \) multiplied logit increases loss of old classes and penalizes
the old classes. When $\gamma = 0$, the modified cross entropy is identical to the original cross entropy. Note that the proposed logit balancing can be combined with any methods that use a vanilla cross-entropy loss for $\mathcal{L}_C(\cdot)$.

We quantify the overconfidence on the new task by Expected Calibration Error (ECE) [46] as in [40]. In Figure 2, we visualize the overconfidence by the Reliability diagram [47] and the ECE of three models (refer to the caption for details). As shown in the figure, the E2E model [9], which uses an advanced forgetting prevention mechanism, and the baseline exhibit the overconfidence indicated by large gaps and high ECE value. On the other hand, our proposal reduces both the gap and the ECE value. Note that the ECE of ours is not zero. It is because our calibration objective is to predict the correct task while the ECE quantifies the difference between class accuracy and predicted confidence. Nevertheless, the significantly reduced ECE value shows that the proposed ‘logit balancing’ method could reduce the calibration error resulted from the overconfidence.

**B. SAMPLE MEMORY MANAGING POLICY**

Following the notion of ‘representative memory’ [9] that resembles human forgetting procedure, we define $\mathcal{M}_k$ as the ‘representative memory’ at task $t_k$ and call it as ‘sample memory’ for simplicity. At the first task transition ($1 \rightarrow 2$, the beginning of $t_2$), examples in the previous tasks ($T^2_p$) are sampled to the sample memory by a policy. Then, at the later task transition ($k \rightarrow k + 1$, where $k > 1$), the examples of $T^k_p$ in the sample memory should be deducted and the examples in $T^k$ are added by a policy while maintaining the size of memory fixed. For better maintaining the past knowledge with the fixed budget, we need a good policy to add or deduct the examples. However, most of the proposed sampling strategies are either uniform random sampling or complicated but marginally better than the uniform random [3], [7], [9]. We argue that the confidence or likelihood of an example could be a measure of knowledge about the class distribution; if an example is highly confident, the example is the most representative (as it is most frequent) about the distribution of its class.

The likelihood at each epoch of learning however, also depends on the learning progress; for instance, at the end of learning of a task, majority of samples have likelihood around 1.0 unless the network underfits. To alleviate the effect of training progresses in extracting the knowledge of the class distribution by the inferred likelihood, we propose to average the likelihoods over the epochs (called cumulative likelihood, denoted as $\xi_i$) and use it as a score for sampling. Formally, $\xi_i$
FIGURE 3. Normalized histogram with respect to the cumulative likelihood of samples with image examples. The examples with high cumulative likelihood exhibits clear shape information about the class and vice versa.

FIGURE 4. More examples in CIFAR-100 with various cumulative likelihood scores. The examples with high cumulative likelihood shows clear shape information about the class and vice versa.

for $i^{th}$ sample’s cumulative likelihood is written as:

$$\xi_i = \frac{1}{E} \sum_{e=1}^{E} \sigma(z_{iy})^{e},$$

(4)

where $E$ is number of epochs in training a model for a task and $\sigma(z_{iy})^{e}$ is a likelihood at epoch $e$. Note that $\sigma(\cdot)$ is a vanilla softmax function, not a modified version proposed in Eq. 2.

Figure 3 shows the histogram of the cumulative likelihood with a few examples shown. We argue that those examples well represent the knowledge about the class (e.g., the shape of tree-class is clearly shown in the examples with high $\xi_i$) and vice versa.

Figure 4 shows more examples; images in CIFAR-100 with various cumulative likelihood scores. Similar to the examples in Figure 3, we observe that the examples with a high score (cumulative likelihood) are ones that have representative shapes of the class they belong to. And the images with the low score exhibit the class-oriented appearance in a fraction of the images (e.g., sunflower, and mushroom) or not very visible (e.g., bus).

Toneva et al. claim that the ‘unforgettable’ examples are not very useful during training [22]. Following their claim, we propose to discard those highly ‘unforgettable’ samples (particularly top 20% of training set sorted by the $\xi_i$ in descending order) and randomly sample the examples from the rest. At the next task transition, say $t_{k+1}$, we discard the $\lfloor M_k \rfloor \cdot |t_k| \cdot \frac{(k-1)}{2}$ examples with the lowest score for each class in the sample memory (the number of classes is $(k-1) \cdot |t_k|$) due to the fixed memory budget, where $|M_k|$ denotes the memory budget, $|t_k|$ denotes the number of classes in task $t_k$, and add examples belonging to the new tasks. In the removal or deduction process of samples for each class, we first discard samples of the top 20% ‘unforgettable’ samples. Among the remainder, we discard all samples scoring lower than top $\eta = \lfloor M_k \rfloor \cdot \frac{(k-1)}{2}$ samples. We then randomly select $\lfloor M_k \rfloor \cdot |t_k|$ samples among $\eta$. We illustrate the proposed sample memory managing policy in Figure 5.

In Figure 6, we empirically show that our scoring function performs better than the scoring functions used in the state of the art methods. Interestingly, reversing the score $(1-\xi_i)$ results in the worst selection policy, worse than random. It implies that our scoring function measures the worth of samples to be kept in the memory.
C. EVALUATION METRICS

The $F$ metric [3] and the $BWT$ [29] have been proposed to measure the forgetting. However, we argue that the $I$ metric [3] that is defined for the ‘single head configuration’ results in artificially high $I$ in more forgettable models because new task samples are barely confused to the old task as shown in ‘Baseline’ of Figure 1-(a). To exclude the effect of forgetting in the $I$ measure, we propose to compare $I_{10}$ evaluated in a multi-head manner, which confines the accuracy computation only on the classes of the new task. At the first glance, it may sound unreasonable to use the multi-head based metric for single head configuration. But we argue that even though the model is trained in a single head configuration, the intransigence in a multi-head configuration is a helpful metric for the model’s plasticity or learnability of new knowledge without the effect of forgetting included because the model classifies the sample only among the classes belonging to the given task in a multi-head configuration. Thus, we additionally report the $I$ at task $t_k$ on the multi-head configuration [3] denoted by $I_{k}^{mh}$ in all our experimental validations.

III. EXPERIMENTS

A. EXPERIMENTAL SET-UP

1) DATASETS

For experimental validations, we use CIFAR100 [48] as it is the most popular benchmark dataset for the IL algorithms (also called iCIFAR100) [3], [7], [9], [20], TinyImageNet [49] used in [50], and a subset of ImageNet (ILSVRC 2012) dataset [51] with 100 classes used in [7], [9], which we call as ImageNet-100.

The CIFAR100 consists of 60,000 images with a size of 32 × 32 pixels in 100 object categories. Among 60,000, 50,000 images are used for training a model and 10,000 images are used for testing.

The TinyImageNet consists of 120,000 images with a size of 64 × 64 pixels in 200 object categories. Among the 120,000 images, 100,000 are used for training, 10,000 for validation, and 10,000 for testing. Note that we opt to use the validation set when testing instead of submitting predicted labels on the test-set to the evaluation server held by Stanford.

For the ImageNet-100 dataset, we used 1,300 training samples and 50 verification samples from each class, following [7], [9]. Note that the random 100 classes of our subset would be different from [7], [9]. We resize (aspect-ratio preserving) and crop to make all images to be the size of 224 × 224 pixels.

2) TASK CONFIGURATION

Following a standard task configuration [7], [9], we define 10 tasks where each task consists of 10 classes (CIFAR100 and ImageNet-100) or 20 classes (TinyImageNet). Since the task configuration should be blind to the incremental learning algorithm, we report the average and standard deviation of three different random task configurations, following [3], [9].

3) SAMPLE MEMORY SIZE

We choose 2,000 for the experiments on CIFAR100 following [7] and choose 4,000, 5,200 for experiments on TinyImageNet and ImageNet-100, respectively (the same relative size considering the training set size).

4) EVALUATION METRICS

For the evaluation metrics, we use average accuracy ($A_{avg}$) in all task transitions (excluding the first task’s accuracy as it is not incrementally learned [9], [20]), final model’s accuracy ($A_k$, where $k$ is the final or last task. It is denoted as ‘Average’ in [44]), forgetting ($F_k$), intransigence ($I_k$) and multi-head intransigence ($I_{k}^{mh}$) metrics at task $t_k$ proposed by [3]. Note that some previous works include the first task’s accuracy in the average accuracy [7].

5) MODEL ARCHITECTURES AND TRAINING DETAILS

We use the ResNet-32 [52] as a backbone for the experiments on CIFAR100 and TinyImageNet with input size modification and ResNet-18 [52] for the experiments on ImageNet-100. We used images with the size of 32 × 32, 64 × 64 and 224 × 224 for CIFAR 100 dataset, TinyImageNet and ILSVRC 2012, each. For each incremental step, we train a network for 80 epochs. Learning rate starts at 0.1, and is divided by 10 every 20 epochs. We train the networks using stochastic gradient descent with mini-batches of 128 samples, weight decay of $10^{-4}$, and momentum of 0.9. As data augmentation, we use flipping and random crop in all experiments.

B. IMPLEMENTATION DETAILS

We compare our method with a baseline and two state of the arts. The baseline is a vanilla IL algorithm with a uniform sampling policy for the sample memory management and performs fine-tuning for $L_{p}(\cdot)$. The two state of the arts are our extension of EWC [8] with the sample memory, denoted as EWC+SM, and the E2E [9]. We evaluate the methods by various metrics: average accuracy ($A_{avg}$), accuracy at the final (10-th) task ($A_{10}$), Forgetting ($F_{10}$) and Intransigence ($I_{t10}$).

We apply the data-augmentation technique (DA) on CIFAR100 dataset, following [9]. Table 1 shows that the DA does not harm the performance gain of our methods to the other state of the arts in all evaluation metrics.

We did not compare with the continual learning methods that require to identify the task for inference such as SI [15], MAS [31], HAT [32], GEM [29] and VCL [30] since this setup is less challenging than our setup that does not require any information. Note that the results of EWC+SM are not directly comparable to those reported in [3] since we use an advanced backbone network architecture (ResNet-32) and budgeted sample memory.

C. COMPARISON WITH STATE OF THE ARTS

Our proposed method outperforms the state of the arts by large margins in all metrics except the $I_{10}$, as shown in Table 1.
TABLE 1. Comparison with state-of-the-art methods on CIFAR100 dataset. The E2E refers to E2E incremental learning [9]. The (DA) denotes usage of data augmentation scheme used in [9] to ensure that our reproduction of E2E method is valid. ‘∼’ refers to the inferred value from graphs in the paper. ‘Ours w/o x’ refers to the ablated models of our proposed method. For x, KD refers to knowledge distillation [28], SM refers to sample memory and LB refers to the proposed ‘Logit Balancing’ [Abl.] denotes the ablation study. ↑ indicates higher the better, ↓ vice versa.

| Unit (%) | $A_{avg}$ ↑ | $A_{10}$ ↑ | $F_{10}$ ↓ | $I_{10}$ ↓ | $I_{10}^{mh}$ ↓ |
|----------|--------------|-------------|----------|------------|----------------|
| Baseline | 50.25 ± 1.53 | 34.17 ± 0.66 | 54.31 ± 0.69 | -19.13 ± 1.17 | 3.37 ± 0.84 |
| EWC+SM   | 50.76 ± 0.84 | 34.77 ± 0.49 | 53.69 ± 0.90 | -19.13 ± 1.80 | 3.30 ± 1.11 |
| E2E [9]  | 58.1 (best)  | ~ 45        | N/A       | N/A         | N/A            |
| Ours     | 59.12 ± 2.98 | 48.87 ± 2.14 | 17.28 ± 0.15 | 24.40 ± 6.31 | 6.77 ± 2.00   |

Ablated

| Unit (%) | $A_{avg}$ ↑ | $A_{10}$ ↑ | $F_{10}$ ↓ | $I_{10}$ ↓ | $I_{10}^{mh}$ ↓ |
|----------|--------------|-------------|----------|------------|----------------|
| Ours − KD      | 58.43 ± 0.76 | 47.53 ± 0.48 | 16.52 ± 0.74 | 18.77 ± 5.49   | 3.07 ± 0.12   |
| Ours − SM      | 56.99 ± 2.10 | 45.56 ± 1.05 | 21.47 ± 0.39 | 22.67 ± 3.19   | 7.83 ± 1.07   |
| Ours − LB      | 56.39 ± 2.50 | 44.95 ± 1.72 | 31.27 ± 0.37 | -2.13 ± 1.40   | 8.10 ± 2.11   |

TABLE 2. Comparison with state-of-the-art methods on TinyImageNet (top) and ImageNet-100 (Top-5) (bottom) dataset. ↑ indicates higher the better, ↓ vice versa. Note that DA is not applied for all experiments in the table. Best results are in bold.

| Unit (%) | $A_{avg}$ ↑ | $A_{10}$ ↑ | $F_{10}$ ↓ | $I_{10}$ ↓ | $I_{10}^{mh}$ ↓ |
|----------|--------------|-------------|----------|------------|----------------|
| TinyImageNet: |                |              |          |            |                |
| Baseline | 37.69 ± 1.86 | 25.12 ± 0.74 | 44.83 ± 1.62 | -12.33 ± 5.12 | 3.47 ± 1.32   |
| EWC+SM  | 38.70 ± 1.42 | 26.06 ± 1.34 | 44.99 ± 1.81 | -12.77 ± 4.40 | 3.13 ± 2.50   |
| E2E     | 44.41 ± 1.76 | 34.21 ± 0.27 | 23.15 ± 0.89 | 4.80 ± 3.72   | 11.27 ± 4.09  |
| Ours    | 48.71 ± 1.73 | 39.81 ± 0.52 | 11.23 ± 1.97 | 21.67 ± 3.91  | 8.50 ± 1.82   |

| Unit (%) | $A_{avg}$ ↑ | $A_{10}$ ↑ | $F_{10}$ ↓ | $I_{10}$ ↓ | $I_{10}^{mh}$ ↓ |
|----------|--------------|-------------|----------|------------|----------------|
| ImageNet:100 |                |              |          |            |                |
| Baseline | 86.37 ± 0.47 | 77.33 ± 0.79 | 23.13 ± 0.69 | -3.07 ± 1.10 | 0.20 ± 0.35   |
| EWC+SM  | 85.77 ± 1.33 | 76.86 ± 1.01 | 23.63 ± 1.34 | -3.00 ± 0.72 | 0.40 ± 0.20   |
| E2E     | 89.63 ± 0.54 | 83.00 ± 1.40 | 14.01 ± 0.47 | -1.20 ± 1.06 | 0.53 ± 0.23   |
| Ours    | 90.57 ± 0.66 | 85.55 ± 1.05 | 9.50 ± 0.40 | 1.73 ± 0.50  | 0.60 ± 0.53   |

Recently, Lee et al. [20] proposed the ‘global distillation’ objective for incremental learning. However, the experimental set-up is significantly different from our set-up. In particular, to use a global distillation objective, it is necessary to use additional neural network which is specialized to the current task. To compare [20] to ours method, we modified followings in the experimental set-up. 1) As they use a large stream of external data while we do not, we excluded the external data to isolate the effect of the proposed loss function. 2) we changed their network backbone (Wide-ResNet [53]) to ResNet [52] which is the same architecture that we use. We show the comparison results in Table 3. Despite the less neural network capacity, ours outperforms [20] in $A_{avg}$, $A_{10}$, and $F_{10}$. Ours shows relatively higher intransigence ($I_{10}$), but we argue that Lee et al. shows better intransigence ($I_{10}$) at the expense of severe forgetting on the previous task.

D. ABLATION STUDY

On CIFAR100 dataset, we conduct ablation studies to show the contribution of each component of our proposal; without knowledge distillation (‘Ours w/o KD’), without the proposed sample memory management policy (‘Ours w/o SM’), without the proposed logit balancing method (‘Ours w/o LB’). Interestingly, ‘Ours w/o KD’ improves forgetting and intransigence but drops accuracy over the full model. It is because the proposed logit-balancing compensates the
benefit of the knowledge distillation (KD) thus removing the KD results in marginal accuracy drop (within the standard deviation of $F_{10}$). Note that the logit balancing significantly improves the forgetting (by 13.99%: 31.38% → 17.28%), and the sample memory management policy also has a significant contribution to them (by 4.19%: 21.47% → 17.28%). However, the logic balancing sacrifices the $I_{10}$ heavily because of the forgetting followed by the overconfidence. However, in $I_{10}^{nh}$ (the intransigence without the effect of the overconfidence), the logit balancing increases intransigence though marginally.

### E. EVALUATION ON VARIOUS TASK CONFIGURATIONS

Yoon et al. [54] emphasized the configurations (e.g., size and order) of of tasks in continual learning affects a lot on the performance of the model. Hence we consider about various task configurations. We compare ours to the E2E on various task sizes in Table 4. Note that ours consistently outperforms the E2E on various task size. We also evaluate our method on task configuration that tasks are remarkably different, and classes in a task are confusing each other (Distinctive-10) in Tab.4. To make a distinctive-10 set-up, we use the super-class definition of 5 classes in CIFAR100; we merge two super-classes into a task and conduct experiments. Ours outperforms E2E with the similar margin in Task size 10, but the accuracy drops by $\sim 2\%$.

### F. COMPARABLE METHODS TO THE LOGIT BALANCING

We argue that the logit balancing alleviates the effect of overfitting to the new classes, leading to better accuracy and less forgetting. To support the argument, we compare the accuracy of the models with various balancing methods; one with no balancing (denoted as ‘Baseline’), one with the proposed logit balancing (denoted as ‘Logit Balanced’), and one with the Balanced fine-tuning (denoted as ‘Balanced Fine-tuning’ or BF) proposed in [9]. To isolate the benefit of the balancing algorithms, we use the simple fine-tuning for $L_p(\cdot)$ and a simple uniform random sampling in the Figure 1-(c), (d).

As shown in the Figure 1-(b), the BF alleviates the class imbalance problem, so does confusion from the classes in the old task to the classes in the new task, for which the logit balancing aims. We hypothesize that the logit balancing also implicitly alleviates the class imbalance problem as well. Since the bias correcting layer [45] requires a held-out set for optimization after training, it is not comparable to our logit balancing.

Figure 7-(b),(e) show that our logit balancing outperforms the BF in accuracy and forgetting in all task transitions and the margin increases as the model learns more classes. At the final task, the logit balancing outperforms the BF by a large margin (+4.41% in accuracy and −19.61% in $F_{10}$). Other than the baseline and the BF, we also compare our method with post-processing method that reduces the bias from data imbalance and of a teacher network during distillation [55] by applying it to the E2E method. Our method ($A_{avg}: 59.12\% \pm 2.98\%$ and $A_{10}: 48.87\% \pm 2.14\%$) outperforms E2E with the post-processing ($A_{avg}: 57.08\% \pm 1.53\%$ and $A_{10}: 46.73\% \pm 0.13\%$) where we use same hyper-parameter used in other experiments, $T = 2$.

### G. COMPARISON WITH OTHER SAMPLE MEMORY MANAGEMENT POLICIES

We now compare our proposed sample memory management (‘Our SM’) with Random sampling (‘Random’) and Herding selection (‘Herding’). To isolate the effect of the sample memory management policy alone, we use the simple
fine-tuning for $L_p(\cdot)$ and do not use the proposed logit balancing or Balanced fine-tuning [9] and summarize the results in Figure 7-(c) and (f).

We observe that our sample memory managing method outperforms the state of the arts by a large margin (+2.44% in accuracy and −2.98% in $F_{10}$ at the last task). More
interestingly, the margin increases as the model learns more classes, i.e., the incremental steps progress. Figure 3-(b) shows that the choice of selection method results in a big difference in performance, especially when there are not many examples left (e.g., 10, 20, 50, 100 examples left per class) in the sample memory. Note that though the improvement in accuracy by our selection policy to the other methods is marginal at the early tasks but increases later because the number of samples per class decreases as the incremental steps progress.

**H. RESULTS ON TinyImageNet AND ImageNet-100**

We further conduct experiments on two other visual recognition datasets: the TinyImageNet and the ImageNet-100 and summarize the results in Table 2. Our proposed method outperforms the other state of the art with a large margin in all metrics except the $I_{10}$. It implies that our method improves the forgetting at the expense of intransigence. We argue that our method finds a better trade-off between the forgetting and the intransigence and results in better accuracy.

1) TinyImageNet

Following the experiments with CIFAR-100, we evaluate the methods with the same metrics: average accuracy ($A_{avg}$), accuracy at the final (10-th) task ($A_{10}$), Forgetting ($F_{10}$), Intransigence ($I_{10}$), and multi-head intransigence ($I_{k}^{mh}$) (Refer to Table 2). At the final task, our proposed method outperforms the end-to-end method by a large margin; +5.60% in accuracy $A_{10}$ and −11.92% in $F_{10}$ as shown in Figure 8-(a) and (d).

To isolate the benefit of the logit balancing, we use the simple fine-tuning for $L_{p}(-)$ and a simple uniform random sampling. Figure 8-(b), (e) show that our logit balancing improves accuracy marginally over the BF while it outperforms the BF in forgetting at all task transitions. As a trade-off between the forgetting and the intransigence due to the stability-plasticity dilemma, the intransigence by our method is higher than the other methods (Refer to Table 2). We hypothesize that our method gets the accuracy gain from the better trade-off followed by less forgetting ($F_{10}$). In addition, the margin between our logit balancing and the BF (and the baseline) increases as the model learns more classes. At the final task, the logit balancing outperforms the baseline by a huge margin (−18.85% in forgetting ($F_{10}$) and +4.7% in accuracy ($A_{10}$)) and the BF by a large margin (−7.76% in forgetting ($F_{10}$) and +0.09% in accuracy ($A_{10}$)).

Figure 8-(c) and (f) show the effect of the sample memory policy using the same setting as Sec. III-G. We observe that our sample memory managing method outperforms the state of the art by a large margin (+2.15% in accuracy $A_{10}$ and −1.95% in $F_{10}$ at the last task). Even though Figure 8-(c) shows that improvement in accuracy by our sampling policy to other methods is marginal at the early tasks but later increases because the sampling strategy becomes more important as the number of samples per class decreases (see related discussion in Sec. III-G).
2) ImageNet-100

For the experiments on ImageNet-100 dataset, we use the same experimental comparisons with the same set of evaluation metrics, following the experiments on CIFAR-100 and TinyImageNet datasets. The only difference of the experiments of ImageNet-100 to the others is that we report our results both in Top-1 and Top-5 accuracy. In Figure 9-(a),(d), our proposed method outperforms the End-to-End method by a large margin (+2.55 % in accuracy $A_{10}$ and $-4.51 \%$ in $F_{10}$) based on Top-5. Also, in Figure 10-(a),(d), our proposed method outperforms the End-to-End method by a large margin on Top-1 by a large margin (+3.20 % in accuracy $A_{10}$ and $-9.92 \%$ in $F_{10}$).

Figure 9-(b),(e) show that our logit balancing outperforms or performs marginally worse than the BF. Figure 10-(b),(e) show that our logit balancing outperforms the BF in accuracy and forgetting in all task transitions and the margin increases as the model learns more classes. At the final task, the logit balancing clearly outperforms the BF by a large margin (+2.44% in accuracy $A_{10}$ and $-7.71 \%$ in $F_{10}$ on Top-1). Despite the label noise of the ImageNet dataset, we argue that the Top-1 based $F$ is a more precise measure as Top-5 averages out the harm of some true-negatives.

Figure 9-(c), (f) and Figure 10-(c), (f) show the effect of the sample memory policy using the same setting described in Sec. III-G. In Figure 9-(c), (f), we observe that the random selection performs better than our sample memory managing method in Top-5 accuracy. Figure 10-(c), (f) shows that our sample memory management policy outperforms the state of the arts by a large margin (+3.77% in accuracy at the last task). At the same time, our sample memory management policy performs on par with the random selection in the forgetting (especially with Top-1 measure) despite the better accuracy. It is because forgetting is defined by the difference of accuracy between the best among previous models on the task and the current model. It implies that our sample memory managing method improves the accuracy but not the forgetting significantly. However, our sample memory management method shows the lowest standard deviation in forgetting compared to the random and Herding selections. It implies that our method samples meaningful examples, even in the various task configurations. In addition, we observe that our sample memory managing method performs marginally worse than the random selection baseline in Fig 9 (c), (f) in Top-5 accuracy. We conjecture the reason being that our sample memory managing method is based on the cumulative likelihood that relies on the Top-1 accuracy. Together with the label noise, our proposed sampling score may not show the best performance due to the discrepancy between the Top-1 and Top-5 evaluation strategies.

IV. CONCLUSION

We present a novel class-incremental learning method for the set-up with a sample memory. The proposal consists of two parts: logit balancing and sample memory managing policy. We observed the unintentional imbalance of confidence between the classes of the past and the new task. Logit balancing is to alleviate this problem in the class incremental learning by adding learning objective for calibration. The most of sample memory managing policy is marginally better than the uniform random sampling. For better sample memory managing policy, we used a cumulative likelihood score of samples as a criterion because it represents the knowledge of the class distribution. In the experimental validations on multiple datasets including CIFAR100, Tiny ImageNet, and ImageNet-100, our approach outperforms the state of the arts such as End-to-End [9] and our extension of EWC [8], denoted as EWC+SM, by large margins in forgetting.

REFERENCES

[1] G. M. van de Ven and A. S. Tolias, “Three continual learning scenarios and a case for generative replay,” in Proc. NIPS Continual Learn. Workshop, 2018, pp. 1–16.
[2] M. Mermillod, A. Bugaiska, and P. Bonin, “The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects,” Frontiers Psychol., vol. 4, pp. 2261–2269, Aug. 2013.
[3] A. Chaudhry, P. K. Dokania, T. Ajjanth, and P. H. S. Torr, “Riemannian walk for incremental learning: Understanding forgetting and intransigence,” in Proc. ECCV, 2018, pp. 532–547.
[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proc. NIPS, 2012, pp. 1097–1105.
[5] M. Norouzi, S. Bengio, Z. Chen, N. Jaitly, M. Schuster, Y. Wu, and D. Schuurmans, “Reward augmented maximum likelihood for neural structured prediction,” in Proc. NIPS, 2016, pp. 1723–1731.
[6] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Jham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, “Mastering the game of go with deep neural networks and tree search,” Nature, vol. 529, no. 7587, pp. 484–489, Jan. 2016.
[7] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, “ICaRL: Incremental classifier and representation learning,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 2001–2010.
[8] J. Kirkpatrick, R. Pascanu, N. C. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell, “Overcoming catastrophic forgetting in neural networks,” Proc. Nat. Acad. Sci. USA, vol. 114, no. 13, pp. 3521–3526, 2017.
[9] F. M. Castro, M. J. Marin-Jimenez, N. Guil, C. Schmid, and K. Alahari, “End-to-end incremental learning,” in Proc. ECCV, 2018, pp. 233–248.
[10] M. McCloskey and N. J. Cohen, “Catastrophic interference in connectionist networks: The sequential learning problem,” Psychol. Learn. Motivat., vol. 24, pp. 109–165, Dec. 1989.
[11] R. Ratcliff, “Connectionist models of recognition memory: Constraints imposed by learning and forgetting functions,” Psychol. Rev., vol. 97, no. 2, pp. 285–308, 1990.
[12] A. Gepperth and B. Hammer, “Incremental learning algorithms and applications,” in Proc. Eur. Symp. Artif. Neural Netw. (ESANN), Bruges, Belgium, 2016, pp. 1–13.
[13] M. Mermillod, A. Bugaiska, and P. Bonin, “The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects,” Frontiers Psychol., vol. 4, p. 504, Aug. 2013.
[14] G. I. Patrini, R. Kenkeri, J. L. Part, C. Kanen, and S. Wermter, “Continual lifelong learning with neural networks: A review,” Neural Netw., vol. 113, pp. 54–71, May 2019.
[15] F. Zenke, B. Poole, and S. Ganguli, “Continual learning through synaptic intelligence,” in Proc. ICML, Aug. 2017, pp. 3987–3995.
[16] 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 Proc. 24th Int. Conf. Pattern Recognit. (ICPR), Aug. 2018, pp. 2262–2268.
[17] Z. Li and D. Hoiem, “Learning without forgetting,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 40, no. 12, pp. 2935–2947, Dec. 2018.

[18] S.-W. Lee, J.-H. Kim, J. Ha, and B.-T. Zhang, “Overcoming catastrophic forgetting by incremental moment matching,” in Proc. NIPS, 2017, pp. 4652–4662.

[19] A. Mallya, D. Davis, and S. Lazebnik, “Piggyback: Adapting a single network to multiple tasks by learning to mask weights,” in Proc. ECCV, 2018, pp. 67–82.

[20] K. Lee, K. Lee, J. Shin, and H. Lee, “Overcoming catastrophic forgetting with unlabeled data in the wild,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 312–321.

[21] J. Yoon, S. Kim, E. Yang, and S. J. Hwang, “Scalable and order-robust continual learning with additive parameter decomposition,” in Proc. ICLR, 2020. [Online]. Available: https://openreview.net/forum?id=r1gdj2EKPB

[22] M. Toneva, A. Ronti, A. Trischler, Y. Bengio, and G. J. Gordon, “An empirical study of example forgetting during deep neural network learning,” in Proc. ICLR, 2019, pp. 1–19.

[23] I. J. Goodfellow, M. Mirza, D. Xiao, A. Courville, and Y. Bengio, “An empirical investigation of catastrophic forgetting in gradient-based neural networks,” 2013, arXiv:1312.6211. [Online]. Available: https://arxiv.org/abs/1312.6211

[24] R. K. Srivastava, J. Masci, S. Kazerounian, F. Gomez, and J. Schmidhuber, “Compete to compute,” in Proc. NIPS, 2013, pp. 2310–2318.

[25] A. Mallya and S. Lazebnik, “PackNet: Adding multiple tasks to a single network to multiple tasks by learning to mask weights,” in Proc. ICLR, 2019, pp. 1–20.

[26] A. Mallya, D. Davis, and S. Lazebnik, “Piggyback: Adapting a single network to multiple tasks by learning to mask weights,” in Proc. ECCV, 2018, pp. 67–82.

[27] M. Toneva, A. Sordoni, R. T. des Combes, A. Trischler, Y. Bengio, and G. J. Gordon, “An empirical study of example forgetting during deep neural network learning,” in Proc. ICLR, 2019, pp. 1–19.

[28] I. J. Goodfellow, M. Mirza, D. Xiao, A. Courville, and Y. Bengio, “An empirical investigation of catastrophic forgetting in gradient-based neural networks,” 2013, arXiv:1312.6211. [Online]. Available: https://arxiv.org/abs/1312.6211

[29] R. K. Srivastava, J. Masci, S. Kazerounian, F. Gomez, and J. Schmidhuber, “Compete to compute,” in Proc. NIPS, 2013, pp. 2310–2318.

[30] A. Mallya and S. Lazebnik, “PackNet: Adding multiple tasks to a single network by iterative pruning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 7765–7773.

[31] K. Shmelkov, C. Schmid, and K. Alahari, “Incremental learning of object detectors without catastrophic forgetting,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 3400–3409.

[32] R. Aljundi, M. Rohrbach, and T. Tuyltaelaars, “Selfless sequential learning,” in Proc. ICLR, 2019, pp. 1–17.

[33] J. Serra, D. Suris, M. Miron, and A. Karatzoglou, “Overcoming catastrophic forgetting with hard attention to the task,” in Proc. ICML, 2018, pp. 1–17.

[34] M. Welling, “Herding dynamical weights to learn,” in Proc. Multiple Classifier Syst., 2009, pp. 1121–1128.

[35] H. Shin, J. K. Lee, J. Kim, and J. Kim, “Continual learning with deep generative replay,” in Proc. NIPS, 2017, pp. 2990–2999.

[36] A. Seff, A. Beaton, D. Suo, and H. Liu, “Continual learning in generative adversarial nets,” 2017, arXiv:1705.08395. [Online]. Available: https://arxiv.org/abs/1705.08395

[37] C. Wu, L. Herranz, X. Liu, Y. Wang, J. van de Weijer, and B. Raducanu, “Memory replay GANs: Learning to generate new categories without forgetting,” in Proc. NIPS, 2018, pp. 5962–5972.

[38] W. Hu, Z. Lin, B. Liu, C. Tao, Z. Tao, J. Ma, and R. Yan, “Overcoming catastrophic forgetting via model adaptation,” in Proc. ICLR, 2019. [Online]. Available: https://openreview.net/forum?id=rYgOcvA5XY

[39] S. Pang, L. Zhu, G. Chen, A. Sarrafzadeh, T. Ban, and D. Inoue, “Dynamic class imbalance learning for incremental LPSVM,” Neural Netw., vol. 44, pp. 87–100, Aug. 2013.

[40] G. Ditlzer, M. D. Muhlbaier, and R. Polikar, “Incremental learning of new classes in unbalanced datasets: Learn+ “ UDNCL,” in Proc. Int. Workshop Multiple Classifier Syst. Berlin, Germany: Springer, 2010, pp. 33–42.

[41] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, “On calibration of modern neural networks,” in Proc. ICLR, 2017, pp. 1–14.

[42] G. Pereyra, G. Tucker, J. Chorowski, L. Kaiser, and G. E. Hinton, “Regularizing neural networks by penalizing confident output distributions,” 2017, arXiv:1701.06548. [Online]. Available: https://arxiv.org/abs/1701.06548

[43] V. Kuleshov and S. Ermon, “Reliable confidence estimation via online learning,” 2016, arXiv:1607.03594. [Online]. Available: https://arxiv.org/abs/1607.03594

[44] A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny, “Efficient lifelong learning with A-GEM,” in Proc. ICLR, 2019, pp. 1–20.
YEONGWOO NAM was born in Bucheon, South Korea, in 1994. He received the B.S. degree in computer science from Soongsil University, Seoul, in 2019. He is currently pursuing the master’s degree with the School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology (GIST). His research interests include continual learning and domain adaptation in machine learning.

JONGHYUN CHOI (Member, IEEE) received the B.S. and M.S. degrees in electrical engineering and computer science from Seoul National University, Seoul, South Korea, in 2003 and 2008, respectively, and the Ph.D. degree from the University of Maryland, College Park, in 2015, under the supervision of Prof. L. S. Davis. He was a Research Intern in a number of research labs, including the U.S. Army Research Laboratory, in 2012, Adobe Research, in 2013, Disney Research, Pittsburgh, in 2014, and Microsoft Research, Redmond, in 2014. He was a Senior Researcher with the Comcast Applied AI Research, Washington, DC, USA, from 2015 to 2016. He was a Research Scientist with the Allen Institute for Artificial Intelligence (AI2), Seattle, WA, USA, from 2016 to 2018. He is currently an Assistant Professor with the Gwangju Institute of Science and Technology, Gwangju, South Korea. He is also an affiliated Research Scientist. His research interests include visual recognition using weakly supervised data for semantic understanding of images and videos and visual understanding for edge devices.