Overcoming Catastrophic Forgetting with Unlabeled Data in the Wild

Kibok Lee∗ Kimin Lee† Jinwoo Shin† Honglak Lee∗

Presenters: Nikhil Kannan, Ying Fan

Catastrophic Forgetting

- Goal of class-incremental learning is to learn a model that performs well on previous and new tasks without task boundaries. But it suffers from catastrophic forgetting.

- Training Neural Networks on new tasks causes it to forget information learned from previously trained tasks, degrading model performance on earlier tasks.

- Primary reason for catastrophic forgetting is limited resources for scalability.

Contributions

- Novel class-incremental learning scheme that uses large stream of unlabeled data.
- Global knowledge distillation
- Learning strategy to avoid overfitting to most recent task
- Confidence based sampling method to effectively leverage unlabeled dataset

Class Incremental Learning Setting

- $(x, y) \in \mathbb{D}$

- T is a supervised task mapping $x \rightarrow y$

- For task T_t , corresponding dataset is $\mathbb{D}_t$ and coreset is $\mathbb{D}^{cor}{}_{t-1} \subseteq \mathbb{D}_{t-1} \cup \mathbb{D}^{cor}{}_{t-2}$ contains representative data of previous tasks $\mathrm{T}_{1:(t-1)} = \{\mathrm{T}_1 , \ldots , \mathrm{T}_t\}$. For task T_t corresponding labeled training data used is represented as $\mathbb{D}_t^{trn} = \mathbb{D}_t \cup \mathbb{D}^{cor}{}_{t-1}$.

- $\mathrm{M}_t = \{\theta, \emptyset_{1:t}\}$ is a set of learnable parameters of a model where θ indicates shared task parameters and $\emptyset_{1:t} = \{\emptyset_1 , \ldots, \emptyset_t\}$ are task specific parameters.

Class Incremental Learning: Approach

- The goal at each task t is to train a model M_t to perform the current task T_t as well as previous tasks $\mathrm{T}_{1:(t-1)}$ without any task boundaries.

- The input at each task stage is the previous model M_{t-1} , coreset $\mathbb{D}^{cor}{}_{t-1}$, training dataset $\mathbb{D}_t$ and large stream of unlabeled data $\mathbb{D}^{Wild}$.

- Output at each task stage is a new coreset $\mathbb{D}^{cor}{}_t$ and model $\mathrm{M}_t = \{\theta, \emptyset_{1:t}\}$

Local Knowledge Distillation

- Train the model M_t by minimizing the classification loss: $L_{cls}(\theta, \emptyset_{1:t} ; \mathbb{D}_t^{trn})$.

- In the class incremental learning setting, the limited capacity of coreset causes the model to suffer from catastrophic forgetting. To overcome this issue, utilize previously trained model M_{t-1} , that contains knowledge of previous tasks to generate soft labels.

- Optimize $\sum_{s=1}^{t-1} L_{dst} (\theta, \emptyset_s ; \mathrm{P}_t , \mathbb{D}_t)$, where $\mathrm{P}_t = \{\theta^P , \emptyset^P_{1:(t-1)}\} = \mathrm{M}_{t-1}$ is a previous trained model

Local Knowledge Distillation (Cont.)

- Minimize the joint objective: $L_{\text{cls}}(\theta, \emptyset_{1:t} ; \mathbb{D}_t^{\text{trn}}) + \sum_{s=1}^{t-1} L_{\text{dst}}(\theta, \emptyset_s ; \mathrm{P}_t , \mathbb{D}_t)$

- Solving the above optimization problem is called local knowledge distillation. Transfers the knowledge within each of the tasks.

- The issue with local knowledge distillation is that is defined in a task-wise manner and misses the knowledge about discriminating between classes in different tasks.

Global Knowledge Distillation

- Distill the knowledge of reference models globally by minimizing the following loss: $L_{dst}(\theta, \emptyset_{1:(t-1)}\ ; \mathrm{P}_t, \mathbb{D}_t^{trn} \cup \mathbb{D}^{ext}_t)$

- Learning using the above function causes bias, since P_t does not have knowledge regarding the current task, hence performance on the current task is degraded.

- Introduce teacher model $\mathrm{C}_t = \{\theta^C, \emptyset^C_t\}$ specialized to learn the current task T_t
$L_{dst}(\theta, \emptyset_t\ ; \mathrm{C}_t, \mathbb{D}_t^{trn} \cup \mathbb{D}^{ext}_t)$.

- Teacher model C_t is trained by minimizing $L_{cls}(\theta^C, \emptyset^C_t; \mathbb{D}_t)$

Global Knowledge Distillation (Cont.)

- P_t learns to perform tasks $\mathrm{T}_{1:(t-1)}$ and C_t learns to perform the current task T_t , but knowledge distillation between $\mathrm{T}_{1:(t-1)}$ and T_t is not captured by the either of the reference models.

- Define Q_t , an ensemble of reference models P_t and C_t

- Ensemble Q_t : $L_{dst}(\theta, \emptyset_{1:t}\ ;\ Q_t, \mathbb{D}^{ext}_t)$

- The global distillation model learns by optimizing the following loss:

$L_{cls}(\theta, \emptyset_{1:t}\ ;\ \mathbb{D}_t^{trn}) + L_{dst}(\theta, \emptyset_{1:(t-1)}\ ;\ \mathrm{P}_t, \mathbb{D}_t^{trn} \cup \mathbb{D}^{ext}_t) + L_{dst}(\theta, \emptyset_t\ ;\ \mathrm{C}_t, \mathbb{D}_t^{trn} \cup \mathbb{D}^{ext}_t) + L_{dst}(\theta, \emptyset_{1:t}\ ;\ Q_t, \mathbb{D}^{ext}_t)$

Fine-Tuning and Normalization

- Since the amount of data from the previous tasks is smaller than the current task, model prediction is biased towards the current task.

- To remove the bias, fine tune the model after the training phase by scaling the computed gradient from the data with label *k*.

- $w^{(k)}{}_D = \frac{1}{|\{(x,y) \in \mathbb{D} \mid y=k\}|}$, scaling the gradient is similar to feeding data multiple times (data weighting).

- Normalizing weights by multiplying them with $^{|\mathbb{D}|}/_{|T|}$ to balance the dataset $\mathbb{D}$

Fine-Tuning and Normalization (Cont.)

- Fine- tuning task-specific ($\emptyset_{1:t}$) using data weighting to remove any bias from training data and to equally weigh training data for all tasks.

- Fine-tuning shared parameters (θ) is not required since it already contains relevant information from all training data.

- Loss Weight: balance the contribution of each loss by the relative size of each task learned in the loss; $w^L = \frac{|T|}{|T_{1:t}|}$

3-step Learning Algorithm

- Learning strategy has three steps
  - Training C_t specialized for learning the current task T_t
  - Training M_t through global knowledge distillation of reference models P_t , Q_t , C_t
  - Fine-tuning model parameters using data weighting.

3-step Learning Algorithm

Algorithm 1 3-step learning with GD.

$t = 1$
while true do
  Input: previous model $\mathcal{P}_t = \mathcal{M}_{t-1}$, coreset $\mathcal{D}_{t-1}^{\texttt{cor}}$, training dataset $\mathcal{D}_t$, unlabeled data stream $\mathcal{D}_t^{\texttt{wild}}$
  Output: new coreset $\mathcal{D}_t^{\texttt{cor}}$, model $\mathcal{M}_t = \{\theta, \phi_{1:t}\}$
  $\mathcal{D}_t^{\texttt{trn}} = \mathcal{D}_t \cup \mathcal{D}_{t-1}^{\texttt{cor}}$
  $N_C = |\mathcal{D}_{t-1}^{\texttt{cor}}|$, $N_D = |\mathcal{D}_t^{\texttt{trn}}|$
  Sample $\mathcal{D}_t^{\texttt{ext}}$ from $\mathcal{D}_t^{\texttt{wild}}$ using Algorithm 2
  Train $\mathcal{C}_t$ by minimizing Eq. (12)
  if $t > 1$ **then**
    Train $\mathcal{M}_t$ by minimizing Eq. (9)
    Fine-tune $\phi_{1:t}$ by minimizing Eq. (9), with data weighting in Eq. (10)
  else
    $\mathcal{M}_t = \mathcal{C}_t$
  end if
  Randomly sample $\mathcal{D}_t^{\texttt{cor}} \subseteq \mathcal{D}_t^{\texttt{trn}}$ such that $|\{(x, y) \in \mathcal{D}_t^{\texttt{cor}} | y = k\}| = N_C / |\mathcal{T}_{1:t}|$ for $k \in \mathcal{T}_{1:t}$
  $t = t + 1$
end while

Sampling External Dataset

- The main issues with using unlabeled data in knowledge distillation.
  - Training is computationally expensive
  - Most of the unlabeled data might be irrelevant to the tasks the model learns

- The paper proposes a sampling method to sample an external dataset from large stream of unlabeled data that benefits knowledge distillation.

Sampling for Confidence Calibration

- Sampling external data that is expected to be in previous tasks is desirable, since it alleviates catastrophic forgetting.

- Neural Nets tend to be highly overconfident as they produce prediction with high confidence for OOD data.

- To achieve confidence calibrated outputs, model learns from certain amount of OOD data and data from previous tasks.

Confidence Calibration

- For the model to produce confidence calibrated outputs, following confidence loss function is considered: $L_{\text{cnf}}(\theta, \emptyset\ ; \mathbb{D}) = \frac{1}{|\mathbb{D}||T|}\sum_{x\in\mathbb{D}}\sum_{y\in T}[-\log p(y|x;\theta,\emptyset)]$

- During 3-step learning, training C_t has no reference model hence it learns from confidence loss. By optimizing on confidence loss, model learns to produce predictions with low confidence for OOD data.

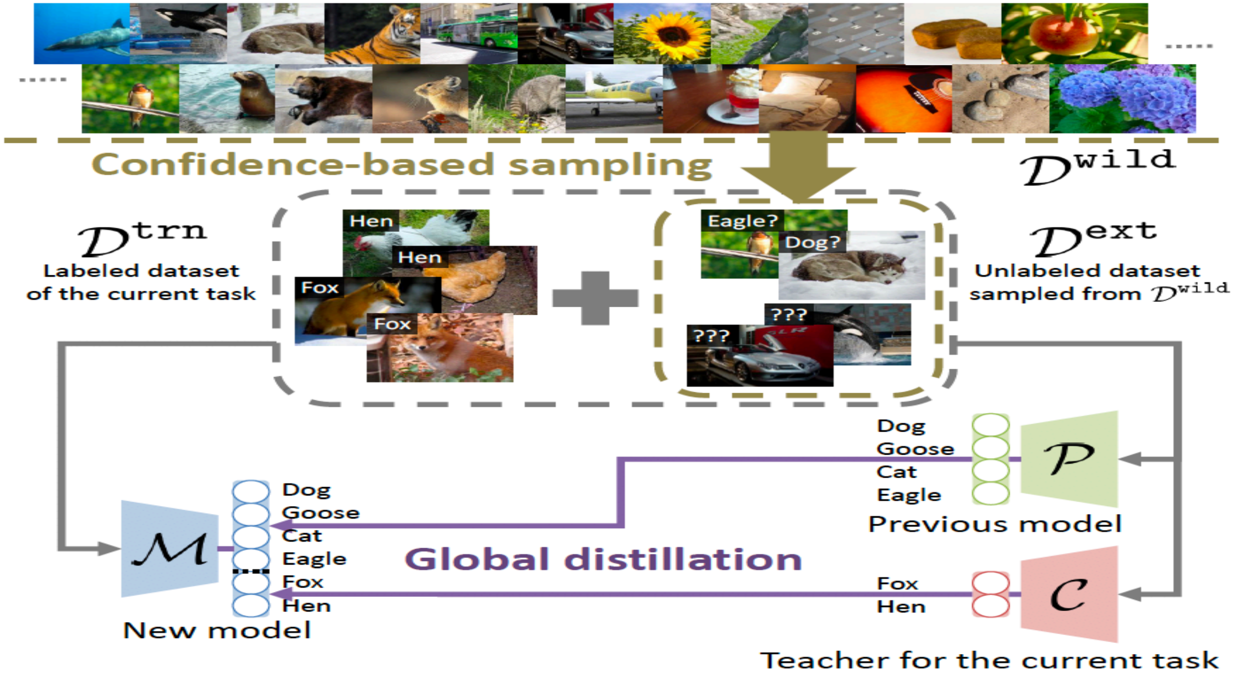
- C_t learns by optimizing $L_{\text{cls}}(\theta^C, \emptyset^C{}_t; \mathbb{D}_t) + L_{\text{cnf}}(\theta^C, \emptyset^C{}_t; \mathbb{D}_{t-1}{}^{\text{cor}} \cup \mathbb{D}^{\text{ext}}{}_t)$

Sampling External Dataset: Algorithm

Algorithm 2 Sampling external dataset.

```
Input: previous model $\mathcal{P}_t = \{\theta^{\mathcal{P}}, \phi^{\mathcal{P}}_{1:(t-1)}\}$,
       unlabeled data stream $\mathcal{D}^{\texttt{wild}}_t$, sample size $N_D$,
       number of unlabeled data to be retrieved $N_{\max}$
Output: sampled external dataset $\mathcal{D}^{\texttt{ext}}_t$
$\mathcal{D}^{\text{prev}} = \emptyset$, $\mathcal{D}^{\text{OOD}} = \emptyset$
$N_{\text{prev}} = 0.3N_D$, $N_{\text{OOD}} = 0.7N_D$
$N(k) \triangleq |\{(x, y, p) \in \mathcal{D}^{\text{prev}} | y = k\}|$
while $|\mathcal{D}^{\text{OOD}}| < N_{\text{OOD}}$ do
    Get $x \in \mathcal{D}^{\texttt{wild}}_t$ and update $\mathcal{D}^{\text{OOD}} = \mathcal{D}^{\text{OOD}} \cup \{x\}$
end while
$N_{\text{ret}} = N_{\text{OOD}}$
while $N_{\text{ret}} < N_{\max}$ do
    Get $x \in \mathcal{D}^{\texttt{wild}}_t$ and compute the prediction of $\mathcal{P}$:
     $\hat{p} = \max_y p\left(y|x; \theta^{\mathcal{P}}, \phi^{\mathcal{P}}_{1:(t-1)}\right)$,
     $\hat{y} = \arg\max_y p\left(y|x; \theta^{\mathcal{P}}, \phi^{\mathcal{P}}_{1:(t-1)}\right)$
    if $N(\hat{y}) < N_{\text{prev}}/|\mathcal{T}_{1:(t-1)}|$ then
        $\mathcal{D}^{\text{prev}} = \mathcal{D}^{\text{prev}} \cup \{(x, \hat{y}, \hat{p})\}$
    else
        Replace the least probable data in class $\hat{y}$:
         $(x', \hat{y}, p') = \arg\min_{\{(x,y,p) \in \mathcal{D}^{\text{prev}} | y = \hat{y}\}} p$
        if $p' < \hat{p}$ then
            $\mathcal{D}^{\text{prev}} = (\mathcal{D}^{\text{prev}} \setminus \{(x', \hat{y}, p')\}) \cup \{(x, \hat{y}, \hat{p})\}$
        end if
    end if
    $N_{\text{ret}} = N_{\text{ret}} + 1$
end while
Return $\mathcal{D}^{\texttt{ext}}_t = \mathcal{D}^{\text{OOD}} \cup \{x | (x, y, p) \in \mathcal{D}^{\text{prev}}\}$
```

Global Distillation Model



Related Work

Continual lifelong learning: **class**/task/data incremental learning

Methods: model-based and data-based

·Model based: parameters for new tasks are directly constrained to be around that for previous tasks

·Data based: data distribution from previous tasks are used to distill knowledge for later tasks; previous works focus on task-wise local distillation (eq.2), previous state-of-art: LwF, DR, E2E.

Scalability: sampling from an external dataset; discarded after learning

Related Work for Comparison

Previous knowledge distillation in data based methods: only distill the task-wise knowledge

·Lw, task incremental--class incremental

·Dr, task incremental, learning with two teachers--class incremental

·E2E, class incremental with fine tuning, but sacrifices the diversity of frequent classes--data weighting

·Orthogonal to model based; can be combined

Proposed methods: GD

Experiments

Datasets:

Labeled:CIFAR_100,ImageNet ILSVRC 2012

Unlabeled: TinyImages, ImageNet2011

Design tasks: total 100 classes, divide into splits of 5,10,20--task size: 20,10,5

Hyper parameters: WRN-16-2, coreset size=2000, temperature for smoothing softmax probabilities: 2 for P,C, 1 for Q

Experiments

Evaluation metric

The accuracy of the s-th model at r-th task, s>=r

$$A_{r,s} = \frac{1}{|\mathcal{D}_r^{\mathtt{test}}|} \sum_{(x,y)\in\mathcal{D}_r^{\mathtt{test}}} \mathbb{I}(\hat{y}(x;\mathcal{M}_s) = y)$$

ACC: weighted combination of all tasks and all models:

$$\mathrm{ACC} = \frac{1}{t-1} \sum_{s=2}^{t} \sum_{r=1}^{s} \frac{|\mathcal{T}_r|}{|\mathcal{T}_{1:s}|} A_{r,s}.$$

FGT: performance decay:

$$\mathrm{FGT} = \frac{1}{t-1} \sum_{s=2}^{t} \sum_{r=1}^{s-1} \frac{|\mathcal{T}_r|}{|\mathcal{T}_{1:s}|} (A_{r,r} - A_{r,s}),$$

Experiments

Evaluation:

Overall performance:

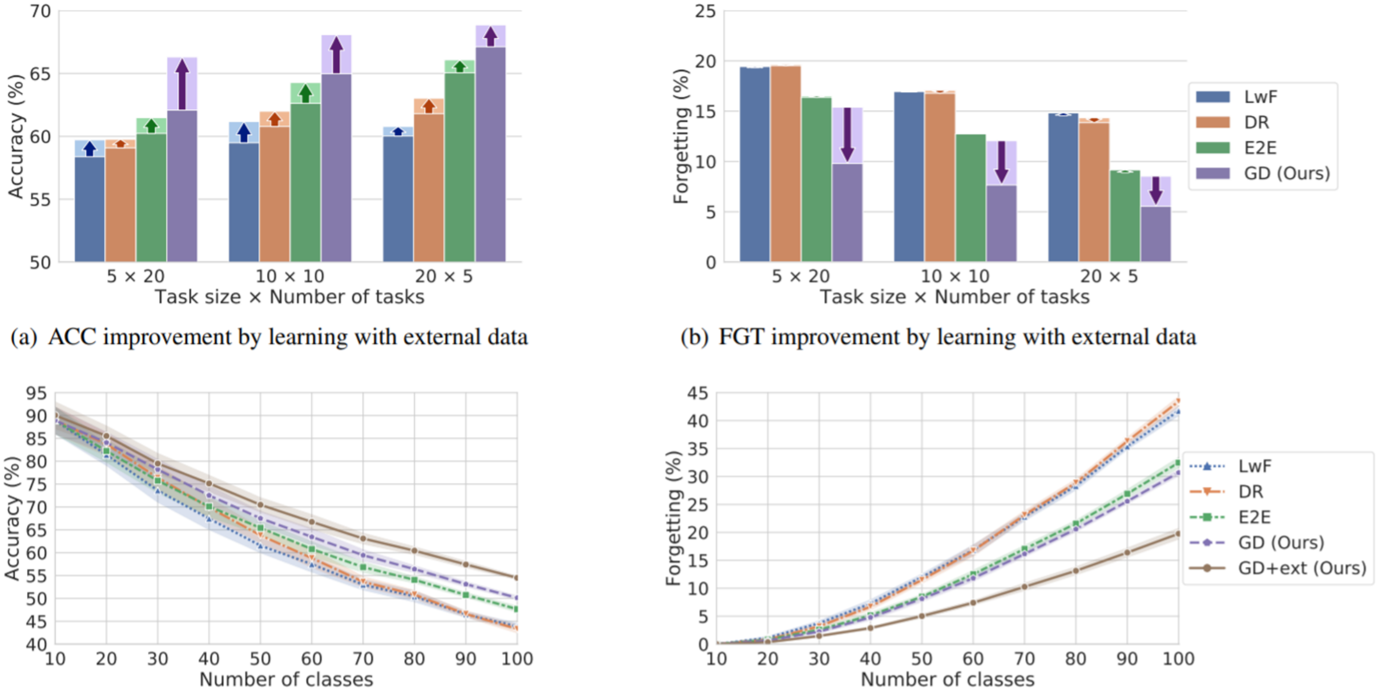
Table 1. Comparison of methods on CIFAR-100 and ImageNet. We report the mean and standard deviation of ten trials for CIFAR-100 and nine trials for ImageNet with different random seeds in %. ↑ (↓) indicates that the higher (lower) number is the better.

Dataset	CIFAR-100						ImageNet					
Task size	5		10		20		5		10		20	
Metric	ACC (↑)	FGT (↓)	ACC (↑)	FGT (↓)	ACC (↑)	FGT (↓)	ACC (↑)	FGT (↓)	ACC (↑)	FGT (↓)	ACC (↑)	FGT (↓)
Oracle	78.6 ± 0.9	3.3 ± 0.2	77.6 ± 0.8	3.1 ± 0.2	75.7 ± 0.7	2.8 ± 0.2	68.0 ± 1.7	3.3 ± 0.2	66.9 ± 1.6	3.1 ± 0.3	65.1 ± 1.2	2.7 ± 0.2
Baseline	57.4 ± 1.2	21.0 ± 0.5	56.8 ± 1.1	19.7 ± 0.4	56.0 ± 1.0	18.0 ± 0.3	44.2 ± 1.7	23.6 ± 0.4	44.1 ± 1.6	21.5 ± 0.5	44.7 ± 1.2	18.4 ± 0.5
LwF [24]	58.4 ± 1.3	19.3 ± 0.5	59.5 ± 1.2	16.9 ± 0.4	60.0 ± 1.0	14.5 ± 0.4	45.6 ± 1.9	21.5 ± 0.4	47.3 ± 1.5	18.5 ± 0.5	48.6 ± 1.2	15.3 ± 0.6
DR [12]	59.1 ± 1.4	19.6 ± 0.5	60.8 ± 1.2	17.1 ± 0.4	61.8 ± 0.9	14.3 ± 0.4	46.5 ± 1.6	22.0 ± 0.5	48.7 ± 1.6	18.8 ± 0.5	50.7 ± 1.2	15.1 ± 0.5
E2E [3]	60.2 ± 1.3	16.5 ± 0.5	62.6 ± 1.1	12.8 ± 0.4	65.1 ± 0.8	8.9 ± 0.2	47.7 ± 1.9	17.9 ± 0.4	50.8 ± 1.5	13.4 ± 0.4	53.9 ± 1.2	8.8 ± 0.3
GD (Ours)	62.1 ± 1.2	15.4 ± 0.4	65.0 ± 1.1	12.1 ± 0.3	67.1 ± 0.9	8.5 ± 0.3	50.0 ± 1.7	16.8 ± 0.4	53.7 ± 1.5	12.8 ± 0.5	56.5 ± 1.2	8.4 ± 0.4
+ ext	**66.3 ± 1.2**	**9.8 ± 0.3**	**68.1 ± 1.1**	**7.7 ± 0.3**	**68.9 ± 1.0**	**5.5 ± 0.4**	**55.2 ± 1.8**	**9.6 ± 0.4**	**57.7 ± 1.6**	**7.4 ± 0.3**	**58.7 ± 1.2**	**5.4 ± 0.3**

Experiments

Evaluation:

Overall performance:



(a) ACC improvement by learning with external data

(b) FGT improvement by learning with external data

(c) ACC with respect to the number of trained classes

(d) FGT with respect to the number of trained classes

Figure 2. Experimental results on CIFAR-100. (a,b) Arrows show the performance gain in the average incremental accuracy (ACC) and average forgetting (FGT) by learning with unlabeled data, respectively. (c,d) Curves show ACC and FGT with respect to the number of trained classes when the task size is 10. We report the average performance of ten trials.

Experiments

Effect of the reference models

Table 2. Comparison of models learned with different reference models on CIFAR-100 when the task size is 10. "$\mathcal{P}$," "$\mathcal{C}$," and "$\mathcal{Q}$" stand for the previous model, the teacher for the current task, and their ensemble model, respectively.

$\mathcal{P}$	$\mathcal{C}$	$\mathcal{Q}$	ACC (↑)	FGT (↓)
✓			62.9 ± 1.2	14.7 ± 0.4
✓	✓		67.0 ± 0.9	10.7 ± 0.3
		✓	65.7 ± 0.9	11.2 ± 0.2
✓	✓	✓	**68.1** ± **1.1**	**7.7** ± **0.3**

Experiments

Effect of the teacher for the current task

Table 3. Comparison of models learned with a different teacher for the current task $\mathcal{C}$ on CIFAR-100 when the task size is 10. For "`cls`," $\mathcal{C}$ is not trained but the model learns by optimizing the learning objective of $\mathcal{C}$ directly. The model learns with the proposed 3-step learning for "`dst`." The confidence loss is added to the learning objective for $\mathcal{C}$ for "`cnf`." We do not utilize $\mathcal{Q}$ for this experiment, because "`cls`" has no explicit $\mathcal{C}$.

$\mathcal{C}$	Confidence	ACC (↑)	FGT (↓)
✗		62.9 ± 1.2	14.7 ± 0.4
`cls`		62.9 ± 1.3	14.5 ± 0.5
`cls`	`cnf`	65.3 ± 1.0	11.7 ± 0.3
`dst`		66.2 ± 1.0	11.2 ± 0.3
`dst`	`cnf`	**67.0 ± 0.9**	**10.7 ± 0.3**

Experiments

Effect of balanced fine-tuning

Table 4. Comparison of different balanced learning strategies on CIFAR-100 when the task size is 10. “DW,” “FT-DSet,” and “FT-DW” stand for training with data weighting in Eq. (10) for the entire training, fine-tuning with a training dataset balanced by removing data of the current task, and fine-tuning with data weighting, respectively.

Balancing	ACC (↑)	FGT (↓)
✗	67.1 ± 0.9	11.5 ± 0.3
DW	67.9 ± 0.9	9.6 ± 0.2
FT-DSet	67.2 ± 1.1	8.4 ± 0.2
FT-DW	**68.1** ± **1.1**	**7.7** ± **0.3**

Experiments

Effect of external data sampling

Table 5. Comparison of different external data sampling strategies on CIFAR-100 when the task size is 10. "Prev" and "OOD" columns describe the sampling method for data of previous tasks and out-of-distribution data, where "Pred" and "Random" stand for sampling based on the prediction of the previous model $\mathcal{P}$ and random sampling, respectively. In particular, for when sampling OOD by "Pred," we sample data minimizing the confidence loss $\mathcal{L}_{\mathrm{cnf}}$. When only Prev or OOD is sampled, the number of sampled data is matched for fair comparison.

Prev	OOD	ACC (↑)	FGT (↓)
✗	✗	65.0 ± 1.1	12.1 ± 0.3
✗	Random	67.6 ± 0.9	9.0 ± 0.3
Pred	✗	66.0 ± 1.2	7.8 ± 0.3
Pred	Pred	65.7 ± 1.1	10.2 ± 0.2
Pred	Random	**68.1** ± **1.1**	**7.7** ± **0.3**

Conclusion

·Leverage a large stream of unlabeled data

·Global distillation aims to keep the knowledge of the reference models without task boundaries, leading better knowledge distillation

·3-step learning scheme effectively leverages the external dataset sampled by the confidence-based sampling strategy from the stream of unlabeled data

Quiz Questions

Which of the following statements are true about the global distillation model

A) Training a reference teacher’s model to specialize in learning only the current task
B) Knowledge distillation for the ensemble model is performed over both the training data and sampled external unlabeled data
C) Fine-tuning using data weighting is performed over all model parameters
D) Global distillation model is trained through knowledge distillation over 3 reference models.

Answer: A and D

Which of the following statements are true about confidence calibration for sampling:

A) Confidence calibration is performed on all reference models
B) It prevents the model from making overconfident predictions on OOD data by optimizing over the confidence loss
C) Confidence calibrated outputs are produced by optimizing the loss function over only the sampled external dataset.
D) Confidence calibrations increase the overall accuracy of the model by sampling better external data from a stream of unlabeled data

Answer: B and D

Which external data sampling strategy provides the highest model accuracy:

A) Random sampling of OOD data and sampling based on predictions of previous model
B) Only random sampling of OOD data
C) sampling based on predictions of previous model and sampling OOD data based on predictions of previous model
D) No external data sampling.

Answer: A