Memory-Efficient Semi-Supervised Continual Learning: The World is its Own Replay Buffer -Supplementary Materials-

James Smith
Georgia Institute of Technology
Atlanta, Georgia, USA
jamessealesmith@gatech.edu

Jonathan Balloch
Georgia Institute of Technology
Atlanta, Georgia, USA
balloch@gatech.edu

Yen-Chang Hsu
Samsung Research America
Mountain View, CA, USA
yenchang.hsu@samsung.com

Zsolt Kira
Georgia Institute of Technology
Atlanta, Georgia, USA
zkira@gatech.edu

Supplementary Metrics: In addition to the metrics used in the main paper, we also report backwards Backward Transfer (BWT) [1] and Forgetting (FTG) [2]. BWT is a measurement of increase in performance on task n after training across all tasks $1 \ldots N$. A higher value is better, indicating that the learner is better at performing task n after learning the subsequent tasks. A negative value indicates a drop in performance, which is typically expected in class incremental learning. A weakness of this metric is that it measures performance relative to *local* tasks and does not reflect performance on the *global* task of class incremental learning (i.e. the softmax outputs are across only the local per-task categories, not across all of the categories encountered throughout training). FGT is a measurement of decrease in performance on task n with respect to the *global* task; it is essentially negative backward transfer adopted for class incremental learning. A lower value is better, indicating that the learner has experienced less average performance decrease on task n throughout training. A weakness of this metric is that it does not account for natural decrease in performance due to the increasingly more difficult global task characteristic in class incremental learning. A key difference between BWT and FGT is that when evaluating task n performance for BWT, only task n classes can be returned during inference, whereas for FGT, all tasks classes $1 \ldots n$ can be returned. We include both of these metrics for experiment results during all subsequent sections because while neither is regularly used for class incremental learning, they may be useful to the reader.

$$BWT = \frac{1}{N-1} \sum_{n=1}^{N-1} (A_{N,n} - A_{n,n}) \tag{10}$$

$$FGT = \frac{1}{N-1} \sum_{i=2}^{N} \sum_{n=1}^{i-1} \frac{|\mathcal{T}_n|}{|\mathcal{T}_{1:i}|} (R_{n,n} - R_{i,n}) \tag{11}$$

where:

$$R_{i,n} = \frac{1}{|\mathcal{D}_n^{test}|} \sum_{(x,y) \in \mathcal{D}_n^{test}} \mathbb{1}(\hat{y}(x, \theta_{i,1:n}) = y) \tag{12}$$

A. *DistillMatch Ablation Study*

Here, we ablate our method in two experiment scenarios: RandomClass Tasks with Uniform Unlabeled Data Distribution (Table Ia) and ParentClass Tasks with PositiveSuperclass Unlabeled Data Distribution (Table Ib). Ω curves for both Tables are given in Figure 1. In the former case, we find that the hard distillation loss (eq. 7) is the most significant contribution, but the semi-supervised consistency loss (eq. 4), class balancing (eq. 3), and soft distillation loss (eq. 1) add significant performance gains as well. In the later case, we actually find the semi-supervised consistency loss (eq. 4) and distillation loss (eq. 1) to be the most important, while class balancing (eq. 3) and hard distillation loss (eq. 7) perform very similarly. This reflects the strength of our method: DM performs well in all of our experiments because it has components which vary in importance depending on the scenario (i.e. coreset size and object-object correlations).

B. *Additional Experiment Details*

We used used a batch size of 64 for labeled training data and 128 for unlabeled training data. As done in [2], we train over 200 epochs per task with a tuned learning rate decaying by 0.1 after 120, 160, and 180 epochs. When a coreset is present, we include finetuning of the final layer in our model using only the coreset and class balancing, as introduced in GD [2]. If finetuning, the model is trained over the first 180 epochs in the same manner, but after 180 epochs the learning rate is reset to 10% of the initial learning rate and is trained for 20 additional epochs with decays by 0.1 after 10, 15 epochs. We

TABLE I: Results (%) for Selected Ablation Studies on CIFAR-100 with 20% Labeled Data. Results are reported as an average of 3 runs with mean and standard deviation. Each row represents a part of our method which is removed as part of the study.

(a) RandomClass Tasks with Uniform Unlabeled Data Distribution, 10 Tasks, no Coreset

Ablation	A_N (↑)	Ω (↑)	BWT (↑)	FGT (↓)
ℓ_{pl} - eq. 7	7.7 ± 0.5	32.0 ± 0.2	-5.8 ± 1.9	56.6 ± 1.9
$w(k)$ - eq. 3	30.2 ± 1.9	69.6 ± 0.5	-4.8 ± 0.2	10.5 ± 0.5
ℓ_{ul} - eq. 4	33.3 ± 0.9	71.2 ± 2.3	-0.7 ± 0.3	7.7 ± 0.2
ℓ_{dst} - eq. 1	35.2 ± 1.1	74.1 ± 1.7	-4.8 ± 0.4	8.0 ± 0.9
Full Method	37.5 ± 0.7	76.9 ± 2.5	-1.0 ± 1.0	6.5 ± 0.5

(b) ParentClass Tasks with PositiveSuperclass Unlabeled Distribution, 20 Tasks, 400 image coreset

Ablation	A_N (↑)	Ω (↑)	BWT (↑)	FGT (↓)
ℓ_{pl} - eq. 7	19.3 ± 1.1	64.6 ± 0.9	-17.9 ± 0.3	28.8 ± 1.0
$w(k)$ - eq. 3	19.4 ± 0.6	63.1 ± 1.4	-17.4 ± 0.4	27.2 ± 0.7
ℓ_{ul} - eq. 4	17.1 ± 0.7	57.6 ± 1.5	-14.0 ± 0.1	21.8 ± 0.6
ℓ_{dst} - eq. 1	17.7 ± 0.8	58.1 ± 1.5	-15.9 ± 0.9	22.7 ± 1.0
Full Method	19.7 ± 0.8	63.3 ± 2.1	-18.2 ± 0.7	24.9 ± 0.6

Fig. 1: Ω curves showing task number t on the x-axis and $A_{t,1:t}$ on the y-axis.

(a) Ω curve for Table Ia (b) Ω curve for Table Ib

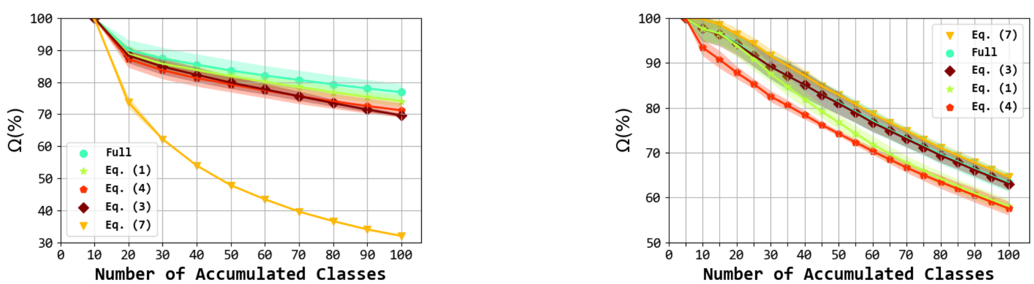


TABLE II: Hyperparameters, chosen with grid search

Coreset		Yes		No	
Hyperparameter	Range	DM	GD	DM	GD
Learning Rate	5e-3, 1e-2, 5e-2, 1e-1, 5e-1	1e-1	1e-1	1e-1	5e-3
Weight FixMatch Loss	0.1, 0.5, 1, 5	1.0		1.0	
TPR	0.01, 0.05, 0.1, 0.2, 0.5, 0.8, 0.95	0.05	-	0.5	-
ϵ (Fix Match)	0.7, 0.85, 0.9, 0.95	0.9	-	0.9	-

use stochastic gradient decent with 0.9 momentum and 0.0005 L2 weight decay.

As also done in [2], we hold λ_{dst} to a constant value, 1, and include a small temperature scaling, 2, for the softmax activations used in eq. 1. All results are averaged over 3 repeats and generated with a common deep learning architecture (WRN-28-2) [3]. Results were generated using a combination of Titan X and 2080 Ti GPUs. Although we did not record specific run-times here as they are machine specific, we find our method to have a similar run-time to GD.

C. Hyperparameter Selection

We tuned hyperparameters using a grid search. We did this for two scenarios: (i) RandomClass Tasks with Uniform Unlabeled Data Distribution and (ii) ParentClass Tasks with PositiveSuperclass Unlabeled Data Distribution. The former is applied for all experimental scenarios which do not include a coreset, and the latter is applied for all scenarios which do include a coreset. We chose this division as we found the coreset size to greatly affect the other hyperparameters. DR and E2E use hyperparameters chosen for GD (as done in [2]), while Base uses hyperparameters from DM. The hyperparameters were tuned using k-fold cross validation with three folds of the training data on only half of the tasks. We do not tune hyperparameters on the full task set because tuning hyperparameters with hold out data from all tasks may violate the principal of continual learning that states each task in visited only once [4]. The results reported outside of this section are on the CIFAR-100 testing split (defined in the dataset).

D. Full Results

We provide additional detail to the results from the main text by reporting (i) the original results with additional metrics and standard deviations (Tables III,IV, and V) and (ii) Ω curves for each experiment in Figures 2 and 3.

TABLE III: Full results (%) on CIFAR-100 with 20% Labeled Data. Results are reported as an average of 3 runs with standard deviation. The results from these tables do not include a coreset (and use the same set of hyperparameters, as described in SM-C)

(a) RandomClass Tasks with Uniform Unlabeled Data Distribution, 5 Tasks

Metric	A_N (↑)	Ω (↑)	BWT (↑)	FGT (↓)
Base	15.6 ± 0.9	52.5 ± 2.5	-25.7 ± 26.2	43.8 ± 2.3
E2E	12.5 ± 0.9	46.1 ± 0.9	1.4 ± 0.6	42.5 ± 1.2
DR	16.0 ± 0.9	53.7 ± 0.7	0.3 ± 0.7	41.6 ± 1.5
GD	32.1 ± 0.2	69.9 ± 0.9	0.5 ± 0.8	5.0 ± 0.3
DM	44.8 ± 1.4	84.4 ± 3.0	2.5 ± 0.1	1.2 ± 0.1

(b) RandomClass Tasks with Uniform Unlabeled Data Distribution, 10 Tasks

Metric	A_N (↑)	Ω (↑)	BWT (↑)	FGT (↓)
Base	8.2 ± 0.1	34.7 ± 0.8	-32.2 ± 24.6	56.2 ± 2.0
E2E	7.5 ± 0.5	32.3 ± 0.6	-0.5 ± 0.4	56.0 ± 1.8
DR	8.3 ± 0.3	36.4 ± 0.2	-1.9 ± 0.3	57.4 ± 1.3
GD	21.4 ± 0.6	60.0 ± 1.9	-14.6 ± 0.1	18.4 ± 1.5
DM	37.5 ± 0.7	76.9 ± 2.5	-1.0 ± 1.0	6.5 ± 0.5

(c) RandomClass Tasks with Uniform Unlabeled Data Distribution, 20 Tasks

Metric	A_N (↑)	Ω (↑)	BWT (↑)	FGT (↓)
Base	4.3 ± 0.4	22.0 ± 0.8	-41.6 ± 13.8	69.4 ± 0.5
E2E	4.0 ± 0.3	21.1 ± 0.6	-4.1 ± 0.8	67.7 ± 1.4
DR	4.3 ± 0.4	22.4 ± 0.7	-7.1 ± 0.2	70.6 ± 1.2
GD	13.4 ± 1.9	42.7 ± 1.1	-29.2 ± 3.5	37.4 ± 0.8
DM	21.1 ± 1.0	60.8 ± 0.8	-8.8 ± 0.7	17.3 ± 1.7

(d) ParentClass Tasks with Uniform Unlabeled Data Distribution, 20 Tasks

Metric	A_N (↑)	Ω (↑)	BWT (↑)	FGT (↓)
Base	3.5 ± 0.1	18.5 ± 0.5	-33.5 ± 6.0	54.3 ± 0.8
E2E	3.2 ± 0.2	18.1 ± 0.6	-14.6 ± 3.5	53.0 ± 0.1
DR	3.7 ± 0.1	19.4 ± 0.6	-17.6 ± 1.3	56.6 ± 0.1
GD	10.5 ± 0.2	37.4 ± 1.8	-25.1 ± 0.1	29.1 ± 0.8
DM	20.8 ± 0.8	57.8 ± 1.4	-10.8 ± 0.8	14.8 ± 0.3

TABLE IV: Full results (%) on CIFAR-100 with 20% Labeled Data. Results are reported as an average of 3 runs with standard deviation. The results from these tables are with a 400 image coreset (and use the same set of hyperparameters, as described in SM-C)

(a) ParentClass Tasks with Uniform Unlabeled Data Distribution, 20 Tasks

Metric	A_N (↑)	Ω (↑)	BWT (↑)	FGT (↓)
Base	14.6 ± 1.4	53.4 ± 2.4	-14.7 ± 6.4	29.8 ± 0.6
E2E	19.5 ± 0.9	59.3 ± 1.7	-14.5 ± 0.2	23.1 ± 0.5
DR	20.1 ± 0.8	57.8 ± 1.5	-15.2 ± 0.4	31.9 ± 3.3
GD	21.4 ± 0.9	57.7 ± 1.8	-12.5 ± 0.4	8.0 ± 1.7
DM	24.4 ± 0.4	67.5 ± 1.3	-15.1 ± 1.3	21.9 ± 1.5

(b) ParentClass Tasks with PositiveSuperclass Unlabeled Data Distribution, 20 Tasks

Metric	A_N (↑)	Ω (↑)	BWT (↑)	FGT (↓)
Base	14.6 ± 1.4	53.4 ± 2.4	-14.7 ± 6.4	29.8 ± 0.6
E2E	18.9 ± 1.2	59.4 ± 1.3	-16.6 ± 1.0	22.2 ± 0.3
DR	18.8 ± 1.0	62.8 ± 1.7	-17.6 ± 0.7	27.5 ± 0.3
GD	17.9 ± 0.8	50.2 ± 0.8	-10.6 ± 0.8	-2.1 ± 2.0
DM	19.7 ± 0.8	63.3 ± 2.1	-18.2 ± 0.7	24.9 ± 0.6

(c) ParentClass Tasks with NegativeSuperclass Unlabeled Data Distribution, 20 Tasks

Metric	A_N (↑)	Ω (↑)	BWT (↑)	FGT (↓)
Base	14.6 ± 1.4	53.4 ± 2.4	-14.7 ± 6.4	29.8 ± 0.6
E2E	19.9 ± 1.2	60.1 ± 0.5	-16.1 ± 1.0	22.5 ± 0.4
DR	20.1 ± 1.9	62.1 ± 1.8	-16.8 ± 0.2	28.7 ± 1.0
GD	18.1 ± 0.6	50.5 ± 0.7	-10.9 ± 1.2	-1.7 ± 1.6
DM	20.7 ± 1.5	64.8 ± 1.3	-17.4 ± 0.7	24.7 ± 1.3

(d) ParentClass Tasks with Random Unlabeled Data Distribution, 20 Tasks

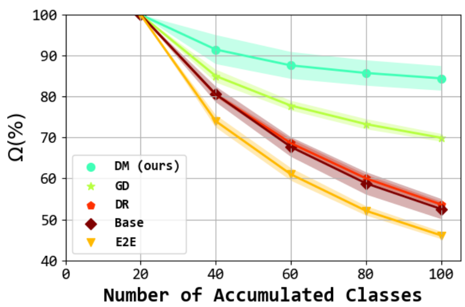
Metric	A_N (↑)	Ω (↑)	BWT (↑)	FGT (↓)
Base	14.6 ± 1.4	53.4 ± 2.4	-14.7 ± 6.4	29.8 ± 0.6
E2E	19.8 ± 0.5	60.0 ± 1.5	-15.1 ± 0.3	23.7 ± 0.6
DR	19.9 ± 1.7	61.8 ± 1.2	-15.7 ± 0.6	29.9 ± 1.6
GD	21.3 ± 0.5	59.9 ± 0.5	-13.7 ± 0.2	8.3 ± 2.7
DM	22.4 ± 1.3	65.1 ± 1.8	-16.1 ± 0.3	23.3 ± 0.9

TABLE V: Full results (%) on Tiny-ImageNet with 20% Labeled Data for RandomClass Tasks with Uniform Unlabeled Data Distribution (10 Tasks, no Coreset). Results are reported as an average of 3 runs with standard deviation. The results from this table use the set of hyperparameters described in SM-C

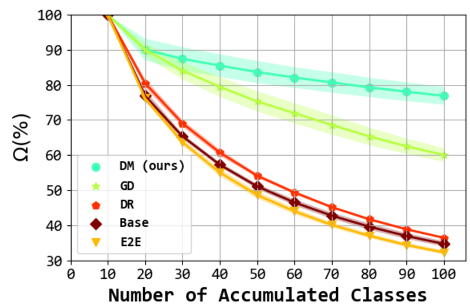
Metric	A_N (↑)	Ω (↑)	BWT (↑)	FGT (↓)
UB	40.7 ± 0.3	100.0 ± 0.0	3.8 ± 0.5	5.2 ± 0.5
Base	6.5 ± 0.6	35.1 ± 1.5	-10.4 ± 2.4	45.1 ± 2.9
E2E	5.8 ± 0.6	30.3 ± 1.9	0.9 ± 0.6	39.3 ± 3.1
DR	6.8 ± 0.4	35.3 ± 1.1	-1.7 ± 0.7	45.0 ± 2.7
GD	11.9 ± 1.3	50.6 ± 2.9	-17.4 ± 2.6	12.5 ± 1.3
DM	24.8 ± 0.7	74.7 ± 1.6	-5.9 ± 0.4	7.6 ± 0.1

Fig. 2: Ω curves showing task number t on the x-axis and Ω up to task t on the y-axis

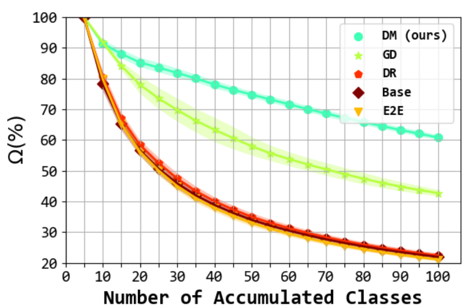
(a) Ω curve for Table IIIa



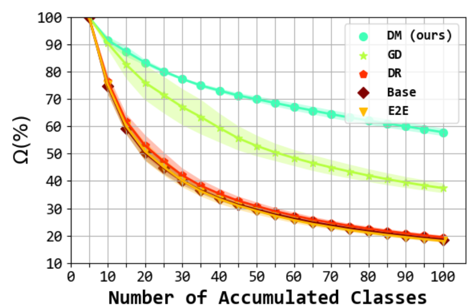
(b) Ω curve for Table IIIb



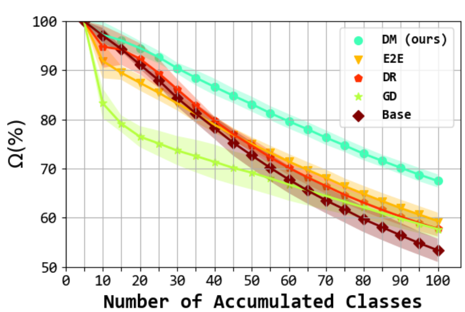
(c) Ω curve for Table IIIc



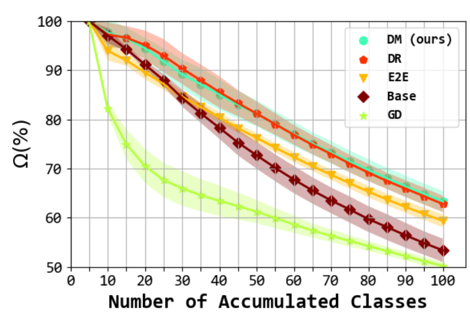
(d) Ω curve for Table IIId



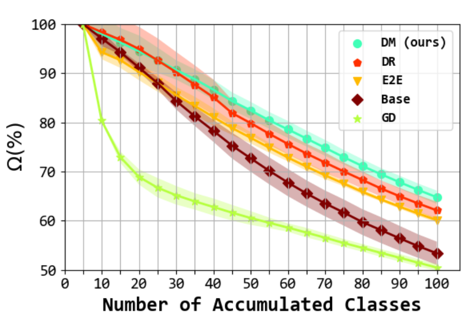
(e) Ω curve for Table IVa



(f) Ω curve for Table IVb



(g) Ω curve for Table IVc



(h) Ω curve for Table IVd

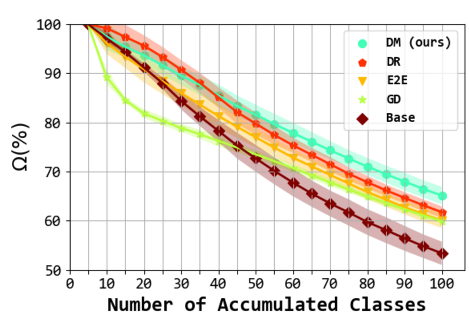


Fig. 3: Ω curve for Table V showing task number t on the x-axis and Ω up to task t on the y-axis

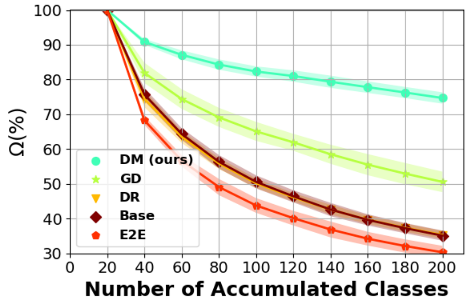
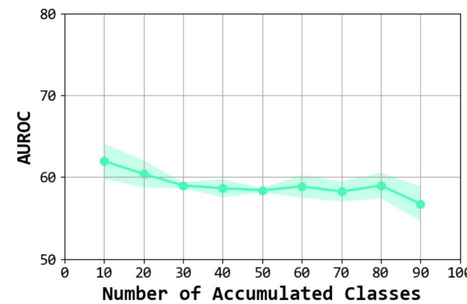
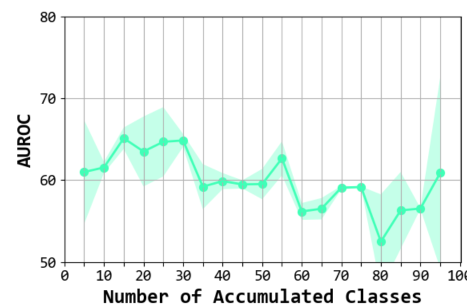


Fig. 4: AUROC over time for DM showing task number t on the x-axis and AUROC on the y-axis

(a) RandomClass Tasks with Uniform Unlabeled Data Distribution



(b) ParentClass Tasks with PositiveSuperclass Unlabeled Data Distribution



E. Performance of OOD Detection

We show AUROC (a metric for OoD detection) over time for DM in both RandomClass Tasks with Uniform Unlabeled Data Distribution (Figure 4a) and ParentClass Tasks with PositiveSuperclass Unlabeled Data Distribution (Figure 4b). A high AUROC means the distributions of the ID data and OoD data are separable. As we can see, AUROC is decreasing over time. In the RandomClass scenario, this is a smooth decline (as expected). In the ParentClass scenario, the decline is not smooth, likely due to the correlations between tasks making the task difficulty highly deviate between runs.

F. Super class and parent class associations for CIFAR-100

We visualize example streams for each task sequence in (Figure 5). As a reminder, we use the following terminology to describe the correlations of the tasks (i.e. labeled data): *RandomClass Tasks*, where no correlations exist in task classes, and *ParentClass Tasks*, where tasks are introduced by CIFAR-100 parent classes (i.e. each task is to learn the five classes of a single CIFAR-100 parent class). For the unlabeled data distribution we have: *Uniform Unlabeled*, where all classes are uniformly dsitributed in unlabeled data for all tasks, *PositiveSuperclass Unlabeled*, where the unlabeled data of each tasks consists of the parent classes in the same super-class as the current task, *NegativeSuperclass Unlabeled*, where the unlabeled data of each tasks consists of parent classes from different super-class as the current task, and *RandomUnlabeled*, where the unlabeled data of each task consists of 20 randomly sampled classes (roughly equal to the average class size in a super-class). We also show the relationship between super classes and parent classes for CIFAR-100 (Figure 6) as defined by [5].

G. Additional Studies

We found that confidence calibration in GD [2] had mixed effects in our experiments. We ablate this contribution for RandomClass Tasks with Uniform Unlabeled Data Distribution (Table VIa), ParentClass Tasks with PositiveSuperclass Unlabeled Data Distribution (Table VIb), and ParentClass Tasks with Random Unlabeled Data Distribution (Table VIc). We contribute this finding to the assumption made in GD that the unlabeled data does not contain data from the current task (which is heavily violated in some of our experiments). Even though removing this mechanism can boost GD performance for some of the experiments (Tables VIa and VIb) and makes it worse for others (Table VIc), it is still significantly below our method (DM) in each case.

H. Additional Background and Related Work

Continual Learning Approaches: Approaches to mitigate catastrophic forgetting in continual learning can be broadly organized into three types: *rehearsal*, *architectural*, and *regularization* [6]. Rehearsal methods include storage to "replay" data or experiences from previous tasks to mitigate catastrophic forgetting [1, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17]. Rather than storing raw data, some methods train a generative model [18, 19, 20] or replay compressed data representations in a late layer [21]. Architectural approaches typically avoid overwriting the current model by expanding the model parameters to make room for knowledge related to novel tasks [22, 23, 24, 25, 26]. Finally, regularization approaches focus on penalizing changes to parameters important to past tasks. Approaches include regularization penalties [27, 28, 29, 30, 31], meta learning [32], model compression [33, 34, 35], or knowledge distillation [2, 36, 37, 38].

Semi-Supervised Learning: Semi-supervised learning leverages plentiful available unlabeled data to boost model perfor-

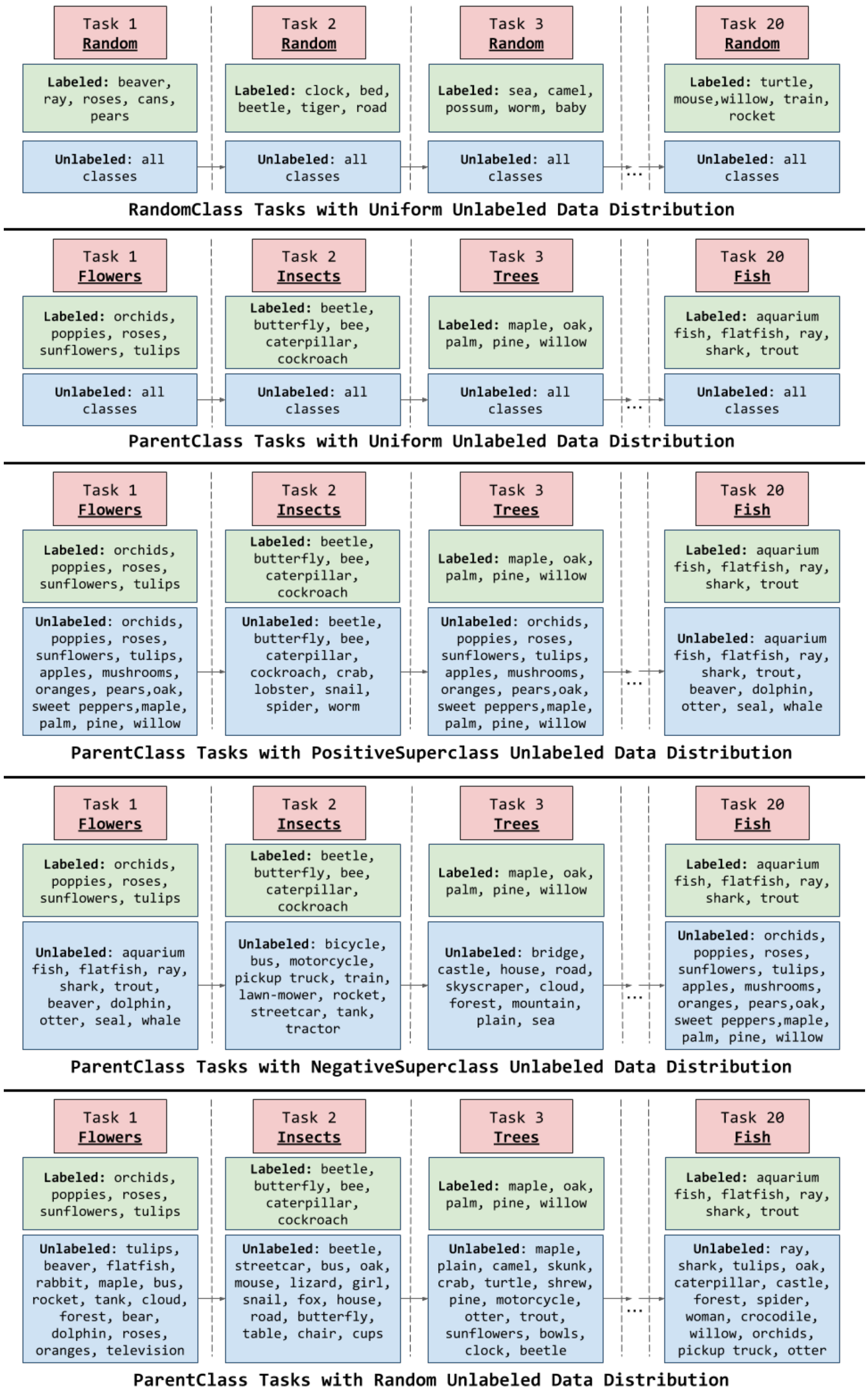


Fig. 5: Example streams for each task sequence

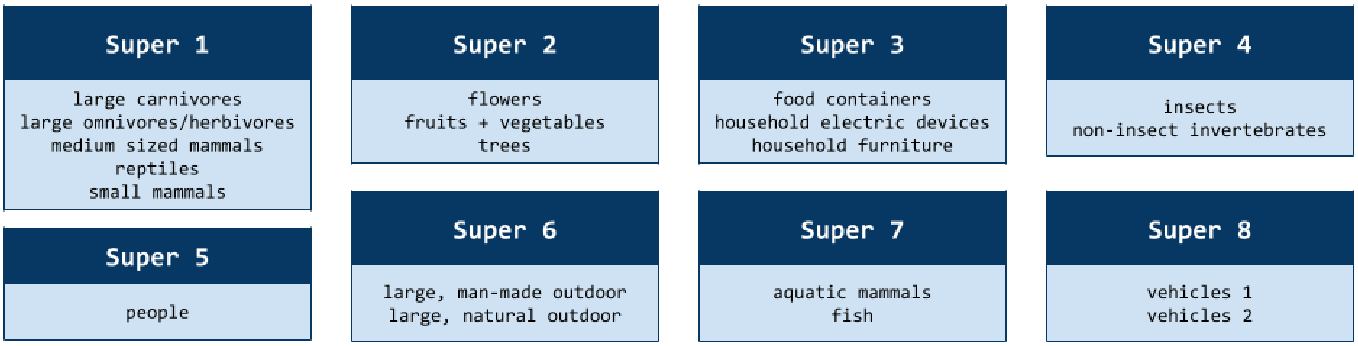


Fig. 6: Super-parent class relationships for CIFAR-100

TABLE VI: Results (%) for GD Confidence Calibration Ablation on CIFAR-100 with 20% Labeled Data. Results are reported as an average of 3 runs with mean and standard deviation.

(a) RandomClass Tasks with Uniform Unlabeled Data Distribution, 10 Tasks, no Coreset

Confidence Calibration	A_N	Ω	BWT	FGT
✓	21.4 ± 0.6	60.0 ± 1.9	-14.6 ± 0.1	18.4 ± 1.5
	23.7 ± 1.2	67.0 ± 3.1	-5.5 ± 1.8	20.3 ± 2.0

(b) ParentClass Tasks with PositiveSuperclass Unlabeled Distribution, 20 Tasks, 400 image coreset

Confidence Calibration	A_N	Ω	BWT	FGT
✓	17.9 ± 0.8	50.2 ± 0.8	-10.6 ± 0.8	-2.1 ± 2.0
	19.5 ± 0.4	54.4 ± 3.8	-12.6 ± 1.0	7.2 ± 3.5

(c) ParentClass Tasks with Random Unlabeled Distribution, 20 Tasks, 400 image coreset

Confidence Calibration	A_N	Ω	BWT	FGT
✓	21.3 ± 0.5	59.9 ± 0.5	-13.7 ± 0.2	8.3 ± 2.7
	18.1 ± 0.9	54.1 ± 0.7	-12.0 ± 1.2	20.3 ± 2.8

mance when given a (typically small) amount of labeled data. Semi-supervised learning is popular because labeling large datasets is an expensive process. A simple yet popular technique is to provide pseudo-labels [39] for *confident* unlabeled data based on the current model's predictions and to treat this pair (the unlabeled data and pseudo-label) as if it were a labeled data pair. Many following methods build on this idea of using predictions on the unlabeled data to boost performance. For example, mean teachers [40] involve averaging model weights for a temporal ensembling approach which encourages consistent label predictions over time. Virtual Adversarial Training (VAT) smooths the decision boundary around each unlabeled data point to be robust against adversarial perturbations. More recent methods include MixMatch [41], which involves using low-entropy labels and strong data augmentations for a Mix-Up loss, and FixMatch [42], which enforces consistent labeling between weakly and strongly augmented versions of unlabeled data. Other approaches for leveraging unlabeled data is to use it for an auxiliary loss such as generative loss [43, 44] or self-supervised learning [45]. The reader is referred to [46] for a recent survey of popular techniques and evaluations.

References

[1] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, (USA), pp. 6470–6479, Curran Associates Inc., 2017.

[2] K. Lee, K. Lee, J. Shin, and H. Lee, "Overcoming catastrophic forgetting with unlabeled data in the wild," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 312–321, 2019.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[4] G. M. van de Ven and A. S. Tolias, "Three scenarios for continual learning," *arXiv preprint arXiv:1904.07734*, 2019.

[5] X. Zhu and M. Bain, “B-cnn: branch convolutional neural network for hierarchical classification,” *arXiv preprint arXiv:1709.09890*, 2017.

[6] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, “Continual lifelong learning with neural networks: A review,” *Neural Networks*, vol. 113, pp. 54 – 71, 2019.

[7] R. Aljundi, E. Belilovsky, T. Tuytelaars, L. Charlin, M. Caccia, M. Lin, and L. Page-Caccia, “Online continual learning with maximal interfered retrieval,” in *Advances in Neural Information Processing Systems*, pp. 11849–11860, 2019.

[8] R. Aljundi, M. Lin, B. Goujaud, and Y. Bengio, “Gradient based sample selection for online continual learning,” in *Advances in Neural Information Processing Systems*, pp. 11816–11825, 2019.

[9] A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny, “Efficient lifelong learning with a-GEM,” in *International Conference on Learning Representations*, 2019.

[10] A. Chaudhry, M. Rohrbach, M. Elhoseiny, T. Ajanthan, P. K. Dokania, P. H. Torr, and M. Ranzato, “Continual learning with tiny episodic memories,” *arXiv preprint arXiv:1902.10486*, 2019.

[11] A. Gepperth and C. Karaoguz, “Incremental learning with self-organizing maps,” *2017 12th International Workshop on Self-Organizing Maps and Learning Vector Quantization, Clustering and Data Visualization (WSOM)*, pp. 1–8, 2017.

[12] T. L. Hayes, N. D. Cahill, and C. Kanan, “Memory efficient experience replay for streaming learning,” in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 9769–9776, IEEE, 2019.

[13] R. Kemker, M. McClure, A. Abitino, T. Hayes, and C. Kanan, “Measuring catastrophic forgetting in neural networks,” *AAAI Conference on Artificial Intelligence*, 2018.

[14] S. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, “icarl: Incremental classifier and representation learning,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR’17, pp. 5533–5542, 2017.

[15] A. Robins, “Catastrophic forgetting, rehearsal and pseudorehearsal,” *Connection Science*, vol. 7, no. 2, pp. 123–146, 1995.

[16] D. Rolnick, A. Ahuja, J. Schwarz, T. Lillicrap, and G. Wayne, “Experience replay for continual learning,” in *Advances in Neural Information Processing Systems*, pp. 348–358, 2019.

[17] J. von Oswald, C. Henning, J. Sacramento, and B. F. Grewe, “Continual learning with hypernetworks,” *arXiv preprint arXiv:1906.00695*, 2019.

[18] N. Kamra, U. Gupta, and Y. Liu, “Deep generative dual memory network for continual learning,” *arXiv preprint arXiv:1710.10368*, 2017.

[19] R. Kemker and C. Kanan, “Fearnet: Brain-inspired model for incremental learning,” *International Conference on Learning Representations (ICLR)*, 2018.

[20] H. Shin, J. K. Lee, J. Kim, and J. Kim, “Continual learning with deep generative replay,” in *Advances in Neural Information Processing Systems 30* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), pp. 2990–2999, Curran Associates, Inc., 2017.

[21] X. Liu, C. Wu, M. Menta, L. Herranz, B. Raducanu, A. D. Bagdanov, S. Jui, and J. van de Weijer, “Generative feature replay for class-incremental learning,” *arXiv preprint arXiv:2004.09199*, 2020.

[22] S. Ebrahimi, F. Meier, R. Calandra, T. Darrell, and M. Rohrbach, “Adversarial continual learning,” *arXiv preprint arXiv:2003.09553*, 2020.

[23] S. Lee, J. Ha, D. Zhang, and G. Kim, “A neural dirichlet process mixture model for task-free continual learning,” *arXiv preprint arXiv:2001.00689*, 2020.

[24] V. Lomonaco and D. Maltoni, “Core50: a new dataset and benchmark for continuous object recognition,” *arXiv preprint arXiv:1705.03550*, 2017.

[25] D. Maltoni and V. Lomonaco, “Continuous learning in single-incremental-task scenarios,” *Neural Networks*, vol. 116, pp. 56–73, 2019.

[26] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, “Progressive neural networks,” *arXiv preprint arXiv:1606.04671*, 2016.

[27] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars, “Memory aware synapses: Learning what (not) to forget,” in *ECCV*, 2018.

[28] S. Ebrahimi, M. Elhoseiny, T. Darrell, and M. Rohrbach, “Uncertainty-guided continual learning with bayesian neural networks,” *arXiv preprint arXiv:1906.02425*, 2019.

[29] 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*, 2017.

[30] M. K. Titsias, J. Schwarz, A. G. d. G. Matthews, R. Pascanu, and Y. W. Teh, “Functional regularisation for continual learning with gaussian processes,” in *International Conference on Learning Representations*, 2019.

[31] F. Zenke, B. Poole, and S. Ganguli, “Continual learning through synaptic intelligence,” in *International Conference on Machine Learning*, 2017.

[32] K. Javed and M. White, “Meta-learning representations for continual learning,” in *Advances in Neural Information Processing Systems*, pp. 1818–1828, 2019.

[33] S. Beaulieu, L. Frati, T. Miconi, J. Lehman, K. O. Stanley, J. Clune, and N. Cheney, “Learning to continually learn,” *arXiv preprint arXiv:2002.09571*, 2020.

[34] X. He, J. Sygnowski, A. Galashov, A. A. Rusu, Y. W. Teh, and R. Pascanu, “Task agnostic continual learning via meta learning,” *arXiv preprint arXiv:1906.05201*, 2019.

[35] G. Saha, I. Garg, A. Ankit, and K. Roy, “Structured compression and sharing of representational space for

continual learning," *arXiv preprint arXiv:2001.08650*, 2020.

[36] F. M. Castro, M. J. Marín-Jiménez, N. Guil, C. Schmid, and K. Alahari, "End-to-end incremental learning," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 233–248, 2018.

[37] S. Hou, X. Pan, C. Change Loy, Z. Wang, and D. Lin, "Lifelong learning via progressive distillation and retrospection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 437–452, 2018.

[38] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.

[39] D.-H. Lee, "Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks," *ICML 2013 Workshop : Challenges in Representation Learning (WREPL)*, 07 2013.

[40] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Advances in Neural Information Processing Systems 30* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), pp. 1195–1204, Curran Associates, Inc., 2017.

[41] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, "Mixmatch: A holistic approach to semi-supervised learning," in *Advances in Neural Information Processing Systems*, pp. 5050–5060, 2019.

[42] K. Sohn, D. Berthelot, C.-L. Li, Z. Zhang, N. Carlini, E. D. Cubuk, A. Kurakin, H. Zhang, and C. Raffel, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," *arXiv preprint arXiv:2001.07685*, 2020.

[43] D. P. Kingma, D. J. Rezende, S. Mohamed, and M. Welling, "Semi-supervised learning with deep generative models," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, (Cambridge, MA, USA), pp. 3581–3589, MIT Press, 2014.

[44] J. T. Springenberg, "Unsupervised and semi-supervised learning with categorical generative adversarial networks," *arXiv preprint arXiv:1511.06390*, 2015.

[45] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[46] A. Oliver, A. Odena, C. A. Raffel, E. D. Cubuk, and I. Goodfellow, "Realistic evaluation of deep semi-supervised learning algorithms," in *Advances in Neural Information Processing Systems*, pp. 3235–3246, 2018.