Always Be Dreaming: A New Approach for Data-Free Class-Incremental Learning -Supplementary Materials (Appendix)-

James Smith[1]*, Yen-Chang Hsu[2], Jonathan Balloch[1], Yilin Shen[2], Hongxia Jin[2], Zsolt Kira[1]
[1]Georgia Institute of Technology, [2]Samsung Research America

A. Additional Results

In this section, we present additional experiments and results. For an alternative view of all results, we show Ω plotted by task in Figure A.

In Tables A/B, we expand our CIFAR-100 results with two additional methods: (1) LwF.MC [9], a more powerful variant of LWF designed for class-incremental learning, and (2) End-to-End Incremental Learning [2] (E2E). In our implementation of E2E, we use the same data augmentations as our other experiments for a fair comparison. As previously published [12], we see that E2E performs slightly worse than BiC and LwF.MC strongly outperforms LWF. Our approach consistently outperforms LwF.MC.

We also report additional results on the Tiny-ImageNet dataset [5] in Tables C/D, which contains 200 classes of 64x64 resolution images with 500 training images per class. We use the same experiment settings as CIFAR-100 with 10 classes per task and 20 tasks total. This is a highly challenging dataset with a low upper bound performance (drops from 69.9% to 55.5%), but we arrive at the same conclusions as we did for our CIFAR-100 experiments: our method outperforms all data-free class-incremental learning approaches, and performs slightly worse than state-of-the-art approaches which store 2000 images for replay. Importantly, the number of parameters stored for replay in these experiments (2000*64*64*3 = 2.5e7) *far exceeds* the number of parameters *temporarily* stored for synthesizing images (8.5e6). Note that this memory usage in our method can be completely removed at the cost of additional computation. Despite requiring only 10 times fewer parameters to store (and not storing *any* training data), our method performs reasonably close to state-of-the-art.

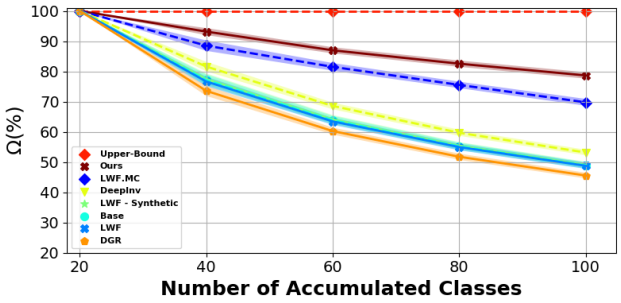
Finally, we expand the main paper results in Table E to include LWF.MC. Our method and LWF.MC perform similarly, indicating that more work is needed to scale our approach to large 224x224x3 images. This is not surprising because prior work [7] requires **1 generator per class** to scale data-free generative distillation up to ImageNet. We do not have the computational resources to perform this (e.g., full 1000 class ImageNet would require 1000 generators). Instead, our work demonstrates the need for generative data-free knowledge distillation to be *efficiently* scaled up to the 224x224x3 images of ImageNet. We leave this to future work. We kindly acknowledge that recent works which replay from a generator (close to our setting) also use small variants of ImageNet in their experiments [1, 3, 8].

B. Additional Baseline Diagnosis with MMD

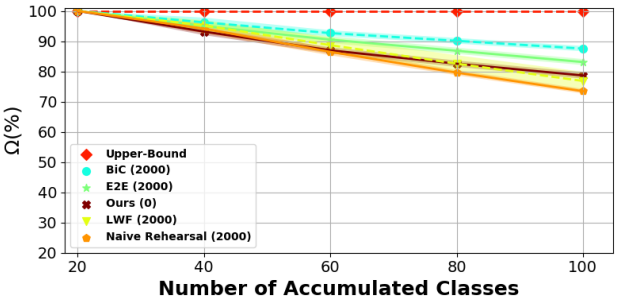
In Section 5, we analyze *representational distance* between embedded features with a metric that captures the distance between mean embedded images of two distribution samples. This metric is Mean Image Distance (MID) and is calculated with a reference sample of images x_a and another sample of images x_b, where a *high score* indicates *dissimilar* features and a *low score* indicates *similar* features. In this section, we repeat the Section 5 experiments with the commonly used unbiased Maximum Mean Discrepancy (MMD) [4], which gives the distance between embeddings of two distributions in a reproducing kernel Hilbert space.

As done in Section 5, we start by training our model for the first two tasks in the ten-task CIFAR-100 benchmark. We calculate MMD between feature embeddings of real task 1 data and real task 2 data, and then we calculate MMD between feature embeddings of real task 1 data and synthetic task 1 data. The results are reported in Figure B. For (a) DeepInversion, the MMD score between real task 1 data and synthetic task 1 data is significantly higher than the MMD score between real task 1 data and real task 2 data. As found in Section 5, this indicates that the embedding space prioritizes *domain* over *semantics*, which is detrimental because the classifier will learn the decision boundary between synthetic task 1 and real task 2, introducing great classification error with real task 1 images. For (b) our method, the MMD score between real task 1 data and synthetic task 1 data is much lower, indicating that our feature embedding prioritizes *semantics* over *domain*.

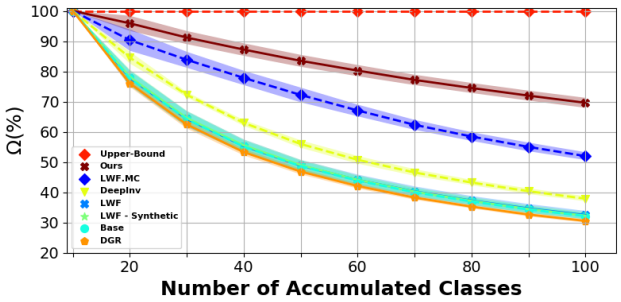
*Correspondence to: James Smith jamessealesmith@gatech.edu



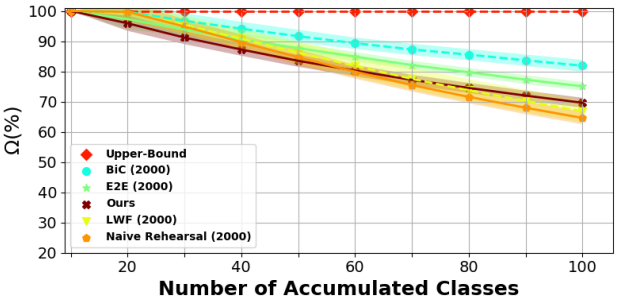
(a) Ω curve for five task CIFAR-100 (without coreset)



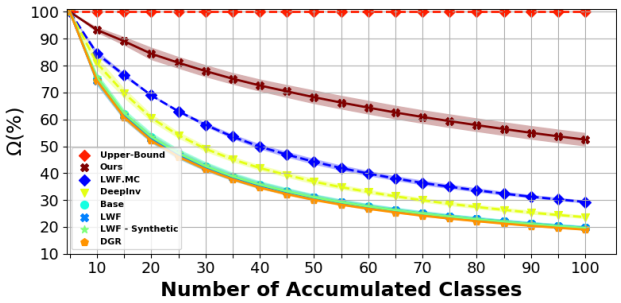
(b) Ω curve for five task CIFAR-100 (with coreset)



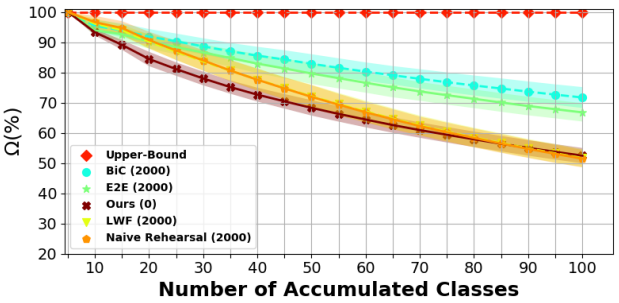
(c) Ω curve for ten task CIFAR-100 (without coreset)



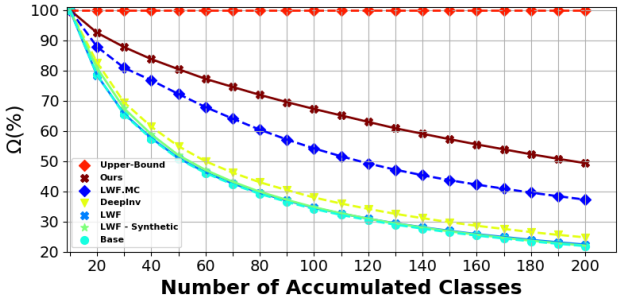
(d) Ω curve for ten task CIFAR-100 (with coreset)



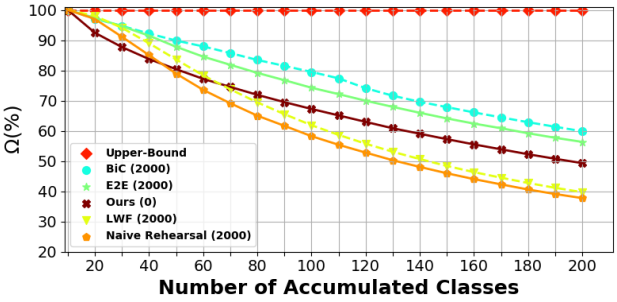
(e) Ω curve for twenty task CIFAR-100 (without coreset)



(f) Ω curve for twenty task CIFAR-100 (with coreset)



(g) Ω curve for twenty task Tiny ImageNet (without coreset)



(h) Ω curve for twenty task Tiny ImageNet (with coreset)

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

Table A: Full Results (%) for *data-free* class-incremental learning on CIFAR-100 for various numbers of tasks (5, 10, 20). Results are reported as an average of 3 runs.

Tasks		5		10		20	
Method	Replay Data	A_N (↑)	Ω (↑)	A_N (↑)	Ω (↑)	A_N (↑)	Ω (↑)
Upper Bound	None	69.9 ± 0.2	100.0 ± 0.0	69.9 ± 0.2	100.0 ± 0.0	69.9 ± 0.2	100.0 ± 0.0
Base	None	16.4 ± 0.4	48.9 ± 1.1	8.8 ± 0.1	32.1 ± 1.1	4.4 ± 0.3	19.7 ± 0.7
LwF [6]	None	17.0 ± 0.1	49.5 ± 0.1	9.2 ± 0.0	33.3 ± 0.9	4.7 ± 0.1	20.1 ± 0.3
LwF.MC [9]	None	32.5 ± 1.0	69.8 ± 1.1	17.1 ± 0.1	52.0 ± 1.3	7.7 ± 0.5	29.3 ± 0.6
DGR [10]	Generator	14.4 ± 0.4	45.5 ± 0.9	8.1 ± 0.1	30.5 ± 0.6	4.1 ± 0.3	19.0 ± 0.3
LwF [6]	Synthetic	16.7 ± 0.1	49.8 ± 0.1	8.9 ± 0.0	32.3 ± 0.0	4.7 ± 0.0	19.7 ± 0.0
DeepInversion [13]	Synthetic	18.8 ± 0.3	53.2 ± 0.9	10.9 ± 0.6	37.9 ± 0.8	5.7 ± 0.3	23.6 ± 0.7
Ours	Synthetic	$\mathbf{43.9 \pm 0.9}$	$\mathbf{78.6 \pm 1.1}$	$\mathbf{33.7 \pm 1.2}$	$\mathbf{69.6 \pm 1.6}$	$\mathbf{20.0 \pm 1.4}$	$\mathbf{52.5 \pm 2.5}$

Table B: Results (%) for class-incremental learning *with replay data* on CIFAR-100 for various numbers of tasks (5, 10, 20). A coreset of 2000 images is leveraged for replay-based methods, and thus *these methods do not meet problem the DFCIL constraints* (note we report for our method numbers *without* any coreset). Results are reported as an average of 3 runs.

Tasks		5		10		20	
Method	Replay Data	A_N (↑)	Ω (↑)	A_N (↑)	Ω (↑)	A_N (↑)	Ω (↑)
Upper Bound	None	69.9 ± 0.2	100.0 ± 0.0	69.9 ± 0.2	100.0 ± 0.0	69.9 ± 0.2	100.0 ± 0.0
Naive Rehearsal	Coreset	34.0 ± 0.2	73.4 ± 0.8	24.0 ± 1.0	64.6 ± 2.1	14.9 ± 0.7	51.4 ± 2.9
LwF [6]	Coreset	39.4 ± 0.3	79.0 ± 0.0	27.4 ± 0.8	69.4 ± 0.4	16.6 ± 0.4	54.2 ± 2.2
E2E [2]	Coreset	47.4 ± 0.8	83.1 ± 1.0	38.4 ± 1.3	75.0 ± 1.4	32.7 ± 1.9	66.8 ± 3.0
BiC [12]	Coreset	$\mathbf{53.7 \pm 0.4}$	$\mathbf{87.5 \pm 0.9}$	$\mathbf{45.9 \pm 1.8}$	$\mathbf{81.9 \pm 2.0}$	$\mathbf{37.5 \pm 3.2}$	$\mathbf{71.7 \pm 3.4}$
Ours	Synthetic	43.9 ± 0.9	78.6 ± 1.1	33.7 ± 1.2	69.6 ± 1.6	20.0 ± 1.4	52.5 ± 2.5

Table C: Results (%) for *data-free* class-incremental learning on Tiny ImageNet (20 tasks, 5 classes per task). Results are reported for a single run.

Method	Replay Data	A_N (↑)	Ω (↑)
Upper Bound	None	55.5	100.0
Base	None	4.1	21.9
LwF [6]	None	4.4	22.4
LwF.MC [9]	None	8.8	37.2
LwF [6]	Synthetic	4.0	22.0
DeepInversion [13]	Synthetic	5.1	24.8
Ours	Synthetic	12.1	49.3

Table D: Results (%) for class-incremental learning *with replay data* on Tiny ImageNet (20 tasks, 5 classes per task). A coreset of 2000 images is leveraged for replay-based methods, and thus *these methods do not meet problem the DFCIL constraints* (note we report for our method numbers *without* any coreset). Results are reported for a single run.

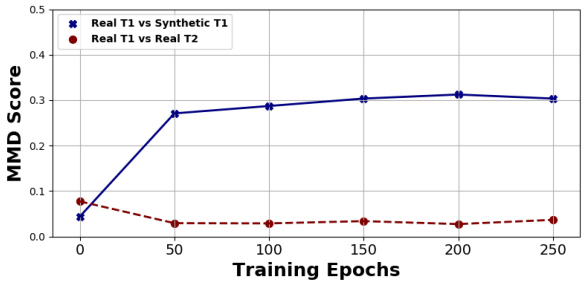
Method	Replay Data	A_N (↑)	Ω (↑)
Upper Bound	None	55.5	100.0
Naive Rehearsal	Coreset	6.6	37.7
LwF [6]	Coreset	6.9	39.7
E2E [2]	Coreset	16.9	56.3
BiC [12]	Coreset	17.4	59.8
Ours	Synthetic	12.1	49.3

Table E: Results (%) for class-incremental learning on five task ImageNet-50. A coreset of 2000 images is leveraged for replay-based methods, and thus *these methods do not meet problem the DFCIL constraints*. Results are reported as a single run.

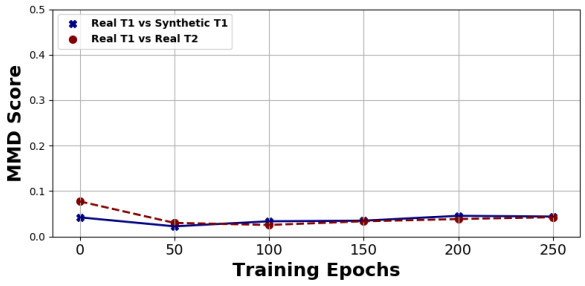
Method	Replay Data	A_N (↑)
Upper Bound	None	89.8
LwF [6]	None	19.4
LwF.MC [9]	None	72.7
Naive Rehearsal	Coreset	78.9
LwF [6]	Coreset	**84.8**
Ours	Synthetic	71.5

Table F: Range and chosen value of our hyperparameters, chosen with grid search

Hyperparam.	Range	Value
α_{con}	1e-1, 1, 1e1	1
α_{div}	1e-1, 1, 1e1	1
α_{stat}	1, 1e1, 5e1, 1e2	5e1
α_{prior}	1e-4, 1e-3, 1e-2, 1e-1, 1	1e-3
α_{temp}	1, 1e1, 1e2, 1e3, 1e4	1e3
λ_{kd}	1e-2, 1e-1, 1	1e-1
λ_{ft}	1e-2, 1e-1, 1	1e-1

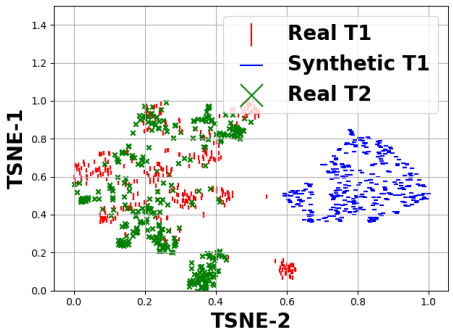


(a) DeepInversion [13]

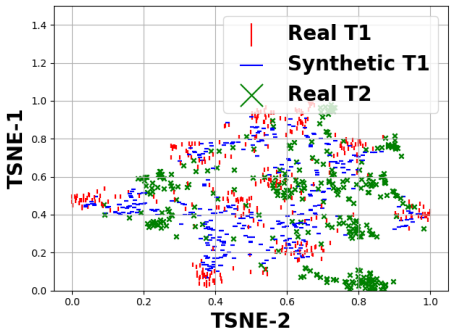


(b) Our Method

Figure B: Maximum Mean Discrepancy (MMD) between feature embeddings of real task 1 data and synthetic task 1 data (blue), real task 2 data (red). Task 1 corresponds to ten classes of CIFAR-100 while task 2 corresponds to a different ten classes of CIFAR-100; the results are generated after training on task 2.



(a)



(b)

Figure C: t-SNE visualizations for (a) Figure 1.a (DeepInversion) and (b) Figure 1.c (Our Method) from the main text.

C. Additional Experiment Details

The majority of experiment details are listed in the main text (Section 7) and are dataset specific. Additionally: (i) we augment training data using standard augmentations such as random horizontal flips and crops, (ii) results were generated using a combination of Titan X and 2080 Ti GPUs, and (iii) synthesized images are sampled from $\mathcal{F}$ at each training step.

D. Hyperparameter Sweeps

We tuned hyperparameters using a grid search. 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 [11]. The results reported outside of this section are on testing splits (defined in the dataset).

E. Discussion of Class Shuffling Seeds

Our results are slightly lower than reported in prior work [9, 12] because we re-implemented each method in our benchmarking environment. A major difference between our implementation and these works is that, instead of using a fixed seed for a single class-order, we instead *randomly shuffle* the class and task order for each experiment run. The class order has a significant effect on the end results, with our top performing class order resulting in performance similar to results reported in [12]. We argue that shuffling the class order gives a better representation of method performance while acknowledging both approaches (shuffling and not shuffling) have merit.

F. t-SNE Visualization

In Figure C, we show real t-SNE visualizations which reasonably approximate Figure 1.a (DeepInversion) and Figure 1.c (Our Method) from the main text. Results are shown after training the second task in the ten-task CIFAR-100 benchmark. *Importantly, the distilling $\theta_{1,1}$ model and the synthetic data are the same for both methods; only the loss functions are different.*

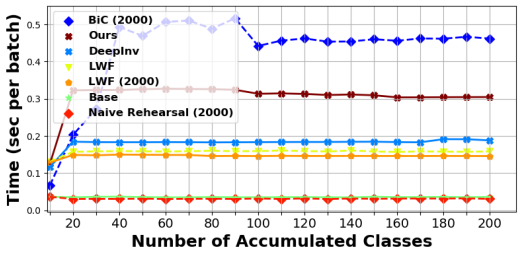


Figure D: Training time for the twenty task Tiny-ImageNet benchmark (Tables C/D).

G. Training Time

In Figure D, we show the training time (seconds per training batch on a single Titain X Pascal GPU) for the twenty task Tiny-ImageNet benchmark (Tables C/D). Our method is faster than the SOTA replay-based method, BIC, yet slower than the other methods. All of these methods produce a model of the same architecture and therefore have the same inference time (except for BIC which has a *very* small logit weighting operation).

References

[1] Ali Ayub and Alan Wagner. {EEC}: Learning to encode and regenerate images for continual learning. In *International Conference on Learning Representations*, 2021. 1

[2] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 233–248, 2018. 1, 3

[3] Yulai Cong, Miaoyun Zhao, Jianqiao Li, Sijia Wang, and Lawrence Carin. Gan memory with no forgetting. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 16481–16494. Curran Associates, Inc., 2020. 1

[4] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012. 1

[5] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7, 2015. 1

[6] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017. 3

[7] Liangchen Luo, Mark Sandler, Zi Lin, Andrey Zhmoginov, and Andrew Howard. Large-scale generative data-free distillation. *arXiv preprint arXiv:2012.05578*, 2020. 1

[8] Oleksiy Ostapenko, Mihai Puscas, Tassilo Klein, Patrick Jahnichen, and Moin Nabi. Learning to remember: A synaptic plasticity driven framework for continual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11321–11329, 2019. 1

[9] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR'17, pages 5533–5542, 2017. 1, 3, 4

[10] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 2990–2999. Curran Associates, Inc., 2017. 3

[11] Gido M van de Ven and Andreas S Tolias. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019. 4

[12] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 374–382, 2019. 1, 3, 4

[13] Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8715–8724, 2020. 3, 4