CLoG: Benchmarking Continual Learning of Image Generation Models

Anonymous Author(s)
Affiliation
Address
email

Abstract

Continual Learning (CL) poses a significant challenge in Artificial Intelligence, aiming to incrementally acquire knowledge and skills. While extensive research has focused on CL within the context of classification tasks, the advent of increasingly powerful generative models necessitates the exploration of Continual Learning of Generative models (CLoG). This paper advocates for shifting the research focus from classification-based CL to CLoG. We systematically identify the unique challenges presented by CLoG compared to traditional classification-based CL. We adapt three types of existing CL methodologies—replay-based, regularization-based, and parameter-isolation-based methods—to generative tasks and introduce comprehensive benchmarks for CLoG that feature great diversity and broad task coverage. Our benchmarks and results yield intriguing insights that can be valuable for developing future CLoG methods. We believe shifting the research focus to CLoG will benefit the CL community and illuminate the path for AI-generated content (AIGC) in a lifelong learning paradigm.

1 Introduction

The development of Artificial Intelligence Generated Content (AIGC) marks a paradigm shift from classification-based applications, such as image recognition [48, 91, 102, 23, 45], to powerful generative models [21, 95, 29, 99]. Continual learning (CL), which involves AI systems incrementally mastering a sequence of tasks $\mathcal{T}^{(1)}, \mathcal{T}^{(2)}, \ldots, \mathcal{T}^{(T)}$ [36, 14], is a crucial challenge in AI research. Currently, advancements on CL [85, 111, 61, 37] mainly lies in *classification-based models* [103] settings. Given the rising importance of generative models, we believe that *now is an opportune time to pivot the research focus towards the Continual Learning of Generative models (**CLoG**)*. CLoG typically necessitates the use of sophisticated generative models, such as VAE [41], GAN [21], score-based models [98], to model the complicated data distributions.

In this paper, we establish a foundational framework for studying CLoG. Initially, we define the problem of CLoG and delve into it by leveraging insights from the existing research on classification-based CL (Section 2). We then meticulously develop benchmarks for CLoG, focusing on task selection (Section 3.1), baseline setup (Section 3.2), metrics design (Section 3.3), and training specifics (Section 3.4). We maintain a clean baseline set by adapting representative CL methods to CLoG, employ unified evaluation metrics, and enhance the efficiency of evaluation by focusing only on the specifics crucial for CL. Our benchmarks provide valuable insights and are intended to inspire further advancements in CLoG methodologies. This paper also aims to reflect on current CL research, inspiring advancing CL methods tailored for foundation models. Our extensible codebase will be released for the benefit of CLoG research.

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

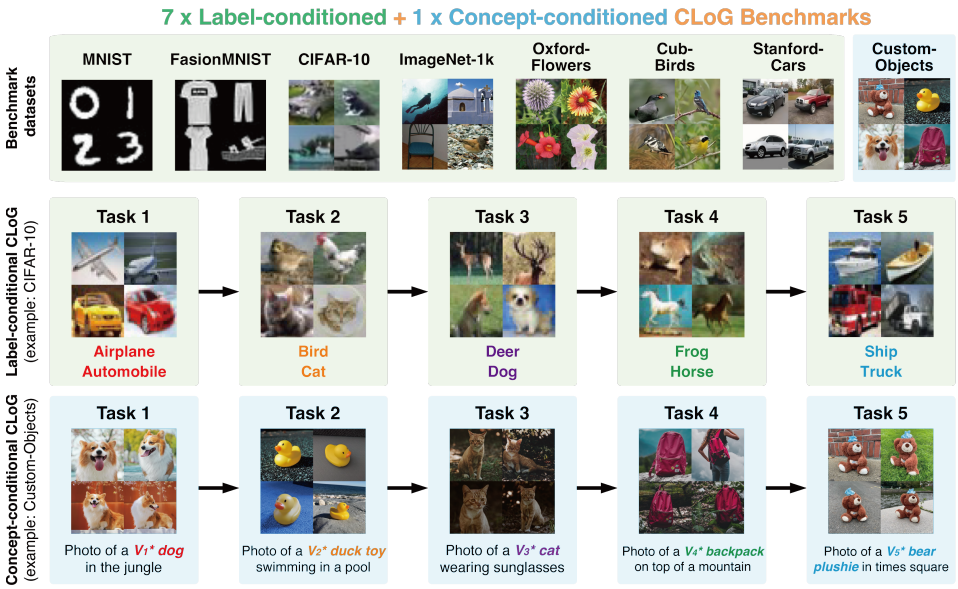


Figure 1: **Overview of benchmarks.** Seven label-conditioned and one concept-conditioned CLoG benchmarks are studied, with details presented in Table 15 and Section 3.1. *Label-conditioned CLoG* learns a sequence of generation tasks conditioned on label indices. *Concept-conditional CLoG* learns to synthesize a sequence of concepts (denoted as V_i^* for the ith concept) given arbitrary text prompts.

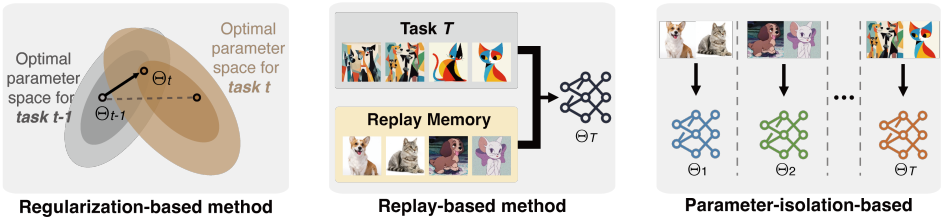


Figure 2: **Overview of baselines.** Three types of CL baselines are adapted to CLoG, which include *regularization-based*, *replay-based*, and *parameter-isolation-based* methods, resulting in a total of twelve different CLoG baselines. The detailed information on the baselines are in Section 3.2.

2 From Traditional CL to CLoG

2.1 Continual Learning

The mathematical formulation of general CL is presented in Appendix A.1. A main assumption of CL is that once a task is learned, its training data $\mathcal{D}^{(t)}$ is no longer accessible (or with limited access). This assumption causes *catastrophic forgetting* for machine learning models, which refers to performance degradation of previous tasks in learning each new task [64]. The existing CL methods to prevent forgetting include: (1) **Regularization-based Methods**: The idea of this family is to add regularization to penalize changes to important parameters learned for previous tasks in learning a new task [44, 122, 3, 6, 53]. (2) **Replay-based Methods**: These methods store a small subset of training data from previous tasks [60, 8, 76, 75, 39] or learn a data generator to synthesize pseudo data [90, 12, 113, 127] of previous tasks. The saved data, the synthesized data, and the new task data are both used in training. (3) **Parameter-isolation Based Methods**: These methods allocate task-specific parameters to prevent subsequent tasks from interfering the previously learned parameters [112, 84, 19, 80, 2, 62, 55].

2.2 Continual learning of generative models (CLoG)

The mathematical formulation of general CL is presented in Appendix A.2. The key difference lies in the input space $\mathcal{X}$ and output space $\mathcal{Y}$. In image generation, the input $\boldsymbol{x}$ may be some label conditions ($e.g.$, one-hot class) or instructions (text or images), and the output $\boldsymbol{y} \in \mathbb{R}^{C\times H\times W}$ should be images (C, H, W denote the number of channels, height, and width). Thus CLoG is more challenging as its output space inherently possesses a significantly larger cardinality. Model architectures also diverge: classification-based CL typically requires a simple mechanism, such as linear mapping or MLP head [70], while CLoG necessitates sophisticated generative models as VAE [41], GAN [21], or score-based models [98], which are generally more complex to optimize [77, 97].

3 Benchmark Design

3.1 Task selection

The fundamental criterion for choosing tasks for our benchmark is to ensure they are *diverse* and *representative*. Diversity is crucial for evaluating the CLoG methods across various dimensions, making them relevant for different real-world applications. Selecting representative tasks is essential for efficiency since executing numerous redundant tasks can be resource-intensive and unproductive. We follow traditional CL literature [75, 4, 40, 55, 109] to split a publicly available dataset into a sequence of tasks for CLoG, including MNIST, CIFAR-10, ImageNet-1k, etc. The detailed description can be found in Appendix F.1. The specific partition of tasks is depicted in Appendix F.2.

3.2 Baseline setup

To establish the baselines, we adapt the three types of CL techniques ($i.e.$, regularization-based, replay-based, parameter-isolation-based) to CLoG. Note that there are several classification-based CL or CLoG methods that combine multiple techniques, but we did not include these baselines in our set as many of their basic components can be unified into the three types of methods (see Appendix B).

Adapted Baselines We adapted 12 baselines, encompassing parameter-isolation-based methods e.g. ensemble, C-LoRA, replay-based methods e.g. experience replay and generative replay, and regularization-based methods e.g. L2, EWC, SI, with Naive Continual Learning and Non-Continual Learning involved for comparison. For detailed introduction, please refer to Appendix D.

Generative models We apply CLoG methods on two representative generative models: Generative Adversarial Networks (GAN) [21] and Diffusion Models [28]. We introduce them in Appendix B.1.

3.3 Metrics design

In this paper, we provide unified metric choices for evaluating CLoG, including **Average Incremental Quality (AIQ)**, **Average Final Quality (AFQ)** and **Forgetting Rate (FR)** to reveal average quality, final performance, and the extent of forgetting respectively. Please refer to Appendix A.3 for definitions.

3.4 Training specifics

Given the diverse training specifics (e.g., image augmentation, network configurations, etc.), *the key idea is to fix the specifics irrelevant to CL performance (which might otherwise affect the generation performance) in implementing CLoG baselines*. Specifically, we fix the backbone for GAN and Diffusion Models to StyleGAN2 [35] and DDIM [96] for label-conditioned CLoG, DreamBooth [78] and Custom Diffusion [49] for concept-conditioned CLoG. We fix CL-irrelevant configurations such as DDIM steps, condition encoding, etc., with full details presented in Appendix G. This standardization improves evaluation efficiency by significantly reducing the hyper-parameter space, allowing us to focus on optimizing the hyper-parameters crucial for CL.

4 Experiments and Results

Experiment details are presented in Appendix G. Besides, we present the AFQ results as Tables 1 and 2, and postpone the AIQ, FR results in Appendices E.2 and E.4. We also conduct additional study on different configurations such as DDIM steps, replay buffer size, task numbers within the same dataset, different alignment score metrics in Appendix E.1, and visualize the generation results, compare efficiencies in Appendices E.3 and E.5. We draw some observations as follows.

NO single method works well across all settings. Specifically, the seemingly best-performing parameter-isolation-based methods ($i.e.$, ensemble, C-LoRA) work well on MNIST, CIFAR-10, ImageNet-1k, and Custom-Objects, but fail on Oxford-Flowers, CUB-Birds, and Stanford-Cars. This is because by isolating parameters, they inhibit knowledge transfer, which is significant for tasks that share similar features. Besides, these methods may also be memory-hungry (see Table 11) as they allocate parameters for each new task. Above all, the current methods are not satisfactory enough and our CLoG benchmark remains an open challenge.

NCL is comparable to regularization-based methods. Although NCL naively trains on the current task data without knowledge-preserving techniques, it exhibits similar performance compared to regularization-based methods. Some of them achieve even worse performance than NCL because the regularization makes them hard to learn new tasks, for example, MAS achieves poor AFQ on Oxford-Flowers and Stanford-Cars based on Diffusion Models, but it also has almost zero forgetting (shown in Tables 9 and 10). Note that we have grid searched the regularization weights from 0.001 to

Table 1: **AFQ results for label-conditioned CLoG benchmarks.** The best result in each column with the same architecture (StyleGAN2, DDIM) is highlighted in **red**, while the second best and third best are highlighted in **blue** and **yellow**, respectively. The quality metric is FID (*the lower value is better*). We average each AFQ value on 5 class orders and show the standard deviations as superscripts. We use dashlines to split different categories of baselines (Non-CL & NCL, replay-based, regularization-based, parameter-isolation-based).

	MNIST	Fashion-MNIST	CIFAR-10	CUB-Birds	Oxford-Flowers	Stanford-Cars	ImageNet-1k
- StyleGAN2							
Non-CL	**41.19**$^{\pm 3.44}$	**66.53**$^{\pm 1.46}$	**63.02**$^{\pm 5.18}$	**48.36**$^{\pm 2.16}$	**99.50**$^{\pm 10.4}$	**33.68**$^{\pm 2.95}$	NA
NCL	60.98$^{\pm 6.13}$	94.10$^{\pm 13.20}$	103.34$^{\pm 10.59}$	**112.57**$^{\pm 23.05}$	131.98$^{\pm 16.39}$	**68.20**$^{\pm 2.05}$	NA
ER	87.91$^{\pm 24.33}$	133.24$^{\pm 35.98}$	236.44$^{\pm 11.18}$	175.99$^{\pm 20.19}$	134.94$^{\pm 2.55}$	147.88$^{\pm 6.00}$	NA
GR	113.37$^{\pm 38.97}$	115.18$^{\pm 26.15}$	128.81$^{\pm 8.37}$	189.27$^{\pm 11.55}$	161.96$^{\pm 10.80}$	161.55$^{\pm 27.85}$	NA
KD	55.04$^{\pm 4.88}$	86.94$^{\pm 4.05}$	105.73$^{\pm 13.27}$	**108.68**$^{\pm 11.16}$	**120.66**$^{\pm 17.47}$	80.45$^{\pm 4.02}$	NA
L2	63.15$^{\pm 13.15}$	113.41$^{\pm 7.12}$	108.52$^{\pm 6.24}$	191.43$^{\pm 17.52}$	158.55$^{\pm 11.97}$	201.80$^{\pm 32.95}$	NA
EWC	54.73$^{\pm 4.52}$	87.20$^{\pm 11.12}$	95.33$^{\pm 19.04}$	156.06$^{\pm 13.38}$	131.62$^{\pm 6.00}$	100.22$^{\pm 10.81}$	NA
SI	93.12$^{\pm 17.59}$	102.29$^{\pm 7.57}$	100.13$^{\pm 4.85}$	204.44$^{\pm 14.61}$	170.53$^{\pm 15.07}$	211.72$^{\pm 46.52}$	NA
MAS	57.89$^{\pm 8.53}$	86.86$^{\pm 5.29}$	**85.22**$^{\pm 2.83}$	186.34$^{\pm 17.63}$	144.31$^{\pm 14.99}$	149.11$^{\pm 19.21}$	NA
A-GEM	**41.51**$^{\pm 16.42}$	**85.37**$^{\pm 18.99}$	98.42$^{\pm 11.47}$	116.37$^{\pm 13.30}$	**127.93**$^{\pm 12.64}$	**75.46**$^{\pm 5.54}$	NA
Ensemble	**8.64**$^{\pm 1.74}$	**27.76**$^{\pm 0.37}$	**45.26**$^{\pm 0.61}$	180.71$^{\pm 2.46}$	145.59$^{\pm 1.61}$	230.74$^{\pm 3.93}$	NA
- DDIM							
Non-CL	**5.59**$^{\pm 3.67}$	**9.02**$^{\pm 0.23}$	**30.19**$^{\pm 1.29}$	**49.30**$^{\pm 4.43}$	**48.81**$^{\pm 0.84}$	**27.97**$^{\pm 0.42}$	**47.27**
NCL	115.47$^{\pm 9.30}$	139.81$^{\pm 19.04}$	115.60$^{\pm 20.51}$	**98.89**$^{\pm 6.06}$	102.98$^{\pm 16.39}$	**42.81**$^{\pm 8.91}$	91.46
ER	28.64$^{\pm 2.74}$	52.47$^{\pm 2.85}$	132.07$^{\pm 8.92}$	**72.53**$^{\pm 6.39}$	**77.03**$^{\pm 2.62}$	81.26$^{\pm 6.44}$	101.15
GR	90.28$^{\pm 4.72}$	34.96$^{\pm 6.31}$	73.15$^{\pm 2.48}$	106.93$^{\pm 4.67}$	180.68$^{\pm 27.60}$	261.59$^{\pm 3.24}$	NA
KD	149.72$^{\pm 13.17}$	233.55$^{\pm 11.89}$	162.13$^{\pm 16.11}$	181.40$^{\pm 8.86}$	176.84$^{\pm 20.88}$	103.06$^{\pm 12.55}$	107.57
L2	184.05$^{\pm 27.14}$	190.04$^{\pm 5.81}$	174.78$^{\pm 16.90}$	182.79$^{\pm 13.50}$	191.90$^{\pm 33.87}$	254.21$^{\pm 28.00}$	119.22
EWC	158.22$^{\pm 22.70}$	139.52$^{\pm 20.07}$	127.09$^{\pm 19.23}$	101.12$^{\pm 14.87}$	99.34$^{\pm 8.27}$	**49.02**$^{\pm 2.72}$	99.93
SI	182.80$^{\pm 25.55}$	156.63$^{\pm 22.67}$	142.32$^{\pm 26.74}$	113.30$^{\pm 15.91}$	98.04$^{\pm 7.78}$	57.06$^{\pm 8.39}$	100.13
MAS	137.28$^{\pm 14.51}$	162.25$^{\pm 19.61}$	124.31$^{\pm 10.24}$	197.73$^{\pm 15.76}$	213.12$^{\pm 33.11}$	282.49$^{\pm 14.23}$	130.21
A-GEM	86.28$^{\pm 5.94}$	139.46$^{\pm 5.21}$	129.24$^{\pm 27.59}$	105.93$^{\pm 2.67}$	121.27$^{\pm 10.92}$	50.13$^{\pm 2.44}$	100.45
Ensemble	**4.12**$^{\pm 0.14}$	**10.42**$^{\pm 0.02}$	**36.52**$^{\pm 0.55}$	133.32$^{\pm 2.07}$	**70.16**$^{\pm 8.67}$	202.15$^{\pm 0.52}$	**56.97**
C-LoRA	**9.45**$^{\pm 0.38}$	**24.83**$^{\pm 5.23}$	**60.11**$^{\pm 6.15}$	148.81$^{\pm 1.22}$	117.11$^{\pm 7.15}$	250.90$^{\pm 35.87}$	**79.72**

Table 2: **AFQ results for concept-conditioned CLoG benchmark.** The best result in each row with the same base method (DreamBooth, Custom Diffusion) is highlighted in **red**, while the second best and third best are highlighted in **blue** and **yellow**, respectively. The quality metric is the average of text and image alignment scores (*the higher value is better*). The AFQ is also averaged over 5 orders.

Model	NCL	Non-CL	KD	L2	EWC	SI	MAS	Ensemble	C-LoRA
DreamBooth	78.54$^{\pm 0.53}$	**80.09**$^{\pm 0.1}$	78.73$^{\pm 0.16}$	79.00$^{\pm 0.38}$	**79.45**$^{\pm 0.41}$	78.54$^{\pm 0.39}$	78.00$^{\pm 0.46}$	**80.09**$^{\pm 0.25}$	**80.42**$^{\pm 0.25}$
Custom Diffusion	79.56$^{\pm 0.17}$	**80.30**$^{\pm 0.21}$	79.71$^{\pm 0.1}$	79.92$^{\pm 0.14}$	**80.10**$^{\pm 0.05}$	79.59 $^{\pm 0.27}$	78.79$^{\pm 0.18}$	**80.39**$^{\pm 0.24}$	-

10000 according to prior works [44, 3, 87]. The failure of regularization-based methods demonstrates the challenge of CLoG due to the use of sophisticated deep generative models.

Replay-based methods face imbalance issue. Surprisingly, replay-based methods don't always outperform non-exemplar methods on CLoG. We relate this phenomenon to the amplification of *CL imbalance* [22] and *data imbalance* [1, 33]: the limited replayed samples have been seen and trained many times which make them easier to learn than new task data, and lead to mode collapse [100]. The severe mode collapse can be observed in GAN-based ER (see Appendix E.3). CLoG tends to be more sensitive to these issues than classification-based CL possibly because the modeling of data distribution $p(\boldsymbol{x})$ is more difficult than classification distribution $p(\boldsymbol{y}|\boldsymbol{x})$ as discussed in Appendix A.2.

Comparison between GAN and Diffusion Models. Generally, GAN is harder to optimize than Diffusion Models on CLoG, with worse Non-CL and Ensemble performance as shown in Table 1. This suggests that Diffusion Models are more promising as the base architecture for CLoG.

Comparison between label-conditioned and concept-conditioned CLoG. With pre-trained backbone and fewer training samples, results on concept-conditioned CLoG exhibit less forgetting (NCL performs well), aligning with [87, 5]. However, the visualization results in Figures 10 and 11 are far from perfect, suggesting the necessity to improve concept-conditioned CLoG methods in future.

5 Conclusion

In this paper, we introduce a foundational framework for Continual Learning of Generative Models (CLoG). We explore the challenges CLoG presents compared to classification-based CL. We establish unified benchmarks, baselines, evaluation protocols and training guidelines for CLoG. Our findings underscore the necessity to develop advanced CLoG methods and advocate for a shift in focus from classification-based CL to CLoG, given the growing importance of generative foundation models.

References

[1] Hongjoon Ahn, Jihwan Kwak, Su Fang Lim, Hyeonsu Bang, Hyojun Kim, and Taesup Moon. Ss-il: Separated softmax for incremental learning. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 824–833, 2020. URL https://api.semanticscholar.org/CorpusID:227240555.

[2] Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3366–3375, 2017.

[3] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European conference on computer vision (ECCV)*, pages 139–154, 2018.

[4] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. *Advances in neural information processing systems*, 33:15920–15930, 2020.

[5] Xusheng Cao, Haori Lu, Linlan Huang, Xialei Liu, and Ming-Ming Cheng. Generative multi-modal models are good class-incremental learners. *arXiv preprint arXiv:2403.18383*, 2024.

[6] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European conference on computer vision (ECCV)*, pages 532–547, 2018.

[7] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*, 2018.

[8] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019.

[9] Patryk Chrabaszcz, Ilya Loshchilov, and Frank Hutter. A downsampled variant of imagenet as an alternative to the cifar datasets. *arXiv preprint arXiv:1707.08819*, 2017.

[10] Brian Christian. *The alignment problem: How can machines learn human values?* Atlantic Books, 2021.

[11] Yulai Cong, Miaoyun Zhao, Jianqiao Li, Sijia Wang, and Lawrence Carin. Gan memory with no forgetting. *ArXiv*, abs/2006.07543, 2020. URL https://api.semanticscholar.org/CorpusID:219686951.

[12] Yulai Cong, Miaoyun Zhao, Jianqiao Li, Sijia Wang, and Lawrence Carin. Gan memory with no forgetting. *Advances in Neural Information Processing Systems*, 33:16481–16494, 2020.

[13] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7): 3366–3385, 2021.

[14] Matthias Delange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Greg Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 1–1, 2021. ISSN 1939-3539. doi: 10.1109/tpami.2021.3057446. URL http://dx.doi.org/10.1109/TPAMI.2021.3057446.

[15] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. doi: 10.1109/MSP.2012.2211477.

[16] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[17] Khanh Doan, Quyen Tran, Tung Lam Tran, Tuan Nguyen, Dinh Phung, and Trung Le. Class-prototype conditional diffusion model with gradient projection for continual learning, 2024.

[18] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *Computer vision–ECCV 2020: 16th European conference, Glasgow, UK, August 23–28, 2020, proceedings, part XX 16*, pages 86–102. Springer, 2020.

[19] Chrisantha Fernando, Dylan Banarse, Charles Blundell, Yori Zwols, David Ha, Andrei A Rusu, Alexander Pritzel, and Daan Wierstra. Pathnet: Evolution channels gradient descent in super neural networks. *arXiv preprint arXiv:1701.08734*, 2017.

[20] Rui Gao and Weiwei Liu. DDGR: Continual learning with deep diffusion-based generative replay. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 10744–10763. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/gao23e.html.

[21] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[22] Yiduo Guo, Bing Liu, and Dongyan Zhao. Dealing with cross-task class discrimination in online continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11878–11887, 2023.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90.

[24] Hamed Hemati, Lorenzo Pellegrini, Xiaotian Duan, Zixuan Zhao, Fangfang Xia, Marc Masana, Benedikt Tscheschner, Eduardo Veas, Yuxiang Zheng, Shiji Zhao, Shao-Yuan Li, Sheng-Jun Huang, Vincenzo Lomonaco, and Gido M. van de Ven. Continual learning in the presence of repetition, 2024.

[25] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[26] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[27] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. URL https://openreview.net/forum?id=qw8AKxfYbI.

[28] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[29] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.

[30] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 831–839, 2019.

[31] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

[32] Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, Andrew M. Dai, Matthew D. Hoffman, Monica Dinculescu, and Douglas Eck. Music transformer, 2018.

[33] Rui Huang, Andrew Geng, and Yixuan Li. On the importance of gradients for detecting distributional shifts in the wild. *Advances in Neural Information Processing Systems*, 34: 677–689, 2021.

[34] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *Advances in neural information processing systems*, 33:12104–12114, 2020.

[35] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *Proc. CVPR*, 2020.

[36] Zixuan Ke and Bing Liu. Continual learning of natural language processing tasks: A survey, 2023.

[37] Zixuan Ke, Haowei Lin, Yijia Shao, Hu Xu, Lei Shu, and Bing Liu. Continual training of language models for few-shot learning. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2022.

[38] Zixuan Ke, Yijia Shao, Haowei Lin, Tatsuya Konishi, Gyuhak Kim, and Bing Liu. Continual pre-training of language models. In *The Eleventh International Conference on Learning Representations (ICLR-2023)*, 2023.

[39] Ronald Kemker and Christopher Kanan. Fearnet: Brain-inspired model for incremental learning. *arXiv preprint arXiv:1711.10563*, 2017.

[40] Gyuhak Kim, Bing Liu, and Zixuan Ke. A multi-head model for continual learning via out-of-distribution replay. In *Conference on Lifelong Learning Agents*, pages 548–563. PMLR, 2022.

[41] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[42] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022.

[43] Diederik P. Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models, 2023.

[44] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[45] Alexander Kolesnikov, Alexey Dosovitskiy, Dirk Weissenborn, Georg Heigold, Jakob Uszkoreit, Lucas Beyer, Matthias Minderer, Mostafa Dehghani, Neil Houlsby, Sylvain Gelly, Thomas Unterthiner, and Xiaohua Zhai. An image is worth 16x16 words: Transformers for image recognition at scale. 2021.

[46] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013.

[47] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[48] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[49] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1931–1941, 2023.

[50] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database, 2010.

[51] Jan Leike, David Krueger, Tom Everitt, Miljan Martic, Vishal Maini, and Shane Legg. Scalable agent alignment via reward modeling: a research direction. *arXiv preprint arXiv:1811.07871*, 2018.

[52] Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip Torr. Controllable text-to-image generation. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/1d72310edc006dadf2190caad5802983-Paper.pdf.

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

[54] Haowei Lin, Baizhou Huang, Haotian Ye, Qinyu Chen, Zihao Wang, Sujian Li, Jianzhu Ma, Xiaojun Wan, James Zou, and Yitao Liang. Selecting large language model to fine-tune via rectified scaling law. *arXiv preprint arXiv:2402.02314*, 2024.

[55] Haowei Lin, Yijia Shao, Weinan Qian, Ningxin Pan, Yiduo Guo, and Bing Liu. Class incremental learning via likelihood ratio based task prediction. *International Conference on Learning Representations (ICLR)*, 2024.

[56] Bing Liu, Sahisnu Mazumder, Eric Robertson, and Scott Grigsby. Ai autonomy: Self-initiated open-world continual learning and adaptation. *AI Magazine*, 44(2):185–199, 2023.

[57] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *ArXiv*, abs/2209.03003, 2022. URL https://api.semanticscholar.org/CorpusID:252111177.

[58] Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, and Q. Liu. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. *ArXiv*, abs/2309.06380, 2023. URL https://api.semanticscholar.org/CorpusID:261697392.

[59] Vincenzo Lomonaco, Lorenzo Pellegrini, Andrea Cossu, Antonio Carta, Gabriele Graffieti, Tyler L Hayes, Matthias De Lange, Marc Masana, Jary Pomponi, Gido M Van de Ven, et al. Avalanche: an end-to-end library for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3600–3610, 2021.

[60] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

[61] Andrea Madotto, Zhaojiang Lin, Zhenpeng Zhou, Seungwhan Moon, Paul Crook, Bing Liu, Zhou Yu, Eunjoon Cho, Pascale Fung, and Zhiguang Wang. Continual learning in task-oriented dialogue systems. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7452–7467, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.590. URL https://aclanthology.org/2021.emnlp-main.590.

[62] Marc Masana, Tinne Tuytelaars, and Joost Van de Weijer. Ternary feature masks: zero-forgetting for task-incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3570–3579, 2021.

[63] Sergi Masip, Pau Rodriguez, Tinne Tuytelaars, and Gido M. van de Ven. Continual learning of diffusion models with generative distillation, 2024.

[64] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

[65] Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. *ArXiv*, abs/2102.09672, 2021. URL https://api.semanticscholar.org/CorpusID:231979499.

[66] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pages 722–729. IEEE, 2008.

[67] Maxime Oquab, Timoth'ee Darcet, Théo Moutakanni, Huy Q. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russ Howes, Po-Yao (Bernie) Huang, Shang-Wen Li, Ishan Misra, Michael G. Rabbat, Vasu Sharma, Gabriel Synnaeve, Huijiao Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *ArXiv*, abs/2304.07193, 2023. URL https://api.semanticscholar.org/CorpusID:258170077.

[68] William S. Peebles and Saining Xie. Scalable diffusion models with transformers. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4172–4182, 2022. URL https://api.semanticscholar.org/CorpusID:254854389.

[69] Ivica Pesovski, Ricardo Santos, Roberto Henriques, and Vladimir Trajkovik. Generative ai for customizable learning experiences. *Sustainability*, 16(7):3034, 2024.

[70] Marius-Constantin Popescu, Valentina E Balas, Liliana Perescu-Popescu, and Nikos Mastorakis. Multilayer perceptron and neural networks. *WSEAS Transactions on Circuits and Systems*, 8(7):579–588, 2009.

[71] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[72] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[73] Omid Rafieian and Hema Yoganarasimhan. Ai and personalization. *Artificial Intelligence in Marketing*, pages 77–102, 2023.

[74] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation, 2021.

[75] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.

[76] Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. *arXiv preprint arXiv:1810.11910*, 2018.

[77] Kevin Roth, Aurelien Lucchi, Sebastian Nowozin, and Thomas Hofmann. Stabilizing training of generative adversarial networks through regularization. *Advances in neural information processing systems*, 30, 2017.

[78] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510, 2023.

[79] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

[80] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

[81] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding, 2022.

[82] Ari Seff, Alex Beatson, Daniel Suo, and Han Liu. Continual learning in generative adversarial nets, 2017.

[83] J. Seo, J. Kang, and G. Park. Lfs-gan: Lifelong few-shot image generation. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11322–11332, Los Alamitos, CA, USA, oct 2023. IEEE Computer Society. doi: 10.1109/ICCV51070.2023.01043. URL https://doi.ieeecomputersociety.org/10.1109/ICCV51070.2023.01043.

[84] Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *International Conference on Machine Learning*, pages 4548–4557. PMLR, 2018.

[85] Joan Serrà, Dídac Surís, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task, 2018.

[86] Mohammad Shahin, F Frank Chen, and Ali Hosseinzadeh. Harnessing customized ai to create voice of customer via gpt3. 5. *Advanced Engineering Informatics*, 61:102462, 2024.

[87] Yijia Shao, Yiduo Guo, Dongyan Zhao, and Bing Liu. Class-incremental learning based on label generation. *arXiv preprint arXiv:2306.12619*, 2023.

[88] Haizhou Shi, Zihao Xu, Hengyi Wang, Weiyi Qin, Wenyuan Wang, Yibin Wang, and Hao Wang. Continual learning of large language models: A comprehensive survey. *arXiv preprint arXiv:2404.16789*, 2024.

[89] Zifan Shi, Sida Peng, Yinghao Xu, Andreas Geiger, Yiyi Liao, and Yujun Shen. Deep generative models on 3d representations: A survey, 2023.

[90] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30, 2017.

[91] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. URL https://api.semanticscholar.org/CorpusID:14124313.

[92] James Seale Smith, Junjiao Tian, Shaunak Halbe, Yen-Chang Hsu, and Zsolt Kira. A closer look at rehearsal-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2409–2419, 2023.

[93] James Seale Smith, Yen-Chang Hsu, Lingyu Zhang, Ting Hua, Zsolt Kira, Yilin Shen, and Hongxia Jin. Continual diffusion: Continual customization of text-to-image diffusion with c-lora, 2024.

[94] Nate Soares and Benja Fallenstein. Aligning superintelligence with human interests: A technical research agenda. *Machine Intelligence Research Institute (MIRI) technical report*, 8, 2014.

[95] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015.

[96] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[97] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *Advances in neural information processing systems*, 33:12438–12448, 2020.

[98] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[99] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2021.

[100] Akash Srivastava, Lazar Valkov, Chris Russell, Michael U Gutmann, and Charles Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning. In *Neural Information Processing Systems*, 2017. URL https://api.semanticscholar.org/CorpusID:9302801.

[101] Gan Sun, Wenqi Liang, Jiahua Dong, Jun Li, Zhengming Ding, and Yang Cong. Create your world: Lifelong text-to-image diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[102] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015. doi: 10.1109/CVPR.2015.7298594.

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

[104] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio, 2016.

[105] Sakshi Varshney, Vinay Kumar Verma, K. SrijithP., Lawrence Carin, and Piyush Rai. Camgan: Continual adaptation modules for generative adversarial networks. In *Neural Information Processing Systems*, 2021. URL https://api.semanticscholar.org/CorpusID:236635024.

[106] Eli Verwimp, Shai Ben-David, Matthias Bethge, Andrea Cossu, Alexander Gepperth, Tyler L Hayes, Eyke Hüllermeier, Christopher Kanan, Dhireesha Kudithipudi, Christoph H Lampert, et al. Continual learning: Applications and the road forward. *arXiv preprint arXiv:2311.11908*, 2023.

[107] Jeffrey S Vitter. Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)*, 11(1):37–57, 1985.

[108] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[109] Fu-Yun Wang, Da-Wei Zhou, Liu Liu, Han-Jia Ye, Yatao Bian, De-Chuan Zhan, and Peilin Zhao. Beef: Bi-compatible class-incremental learning via energy-based expansion and fusion. In *The Eleventh International Conference on Learning Representations*, 2022.

[110] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application, 2023.

[111] Mitchell Wortsman, Vivek Ramanujan, Rosanne Liu, Aniruddha Kembhavi, Mohammad Rastegari, Jason Yosinski, and Ali Farhadi. Supermasks in superposition. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 15173–15184. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/ad1f8bb9b51f023cdc80cf94bb615aa9-Paper.pdf.

[112] Mitchell Wortsman, Vivek Ramanujan, Rosanne Liu, Aniruddha Kembhavi, Mohammad Rastegari, Jason Yosinski, and Ali Farhadi. Supermasks in superposition. *Advances in Neural Information Processing Systems*, 33:15173–15184, 2020.

[113] Chenshen Wu, Luis Herranz, Xialei Liu, Joost Van De Weijer, Bogdan Raducanu, et al. Memory replay gans: Learning to generate new categories without forgetting. *Advances in neural information processing systems*, 31, 2018.

[114] Chenshen Wu, Luis Herranz, Xialei Liu, Yaxing Wang, Joost van de Weijer, and Bogdan Raducanu. Memory replay gans: learning to generate images from new categories without forgetting. In *Neural Information Processing Systems*, 2018. URL https://api.semanticscholar.org/CorpusID:55701876.

[115] Tongtong Wu, Linhao Luo, Yuan-Fang Li, Shirui Pan, Thuy-Trang Vu, and Gholamreza Haffari. Continual learning for large language models: A survey. *arXiv preprint arXiv:2402.01364*, 2024.

[116] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017. URL http://arxiv.org/abs/1708.07747.

[117] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

[118] Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, and Jian Tang. Geodiff: A geometric diffusion model for molecular conformation generation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=PzcvxEMzvQC.

[119] Wenpeng Yin, Jia Li, and Caiming Xiong. Contintin: Continual learning from task instructions. *arXiv preprint arXiv:2203.08512*, 2022.

[120] Michał Zając, Kamil Deja, Anna Kuzina, Jakub M. Tomczak, Tomasz Trzciński, Florian Shkurti, and Piotr Miłoś. Exploring continual learning of diffusion models, 2023.

[121] Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. Lion: Latent point diffusion models for 3d shape generation, 2022.

[122] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, pages 3987–3995. PMLR, 2017.

[123] Mengyao Zhai, Lei Chen, Frederick Tung, Jiawei He, Megha Nawhal, and Greg Mori. Lifelong gan: Continual learning for conditional image generation. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2759–2768, 2019. URL https://api.semanticscholar.org/CorpusID:198229709.

[124] Mengyao Zhai, Lei Chen, and Greg Mori. Hyper-lifelonggan: Scalable lifelong learning for image conditioned generation. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2246–2255, 2021. URL https://api.semanticscholar.org/CorpusID:232351216.

[125] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3836–3847, October 2023.

[126] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023.

[127] Fei Zhu, Xu-Yao Zhang, Chuang Wang, Fei Yin, and Cheng-Lin Liu. Prototype augmentation and self-supervision for incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5871–5880, 2021.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [NA] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See our supplmentary materials.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [NA]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See our abstract and introduction.
   (b) Did you describe the limitations of your work? [Yes] See Appendix C.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Appendix H.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] We have read the ethics review guidelines and we ensure that our paper conforms to them.
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [NA]
   (b) Did you include complete proofs of all theoretical results? [NA]
3. If you ran experiments (e.g. for benchmarks)...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Our code is attached in the supplementary materials. The datasets are publicly available. The instructions to reproduce the main experimental results can be found in Appendix G.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Appendix G.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Our main tables include standard deviations.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Appendix G.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We cite the papers for the datasets. Our code largely depends on HuggingFace and PyTorch, and we also include the link to their website and publication in Appendix. The models were trained by ourselves.
   (b) Did you mention the license of the assets? [NA]
   (c) Did you include any new assets either in the supplemental material or as a URL? [NA]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [NA]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [NA]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [NA]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [NA]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [NA]

Appendix

Table of Contents

A Mathemetical Definitions

A.1 Formulation of general CL

CL learns a sequence of tasks $\mathcal{T}^{(1)}, \mathcal{T}^{(2)}, \cdots, \mathcal{T}^{(T)}$ incrementally. Each task $\mathcal{T}^{(t)}$ has an input space $\mathcal{X}^{(t)}$, an output space $\mathcal{Y}^{(t)}$, and a training set $\mathcal{D}^{(t)} = \{(\boldsymbol{x}_j^{(t)}, \boldsymbol{y}_j^{(t)})\}_{j=1}^{|\mathcal{D}^{(t)}|}$ drawn $i.i.d.$ from distribution $\mathcal{P}_{\mathcal{X}^{(t)}\mathcal{Y}^{(t)}}$. The goal of continual learning is to learn a function $f : \cup_{t=1}^{T}\mathcal{X}^{(t)} \to \cup_{t=1}^{T}\mathcal{Y}^{(t)}$ that can achieve good performance on each task $\mathcal{T}^{(t)}$.

A.2 Formulation of CLoG

CLoG learns a sequence of generation tasks $\mathcal{T}^{(1)}, \mathcal{T}^{(2)}, \cdots, \mathcal{T}^{(T)}$ incrementally. Each task $\mathcal{T}^{(t)}$ has an input space $\mathcal{X}^{(t)}$ (generation conditions) and output space $\mathcal{Y}$ (generation targets), and a training set $\mathcal{D}^{(t)} = \{(\boldsymbol{x}_j^{(t)}, \boldsymbol{y}_j^{(t)})\}_{j=1}^{|\mathcal{D}^{(t)}|}$ drawn $i.i.d.$ from distribution $\mathcal{P}_{\mathcal{X}^{(t)}\mathcal{Y}^{(t)}}$. The goal of CLoG is to learn a mapping $f : \cup_{t=1}^{T}\mathcal{X}^{(t)} \to \cup_{t=1}^{T}\mathcal{Y}^{(t)}$ that can achieve good performance on each task $\mathcal{T}^{(t)}$. The generation conditions can be text [52, 126], images [124, 126], or label indices [27], while the target can be various modalities such as images [74, 81], audio [32, 104], or 3D objects [89, 121]. As an initial step towards CLoG, we only focus on image generation conditioned on text or label indices in this paper.

A.3 CLoG Metrics

Suppose $m(f, \mathcal{T})$ is a metric to evaluate the generation quality of a generative model f on a task $\mathcal{T}$, when learning the task sequence $\mathcal{T}^{(1)}, \mathcal{T}^{(2)}, \cdots, \mathcal{T}^{(T)}$, we denote the model after learning $\mathcal{T}^{(i)}$ as $f^{(i)}$, CLoG metrics are as follows:

Average Incremental Quality (AIQ) [18, 30] We first define the average quality (AQ) when the model just learns the t-th task $\mathcal{T}^{(t)}$ as $AQ^{(t)} = \frac{1}{t}\sum_{i=1}^{t} m(f^{(t)}, \mathcal{T}^{(i)})$. Then the *average incremental quality* (AIQ) is defined to evaluate the historical performance as $AIQ = \frac{1}{T}\sum_{t=1}^{T} AQ^{(t)}$.

Average Final Quality (AFQ) [6, 60] Since AIQ evaluates the historical performance of the model during CL, while in downstream applications we may only care about the final performance of the model (i.e., the performance of $f^{(T)}$), *average final quality* (AFQ) is defined as $AFQ = AQ^{(T)}$.

Forgetting Rate (FR) [6, 60] Defined to measure the capability to preserve the learned "knowledge" during the CL process, the *forgetting rate* (FR) of task $\mathcal{T}^{(t)}$ is calculated by the performance difference between the current $m(f^{(T)}, \mathcal{T}^{(t)})$ and that when the model first learns this task $m(f^{(t)}, \mathcal{T}^{(t)})$:

$$FR = \begin{cases} \frac{1}{T-1}\sum_{t=1}^{T-1}\left(m(f^{(t)}, \mathcal{T}^{(t)}) - m(f^{(T)}, \mathcal{T}^{(t)})\right) & \text{(if larger } m \text{ is better)} \\ \frac{1}{T-1}\sum_{t=1}^{T-1}\left(m(f^{(T)}, \mathcal{T}^{(t)}) - m(f^{(t)}, \mathcal{T}^{(t)})\right) & \text{(if smaller } m \text{ is better)} \end{cases} \tag{1}$$

For label-conditioned CLoG, we choose Fréchet inception distance (FID) [25] as $m(f, \mathcal{T})$ (smaller m is better), which is commonly used to assess the generation quality of image generation models [96, 28]. [1] For concept-conditioned CLoG, we follow DreamBooth [78] to compute the average of the CLIP alignment score [72] between generated image and provided concept (image alignment score), and between generated image and text prompts (text alignment score).

[1] Some existing CLoG works used pre-trained classifiers to compute the accuracy of conditional generation, while we find it unsuitable for CLoG and do not adopt it. For example, a pre-trained classifier is not always available for some datasets, and the classifiers often assign wrong prediction for OOD generated images.

B Related Work

B.1 Generative Models

Generative Adversarial Networks. (GAN) GAN [21] consists of two interacting networks: a generator and a discriminator. The generator G_{θ_g}, fed with random noise $\boldsymbol{z} \sim p_{\boldsymbol{z}}$, is designed to produce images that mimic the true samples from a data distribution p_{data} closely enough to deceive the discriminator. Conversely, the discriminator D_{θ_d} attempts to discern between the authentic data points $\boldsymbol{x}$ and the synthetic images $G_{\theta_g}(\boldsymbol{z})$ produced by the generator. The training objective for this adversarial process is formulated as follows [21] :

$$\min_{\theta_g} \max_{\theta_d} [\mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}} \log D_{\theta_d}(\boldsymbol{x}) + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}} \log(1 - D_{\theta_d}(G_{\theta_g}(\boldsymbol{z}))] \tag{2}$$

Diffusion Models. Diffusion probabilistic models [95, 29, 99] generate samples by an iterative denoising process. It defines a gradual process of adding noises, which is called the diffusion process or forward process and generate images by removing the noises step-by-step, which is referred to as the reverse process. In forward process, gaussian noises are added to $\boldsymbol{x}_t$, beginning from data $\boldsymbol{x}_0$ [29]:

$$q(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t) = \mathcal{N}(\sqrt{1-\beta_t}\boldsymbol{x}_t, \beta_t \boldsymbol{I}),\ 0 \leq t < T \tag{3}$$

where β_t stands for the variance schedule of noise added at time t. With diffusion steps $T \to \infty$, $\boldsymbol{x}_T$ virtually becomes a random noise sampled from $\mathcal{N}(0, \boldsymbol{I})$. In contrast, the reverse process starts from Gaussian noise $\boldsymbol{x}_T \sim \mathcal{N}(0, \boldsymbol{I})$, during which diffusion network predicts noises x_t [29]:

$$p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t), \boldsymbol{\Sigma}_\theta(\boldsymbol{x}_t, t)),\ 0 < t \leq T \tag{4}$$

where $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)$ is parameterized by a neural network and can be converted to $\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t)$ with reparameterization trick [42] and $\boldsymbol{\Sigma}_\theta(\boldsymbol{x}_t, t) = \sigma_t \boldsymbol{I}$ under the isotropic Gaussian assumption of noises [29]. To learn the reverse process, diffusion models are trained by optimizing the variational lower bound [43] of probability $p_\theta(\boldsymbol{x}_{0:T}) = p_\theta(\boldsymbol{x}_T)\Pi_{t=1}^T p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$. One commonly used and simple loss equivalent is written as [29]:

$$L(\theta) = \mathbb{E}_{t, \boldsymbol{x}_0, \boldsymbol{\epsilon}} \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t\|^2 \tag{5}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$ and $\bar{\alpha}_t = \Pi_{s=1}^t (1 - \beta_s)$. The sampling process starts from $x_T \sim \mathcal{N}(0, \boldsymbol{I})$, iterates $t = T, \ldots, 1$ and denoises according to formula [29]:

$$\boldsymbol{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}(\boldsymbol{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)) + \sigma_t \boldsymbol{z} \tag{6}$$

where $\boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{I})$ if $t > 1$ else $\boldsymbol{z} = 0$.

B.2 Continual learning of generative models

The earliest related work of CLoG proposes *generative replay* (GR) [90] for classification-based CL. GR utilizes a continually trained generator to synthesize data from previous tasks, thereby preventing forgetting when training a continual learning classifier. The primary goal of GR methods, however, is to enhance classification performance rather than the quality of the generated data. In past years, pioneering studies in CLoG have emerged [123, 11, 101], but they often lacked a unified evaluation protocol and tested on distinctive distinct tasks, making the comparison difficult. Additionally, the diversity in model architectures, data processing techniques, training pipelines, and evaluation metrics complicates fair comparisons between different CLoG methods. This contrasts with classification-based CL, where these choices are relatively standardized [59, 13, 110]. Relevant works on CLoG are listed below, classified based on model architecture.

Continual Learning of GAN Since GAN [21] was proposed, several works have brought out continual learning settings for GANs and incorporated different methods to overcome catastrophic forgetting. Seff et al. [82] first integrated EWC [44] into continual learning for GANs. Zhai et al. [123] adopted knowledge distillation to distill knowledge from the previous model to the current model to mitigate catastrophic forgetting. Wu et al. [114] implemented deep generative replay [90], e.g., joint retraining and replay alignment, on a conditional GAN to avoid potential accumulate classification errors. Cong et al. [11] prevents forgetting by adding additional parameters to learn

newly encountered tasks. Following FiLM and mAdaFM, the authors tailored these modulators for fully connected and convolutional networks to better perceive new information. Following this strategy, CAM-GAN [105] proposed a combination of group-wise and point-wise convolutional filters to learn novel tasks while further improved CL performance by leveraging task-similarity estimation with Fisher information matrix. Hyper-LifelongGAN [124] decomposed convolutional filters into dynamic task-specific filters generated by a filter generator and task-agnostic fixed weight components. Knowledge distillation techniques were adopted to further reduce forgetting issues. LFS-GAN [83] introduced newly proposed modulators termed LeFT, a rank-constrained weight factorization method while additional mode-seeking losses are adopted to prevent mode collapsing and enhance generation diversity.

Continual Learning of Diffusion Models Diffusion models [95, 29, 99], a model that have been proved to be capable of high-quality image generations recent years, have also been experimented in CL. Gao and Liu [20] trained a classifier and diffusion model bi-directional way, where the classifier is used to guide the conditional diffusion sampling. Doan et al. [17] added trainable class prototypes to represent previous classes and utilize gradient projection in diffusion process to alleviate forgetting. In addition to classifier-guided methods, Zając et al. [120] tested several common forgetting-prevent methods on MNIST [15] and Fashion-MNIST [117] including experience replay, generative replay and L2 regularization, scratched the surface of continual diffusion model learning. Masip et al. [63] introduced generative distillation process, aligning predicted noises with previous task models at each step of the reverse sampling trajectory. Smith et al. [93] proposed C-LoRA that trained distinct self-regulated LoRA [31] blocks in cross attention layers respectively for different tasks. We extend the C-LoRA method in our benchmarks to more general CLoG settings.

C Discussions and Limitations

C.1 Discussions

There has long been a period that CL research focusing on addressing forgetting for classification tasks, and although many advancements have been achieved in the past years, CL methods are rarely applied in real world applications. With the emergence of various "all-in-one" foundation models, the relevance of focusing mainly on classification-based CL is increasingly questionable. Current trends suggest that generative-based foundation models are poised to become the next generation of AI products, integral to everyday life. In this context, CLoG becomes crucial, addressing how these models of diverse architectures, complex learning objectives, and open-ended domains can continuously learn newly emerged knowledge [38, 106, 56], cater to personalized needs [73, 69, 86], and possibly enhance human-AI alignment in an evolving world [94, 10, 51]. Our benchmark results reveal disappointing performance with traditional CL methods, highlighting a pressing need for refined CLoG strategies for future applications.

C.2 Limitations and outlook

This study primarily presents initial benchmarks and baselines for CLoG, with a focus on traditional representative methods. Future work will include expanding these benchmarks across a wider range of image generation tasks, incorporating various generative conditions [123, 125], and extending to additional modalities such as molecules [118]. Although we only include baselines from classification-based CL, it is interesting to design methods specifically for generative models by applying techniques such as classifier-guidance [16], and include more generative models [57, 68, 58] other than GAN and Diffusion Models. Our current analysis is based solely on existing datasets; hence, we plan to enhance the scope of concept-conditioned CLoG benchmarks by acquiring and incorporating more diverse datasets and domains. Furthermore, while this paper primarily conducts an empirical investigation, advancing the theoretical framework of CLoG will be crucial for its development and understanding.

C.3 Remarks on text generation in CL.

CL of text generation tasks [88, 115] has garnered increasing interest in the so-called "post-LLM era" [54]. Including CL of text generation tasks in CLoG is logical, given their generative nature. However, text generation typically operates under the framework of "next token prediction" [71], which substantially differs from the typical probabilistic generative models. Specifically, text generation models the conditional data distribution, $\mathbf{P}(\boldsymbol{y}|\boldsymbol{x})$, through autoregressive generation of the form $\prod_{j=1}^{|\boldsymbol{y}|}\mathbf{P}(\boldsymbol{y}_j|\boldsymbol{y}_1,\boldsymbol{y}_2,\cdots,\boldsymbol{y}_{j-1},\boldsymbol{x})$, with $\boldsymbol{y}_0$ representing the "start of sentence" token and $\boldsymbol{y}_j$ the jth token of $\boldsymbol{y}$. This approach simplifies the modeling of complex data distributions into predicting a sequence of categorical distributions over the vocabulary space. Existing studies have shown that text generation tasks tend to exhibit less forgetting when integrated into CL frameworks, indicating a potentially smoother adaptation to continual learning [119, 87, 5]. While we advocate for the inclusion of text generation in CLoG due to its alignment with CLoG's formulation, the primary focus of this paper remains on general probabilistic generative modeling due to its broader applications and the insufficient attention it has received in research.

D Adapted Baselines

(1) **Naive Continual Learning (NCL)** means continually training the same model without any CL techniques to deal with forgetting, which is the simplest baseline in CL.

(2) **Non-Continual Learning (Non-CL)** means pooling the data from all tasks together and training only one model for all tasks. This is not under a CL setting but its performance can be viewed as an upper bound for CL baselines.

(3) **Ensemble** trains a separate model for each task. This baseline is forgetting-free, but the memory consumption is huge when more tasks arrive, and there is no knowledge transfer between different tasks.

(4) **Experience Replay (ER) [60]** directly combines replay samples and current task samples in training batches to train the model. The replay data is saved by reservoir sampling [8, 76].

(5) **Generative Replay (GR) [90]** replaces the replay samples used in ER with generative replay samples. When training a new task in CLoG, the model is copied and the replay samples are generated via the copied model.

(6) **Knowledge Distillation (KD) [26]** is a regularization-based method in CL. The model is copied as a fixed teacher model before learning the new task. An ℓ_2 auxiliary loss between the new and old model outputs is added to the NCL objective.

(7) **L2 [92]** is also a regularization-based method and copies the model before learning a new task. An ℓ_2 distance between the current and copied network parameters is added as an auxiliary loss.

(8) **Elastic Weight Consolidation (EWC) [44]** is also a regularization technique that reweights the ℓ_2 loss for different parameters. The weights are based on the degree of overlap between the two tasks' Fisher matrices.

(9) **Synapse Intelligence (SI) [122]** is a regularization method that is similar to EWC, while the parameter weights are computed by measuring the parameter updating trajectory during training.

(10) **Memory Aware Synapses (MAS) [3]** is also a regularization-based method. It measures the importance of parameters by the magnitude of the gradient and penalizes changes to parameters that are essential to previous tasks.

(11) **Averaged Gradient Episodic Memory (A-GEM) [7]** is a regularization-based method that exploits replay data. It prevents the loss increasing on replay samples by gradient projection.

(12) **C-LoRA** [93] is a parameter-isolation-based CL method that was first designed for concept-conditional CLoG. It overcomes forgetting by learning task-specific LoRA [31] upon a pre-trained backbone. We adapt it to from-scratch-training by fully training the backbone on the first task and adopting LoRA tuning in the subsequent tasks.

E Comprehensive Results

E.1 Ablation study

In this section, we conduct a series of ablation studies on the configurations that we fixed in our benchmark experiments.

Different DDIM steps. Since a larger DDIM step, though may improve the generation quality, will result in significant inference overhead and is irrelevant to CL capability, we fix it to a small value. In our experiments, we set DDIM steps as 50 for all the DDIM-based baselines. We evaluate the CLoG baselines with a larger number of DDIM steps on CIFAR-10, and the results are in Table 3. It shows that the DDIM step as 50 can already faithfully reflect the performance of CLoG baselines without $2\times$ or $4\times$ computations.

Table 3: Performance of Different DDIM Steps on CIFAR-10

DDIM Step		50	100	200
NCL	AFQ	$115.60^{\pm 20.51}$	$112.41^{\pm 16.62}$	$105.34^{\pm 13.65}$
	AIQ	$108.19^{\pm 15.02}$	$96.82^{\pm 8.82}$	$95.04^{\pm 6.47}$
	FR	$107.04^{\pm 27.11}$	$104.75^{\pm 21.75}$	$95.94^{\pm 18.22}$
ER	AFQ	$132.07^{\pm 8.92}$	$132.94^{\pm 2.91}$	$131.77^{\pm 2.35}$
	AIQ	$138.22^{\pm 6.12}$	$131.59^{\pm 3.84}$	$136.80^{\pm 3.82}$
	FR	$93.81^{\pm 13.71}$	$95.53^{\pm 5.78}$	$90.00^{\pm 5.49}$
EWC	AFQ	$127.09^{\pm 19.23}$	$126.23^{\pm 10.22}$	$129.14^{\pm 7.79}$
	AIQ	$113.06^{\pm 8.89}$	$104.48^{\pm 4.22}$	$109.01^{\pm 2.16}$
	FR	$119.74^{\pm 25.80}$	$120.61^{\pm 13.86}$	$118.46^{\pm 3.53}$
Ensemble	AFQ	$36.52^{\pm 0.55}$	$35.91^{\pm 0.48}$	$37.73^{\pm 0.95}$
	AIQ	$36.57^{\pm 1.57}$	$34.97^{\pm 1.70}$	$40.70^{\pm 3.52}$
	FR	0	0	0
C-LoRA	AFQ	$60.11^{\pm 6.15}$	$61.21^{\pm 5.89}$	$63.94^{\pm 5.30}$
	AIQ	$173.43^{\pm 45.28}$	$56.67^{\pm 6.21}$	$58.54^{\pm 5.33}$
	FR	0	0	0

Different memory sizes. In our benchmarks, we follow the existing CL works to set replay buffer sizes as 200 for small-scale CL datasets, and 5000 for large-scale ImageNet-1k. Table 4 shows the results of varying replay buffer sizes on CIFAR-10. It suggests that the performance of ER method is not sensitive to the memory size ranging from 20 to 400.

Table 4: Performance of Different Memory Size of Experience Replay (ER)

Memory Size		20	50	100	200	400
ER	AFQ	133.65	136.11	138.15	135.47	133.83
	AIQ	109.35	133.10	110.96	113.66	137.00
	FR	96.58	100.11	102.45	97.98	95.99

Different class separation. As a dataset can be split into different numbers of tasks, here we experiment on CIFAR-10 with 2, 5, 10 tasks. The results are shown in Table 5.

Table 5: Performance of Different Class Seperation on CIFAR-10

Task number		10	5	2
NCL	AFQ	153.32	115.60	83.50
	AIQ	159.97	118.19	61.09
	FR	125.27	107.04	96.84
ER	AFQ	227.92	132.07	119.95
	AIQ	251.29	138.22	86.94
	FR	189.41	93.81	126.09
EWC	AFQ	174.98	127.09	96.50
	AIQ	167.55	113.06	67.73
	FR	149.96	119.74	123.77
Ensemble	AFQ	54.08	36.52	38.12
	AIQ	57.87	36.57	38.39
	FR	0	0	0
C-LoRA	AFQ	93.47	60.11	44.68
	AIQ	87.04	173.43	41.91
	FR	0.01	0.31	0.21

Different alignment scores. We also evaluate the concept-conditioned CLoG on another alignment score computed by DINO [67]. The results are shown in Table 6.

Table 6: **AFQ results for concept-conditioned CLoG benchmark with different alignment scores.** The best result in each row with the same base method (DreamBooth, Custom Diffusion) is highlighted in **red**, while the second best and third best are highlighted in **blue** and **yellow**, respectively. The quality metric is the average of text and image alignment scores (*the higher value is better*). The AFQ is also averaged over 5 orders.

Metric	Model	NCL	Non-CL	KD	L2	EWC	SI	MAS	Ensemble	C-LoRA
CLIP Avg	DreamBooth	$78.54^{\pm 0.53}$	**80.09**$^{\pm 0.1}$	$78.73^{\pm 0.16}$	$79.00^{\pm 0.38}$	**79.45**$^{\pm 0.41}$	$78.54^{\pm 0.39}$	$78.00^{\pm 0.46}$	**80.09**$^{\pm 0.25}$	**80.42**$^{\pm 0.25}$
	Custom Diffusion	$79.56^{\pm 0.17}$	**80.30**$^{\pm 0.21}$	$79.71^{\pm 0.1}$	$79.92^{\pm 0.14}$	**80.10**$^{\pm 0.05}$	$79.59^{\pm 0.27}$	$78.79^{\pm 0.18}$	**80.39**$^{\pm 0.24}$	-
DINO Avg	DreamBooth	$70.81^{\pm 0.72}$	**71.76**$^{\pm 0.2}$	$70.55^{\pm 0.71}$	$70.90^{\pm 0.76}$	$70.57^{\pm 0.58}$	$69.69^{\pm 0.81}$	$64.47^{\pm 2.3}$	**72.81**$^{\pm 0.68}$	**73.677**$^{\pm 0.38}$
	Custom Diffusion	$67.43^{\pm 0.71}$	**69.96**$^{\pm 0.4}$	$67.64^{\pm 0.28}$	$67.91^{\pm 0.51}$	$69.21^{\pm 0.47}$	**69.60**$^{\pm 1.2}$	$64.93^{\pm 1.6}$	**70.95**$^{\pm 1.30}$	-

E.2 AIQ and FR Results

In this section, we present the AIQ and FR results on all benchmarks.

We can observe the average forgetting rate becomes negative in some cases on the CUB-Birds, Oxford-Flowers and Stanford-Cars datasets. This phenomenon suggests the existence of positive knowledge transfer among these datasets. Note that the FR of the ensemble and C-LoRA method is set to zero since we train a separate model for each task.

Table 7: AIQ results for label-conditioned benchmarks.

	MNIST	**Fashion-MNIST**	**CIFAR-10**	**CUB-Birds**	**Oxford-Flowers**	**Stanford-Cars**	**ImageNet**
- GAN							
Non-CL	$38.33^{\pm 1.89}$	$49.87^{\pm 3.94}$	$57.13^{\pm 3.63}$	$86.62^{\pm 7.27}$	$118.11^{\pm 2.21}$	$67.97^{\pm 1.13}$	NA
NCL	$45.50^{\pm 4.41}$	$73.49^{\pm 5.27}$	$80.23^{\pm 5.46}$	$125.13^{\pm 10.88}$	$127.18^{\pm 13.54}$	$97.77^{\pm 10.44}$	NA
ER	$37.23^{\pm 6.24}$	$61.62^{\pm 10.59}$	$173.08^{\pm 3.13}$	$180.04^{\pm 6.14}$	$151.53^{\pm 12.58}$	$159.86^{\pm 4.94}$	NA
GR	$54.19^{\pm 12.55}$	$36.13^{\pm 12.30}$	$71.66^{\pm 0.67}$	$180.82^{\pm 2.02}$	$158.24^{\pm 2.65}$	$190.07^{\pm 19.21}$	NA
KD	$39.31^{\pm 0.34}$	$69.12^{\pm 3.48}$	$80.98^{\pm 4.71}$	$135.21^{\pm 8.06}$	$131.76^{\pm 13.04}$	$102.79^{\pm 9.81}$	NA
L2	$44.01^{\pm 9.44}$	$81.90^{\pm 10.70}$	$82.65^{\pm 4.54}$	$182.36^{\pm 11.50}$	$159.24^{\pm 9.57}$	$202.29^{\pm 37.37}$	NA
EWC	$38.94^{\pm 2.48}$	$58.17^{\pm 3.96}$	$67.39^{\pm 9.77}$	$155.24^{\pm 6.27}$	$134.34^{\pm 3.98}$	$150.31^{\pm 22.91}$	NA
SI	$75.24^{\pm 28.81}$	$77.05^{\pm 9.37}$	$78.55^{\pm 3.69}$	$189.11^{\pm 13.55}$	$164.99^{\pm 8.35}$	$198.58^{\pm 37.99}$	NA
MAS	$48.93^{\pm 2.05}$	$61.70^{\pm 2.27}$	$70.24^{\pm 4.03}$	$179.28^{\pm 9.21}$	$143.04^{\pm 9.29}$	$169.43^{\pm 12.43}$	NA
A-GEM	$31.99^{\pm 7.05}$	$60.94^{\pm 5.47}$	$78.23^{\pm 3.48}$	$125.17^{\pm 5.61}$	$131.95^{\pm 7.06}$	$101.79^{\pm 4.75}$	NA
Ensemble	$10.85^{\pm 3.26}$	$27.30^{\pm 1.04}$	$44.35^{\pm 2.04}$	$177.25^{\pm 3.59}$	$148.64^{\pm 6.28}$	$232.97^{\pm 6.27}$	NA
- Diffusion Model							
Non-CL	$4.47^{\pm 1.30}$	$9.13^{\pm 0.32}$	$31.08^{\pm 2.32}$	$65.24^{\pm 1.60}$	$53.76^{\pm 2.55}$	$33.56^{\pm 0.39}$	46.08
NCL	$105.79^{\pm 4.02}$	$128.78^{\pm 13.05}$	$108.19^{\pm 15.02}$	$104.31^{\pm 2.03}$	$101.15^{\pm 9.07}$	$54.47^{\pm 4.27}$	92.08
ER	$19.76^{\pm 1.02}$	$36.91^{\pm 2.13}$	$138.22^{\pm 6.12}$	$79.46^{\pm 4.26}$	$77.44^{\pm 2.09}$	$77.75^{\pm 3.06}$	97.16
GR	$61.22^{\pm 1.27}$	$27.28^{\pm 4.84}$	$60.58^{\pm 1.08}$	$194.27^{\pm 8.32}$	$98.31^{\pm 3.77}$	$244.96^{\pm 7.85}$	NA
KD	$150.13^{\pm 3.35}$	$237.93^{\pm 10.01}$	$185.38^{\pm 2.31}$	$178.04^{\pm 1.76}$	$169.74^{\pm 10.38}$	$113.08^{\pm 7.12}$	110.09
L2	$158.51^{\pm 14.52}$	$175.01^{\pm 13.47}$	$164.06^{\pm 6.92}$	$175.35^{\pm 10.14}$	$188.30^{\pm 21.85}$	$267.73^{\pm 25.74}$	112.21
EWC	$137.11^{\pm 17.60}$	$131.18^{\pm 5.44}$	$113.06^{\pm 8.89}$	$104.53^{\pm 8.63}$	$101.60^{\pm 4.54}$	$59.11^{\pm 3.80}$	98.19
SI	$149.27^{\pm 12.98}$	$130.66^{\pm 12.96}$	$114.16^{\pm 13.86}$	$115.62^{\pm 6.39}$	$105.92^{\pm 3.90}$	$64.62^{\pm 1.77}$	102.01
MAS	$112.17^{\pm 10.32}$	$135.52^{\pm 13.56}$	$109.80^{\pm 10.04}$	$189.30^{\pm 13.95}$	$191.96^{\pm 32.86}$	$227.41^{\pm 16.70}$	113.23
A-GEM	$106.25^{\pm 6.83}$	$135.17^{\pm 10.41}$	$115.26^{\pm 10.26}$	$108.94^{\pm 2.31}$	$100.64^{\pm 5.55}$	$56.85^{\pm 3.03}$	62.99
Ensemble	$4.13^{\pm 0.21}$	$10.29^{\pm 0.22}$	$36.57^{\pm 1.57}$	$131.94^{\pm 3.45}$	$72.84^{\pm 14.34}$	$201.71^{\pm 2.99}$	56.86
C-LoRA	$140.51^{\pm 4.74}$	$229.63^{\pm 5.09}$	$173.43^{\pm 45.28}$	$186.01^{\pm 23.20}$	$288.38^{\pm 7.12}$	$269.84^{\pm 29.35}$	73.16

Table 8: AIQ results for the concept-conditioned CLoG benchmarks.

	Model	**NCL**	**Non-CL**	**KD**	**L2**	**EWC**	**SI**	**MAS**	**Ensemble**	**C-LoRA**
CLIP Avg	DreamBooth	78.40	79.07	78.57	78.75	79.43	78.73	77.87	0	0
	Custom Diffusion	79.86	79.40	79.64	79.86	79.77	-	79.39	0	-
DINO Avg	DreamBooth	73.66	73.71	72.06	73.15	73.02	72.32	69.24	0	0
	Custom Diffusion	71.89	72.87	71.55	71.94	71.88	-	70.46	0	-

Table 9: FR results for label-conditioned CLoG benchmarks.

	MNIST	**Fashion-MNIST**	**CIFAR-10**	**CUB-Birds**	**Oxford-Flowers**	**Stanford-Cars**	**ImageNet**
- GAN							
Non-CL	$5.13^{\pm 4.30}$	$10.42^{\pm 7.03}$	$5.91^{\pm 1.29}$	$-44.36^{\pm 8.99}$	$-18.58^{\pm 9.94}$	$-38.33^{\pm 4.21}$	NA
NCL	$68.91^{\pm 7.78}$	$94.67^{\pm 17.44}$	$74.34^{\pm 13.26}$	$-8.44^{\pm 20.19}$	$19.52^{\pm 10.22}$	$-32.05^{\pm 7.63}$	NA
ER	$94.27^{\pm 31.22}$	$128.76^{\pm 3.26}$	$209.16^{\pm 16.74}$	$15.72^{\pm 26.08}$	$13.42^{\pm 11.76}$	$56.72^{\pm 3.45}$	NA
GR	$120.05^{\pm 45.12}$	$110.12^{\pm 24.27}$	$112.96^{\pm 10.23}$	$101.54^{\pm 15.07}$	$56.11^{\pm 13.07}$	$85.51^{\pm 25.89}$	NA
KD	$60.64^{\pm 5.64}$	$87.50^{\pm 4.45}$	$74.82^{\pm 16.83}$	$-22.49^{\pm 13.06}$	$-3.91^{\pm 15.64}$	$-24.21^{\pm 6.45}$	NA
L2	$49.38^{\pm 11.48}$	$66.41^{\pm 10.29}$	$40.87^{\pm 12.91}$	$12.41^{\pm 7.87}$	$0.57^{\pm 2.88}$	$5.21^{\pm 5.22}$	NA
EWC	$58.96^{\pm 6.14}$	$80.06^{\pm 13.34}$	$50.91^{\pm 32.37}$	$7.16^{\pm 12.56}$	$2.30^{\pm 4.31}$	$-47.16^{\pm 11.31}$	NA
SI	$4.93^{\pm 3.61}$	$0.08^{\pm 0.64}$	$2.46^{\pm 3.16}$	$14.99^{\pm 12.91}$	$1.68^{\pm 5.34}$	$17.05^{\pm 12.49}$	NA
MAS	$52.57^{\pm 15.81}$	$72.31^{\pm 7.88}$	$32.45^{\pm 8.39}$	$9.18^{\pm 10.15}$	$4.18^{\pm 7.28}$	$-14.71^{\pm 7.45}$	NA
A-GEM	$44.45^{\pm 20.32}$	$84.84^{\pm 23.35}$	$66.91^{\pm 14.49}$	$0.48^{\pm 11.85}$	$0.45^{\pm 12.30}$	$-30.09^{\pm 7.05}$	NA
Ensemble	0	0	0	0	0	0	0
- Diffusion Model							
Non-CL	$1.82^{\pm 3.74}$	$-0.72^{\pm 1.34}$	$-2.46^{\pm 0.99}$	$-30.08^{\pm 4.29}$	$-12.73^{\pm 6.43}$	$-20.21^{\pm 2.31}$	0.19
NCL	$139.69^{\pm 11.51}$	$163.36^{\pm 23.95}$	$107.04^{\pm 27.11}$	$6.99^{\pm 8.89}$	$36.76^{\pm 13.03}$	$1.76^{\pm 12.15}$	62.89
ER	$14.70^{\pm 30.99}$	$54.18^{\pm 3.92}$	$93.81^{\pm 13.71}$	$-26.90^{\pm 4.34}$	$8.34^{\pm 9.06}$	$53.60^{\pm 7.25}$	57.61
GR	$100.28^{\pm 6.87}$	$32.37^{\pm 5.54}$	$54.74^{\pm 3.98}$	$75.37^{\pm 24.00}$	$51.59^{\pm 12.10}$	$178.51^{\pm 7.15}$	NA
KD	$70.52^{\pm 11.67}$	$58.28^{\pm 19.35}$	$17.22^{\pm 24.80}$	$14.89^{\pm 8.13}$	$35.30^{\pm 32.65}$	$-17.80^{\pm 18.44}$	25.68
L2	$202.09^{\pm 35.42}$	$193.35^{\pm 15.23}$	$132.67^{\pm 24.35}$	$25.69^{\pm 6.33}$	$26.03^{\pm 34.77}$	$1.25^{\pm 20.09}$	21.49
EWC	$192.15^{\pm 29.17}$	$161.97^{\pm 25.18}$	$119.74^{\pm 25.80}$	$24.14^{\pm 15.32}$	$45.24^{\pm 5.50}$	$0.05^{\pm 3.74}$	69.17
SI	$212.38^{\pm 33.81}$	$179.85^{\pm 28.77}$	$122.83^{\pm 35.85}$	$20.66^{\pm 17.76}$	$35.40^{\pm 11.75}$	$7.57^{\pm 8.37}$	62.61
MAS	$-0.16^{\pm 0.28}$	$0.75^{\pm 0.83}$	$0.96^{\pm 0.94}$	$0.72^{\pm 0.99}$	$-0.46^{\pm 0.34}$	$0.35^{\pm 0.26}$	0.27
A-GEM	$138.81^{\pm 11.12}$	$163.41^{\pm 5.92}$	$123.51^{\pm 34.69}$	$11.80^{\pm 5.91}$	$57.86^{\pm 14.35}$	$-0.61^{\pm 1.22}$	62.99
Ensemble	0	0	0	0	0	0	0
C-LoRA	0	0	0	0	0	0	0

Table 10: FR results for concept-conditioned CLoG benchmark

	Model	**NCL**	**Non-CL**	**KD**	**L2**	**EWC**	**SI**	**MAS**	**Ensemble**	**C-LoRA**
CLIP Avg	DreamBooth	0.6827	-0.7118	1.4377	0.4194	0.9916	1.5471	0.2172	0	0
	Custom Diffusion	0.3503	0.0846	0.256	0.0588	-0.1222	0.276	0.0049	0	0
DINO Avg	DreamBooth	2.5382	0.6054	3.3108	2.1836	3.3122	3.2506	0.3461	0	-
	Custom Diffusion	2.5568	1.5127	2.3811	1.5137	0.5065	1.1772	-0.0017	0	-

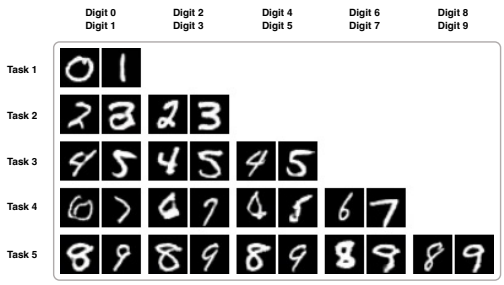
E.3 Visualization Results

We present visualization results of the generated images in this section. Figures 3, 4, 5, 6, 7, 8, and 9 showcase synthesized images in the label-conditioned CLoG across the seven datasets in our benchmark. We visualize the synthesized images from the models over the last five tasks using the first class order, with images selected randomly to avoid cherry-picking. We select five representative methods to showcase the results: NCL, Non-CL, ER (replay-based), EWC (regularization-based), and Ensemble (parameter-isolation based).

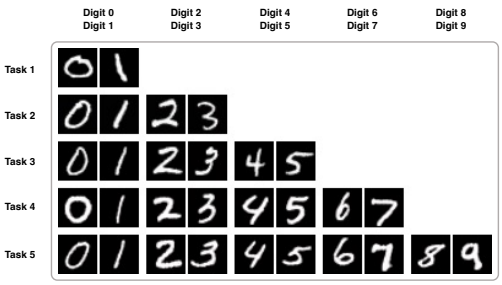
As shown in the figure, a naive way of CL without additional techniques leads to severe forgetting. The regularization-based methods can preserve knowledge of previous tasks to some extent, but the results are still far from satisfying, especially as the number of learning tasks increases. Replay-based methods significantly mitigate the challenges of catastrophical forgetting. However, our empirical studies suggest that they are prone to mode collapse when training GANs, mainly due to the limited size of the replay memory. This may reveal a novel challenge in CLoG compared to traditional classification-based CL. Furthermore, the ensembling method achieves superior performance on each task on the first three datasets, including MNIST, Fashion-MNIST and CIFAR-10. Nevertheless, it synthesizes images with relatively low quality on the other three datasets (see Fig. 6, 7, 8). Take Oxford-Flowers as an example, the separate model trained on each task fail to capture the correct structures of flowers, in contrast to other CL methods. This verifies our analysis that knowledge transfer among different tasks contribute to performance boost on these datasets.

Figures 10, and 11, showcase synthesized images in the concep-conditional CLoG with Custom Diffusion [49] and DreamBooth [78] in our benchmark. We visualize the synthesized images from the models over the five tasks using the third class order, with images selected randomly to avoid cherry-picking. We select five representative methods to showcase the results: NCL, Non-CL, KD (regularization-based), EWC (regularization-based), and Ensemble (parameter-isolation-based).

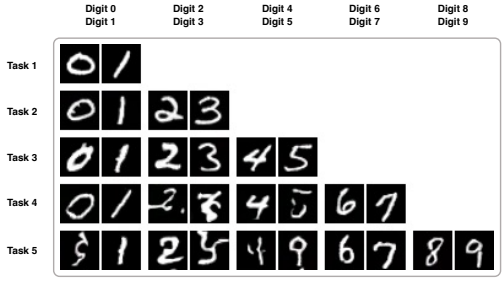
A naive method of continual learning without additional techniques produces relatively high-quality images, particularly when using the DreamBooth method with more training parameters. Regularization-based methods can preserve knowledge from previous tasks to some extent, but the results are still unsatisfactory, especially as the number of learning tasks increases. For example, in Figures 11, the EWC method shows that by the fifth task, the Custom Diffusion has almost entirely forgotten the color of the bear plushie and the blue hat decoration. Furthermore, Ensemble method achieves superior performance with both Custom Diffusion and DreamBooth.



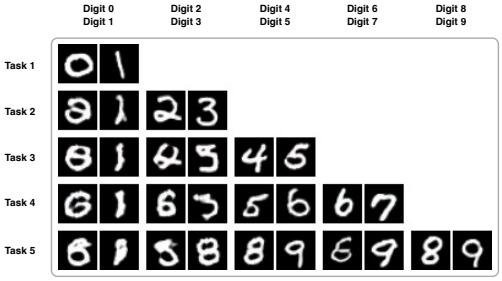
(a) NCL (GAN)



(b) Non-CL (GAN)

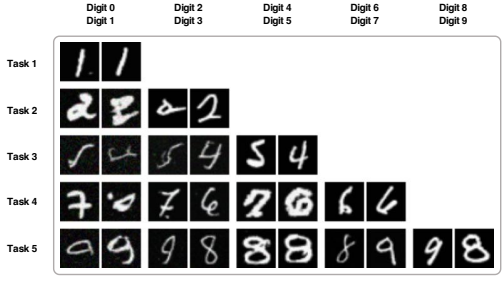


(c) ER (GAN)

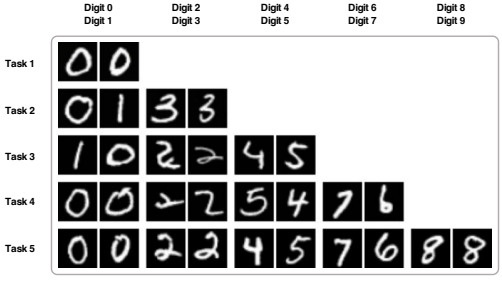


(d) EWC (GAN)

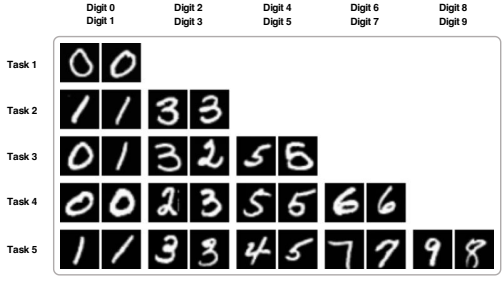
(e) Ensemble (GAN)



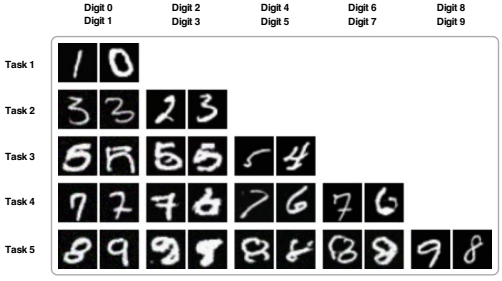
(f) NCL (DDIM)



(g) Non-CL (DDIM)



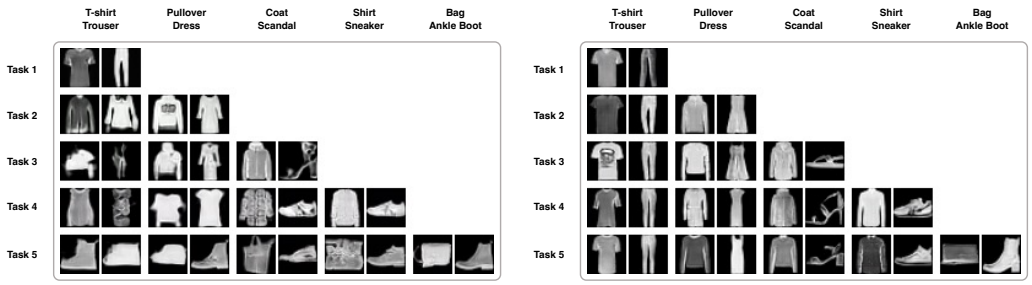
(h) ER (DDIM)



(i) EWC (DDIM)

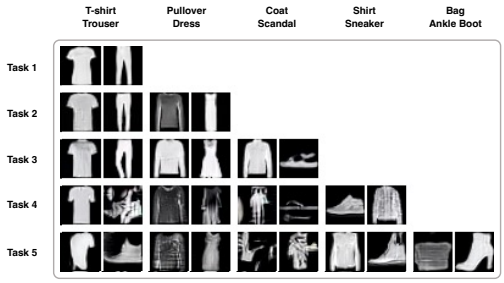
(j) Ensemble (DDIM)

Figure 3: Visualization results of label-conditioned CLoG on the MNIST [15] dataset.

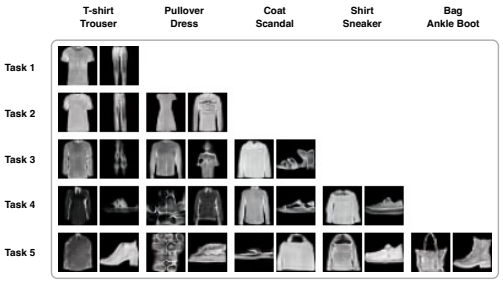


(a) NCL (GAN)

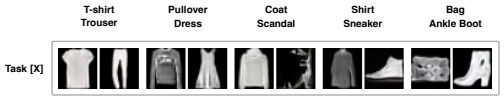
(b) Non-CL (GAN)



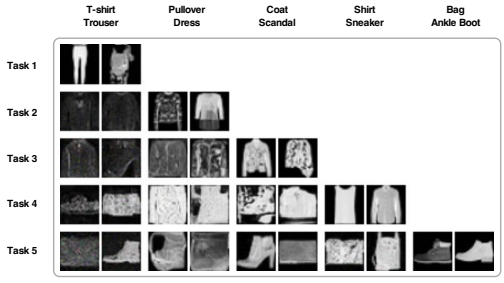
(c) ER (GAN)



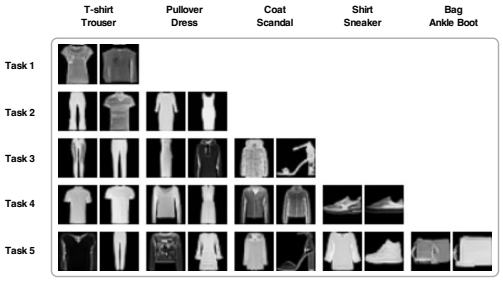
(d) EWC (GAN)



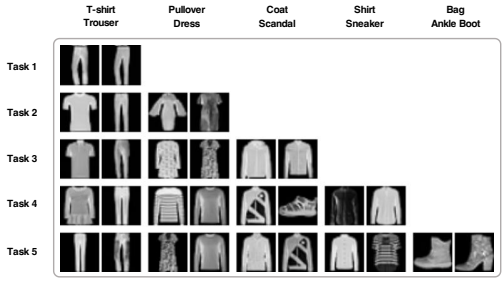
(e) Ensemble (GAN)



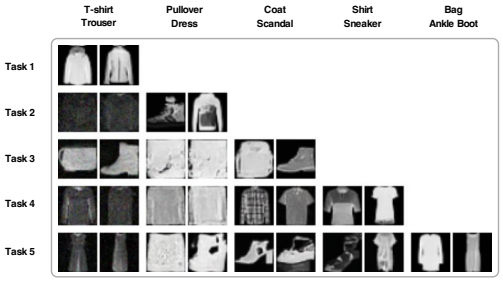
(f) NCL (DDIM)



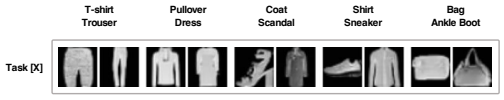
(g) Non-CL (DDIM)



(h) ER (DDIM)

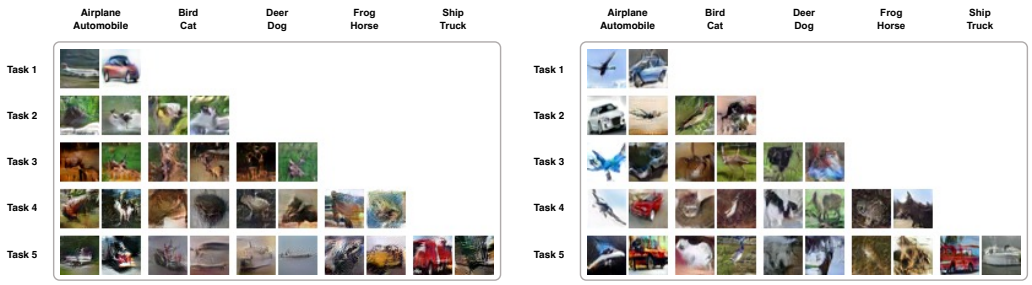


(i) EWC (DDIM)



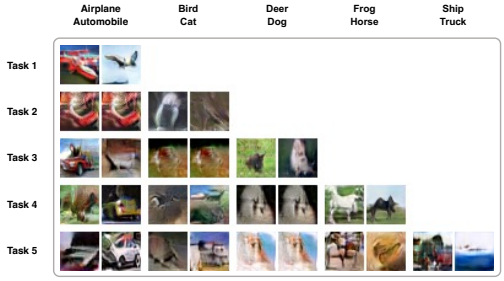
(j) Ensemble (DDIM)

Figure 4: Visualization results of label-conditioned CLoG on the Fashion-MNIST [116] dataset.

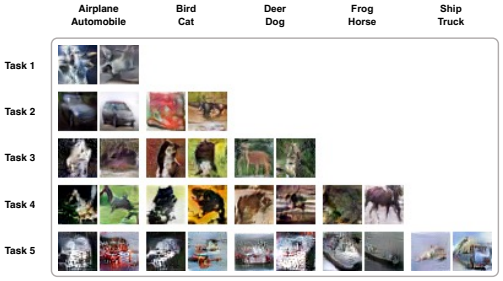


(a) NCL (GAN)

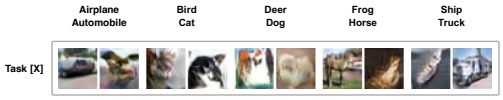
(b) Non-CL (GAN)



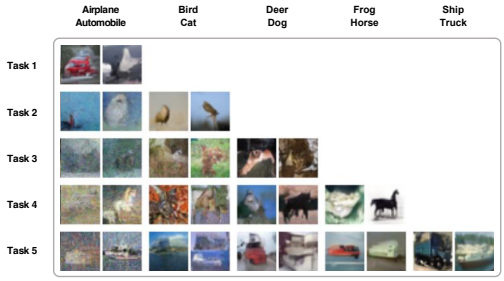
(c) ER (GAN)



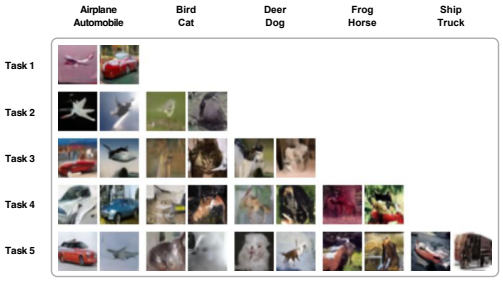
(d) EWC (GAN)



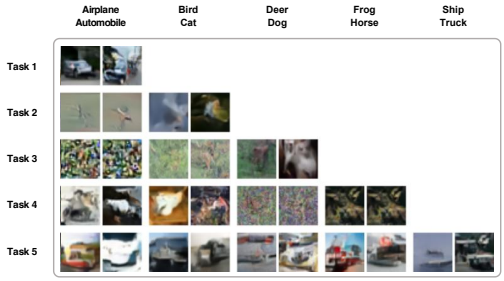
(e) Ensemble (GAN)



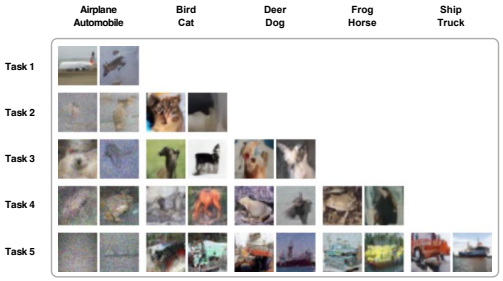
(f) NCL (DDIM)



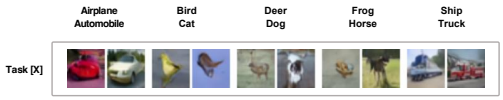
(g) Non-CL (DDIM)



(h) ER (DDIM)

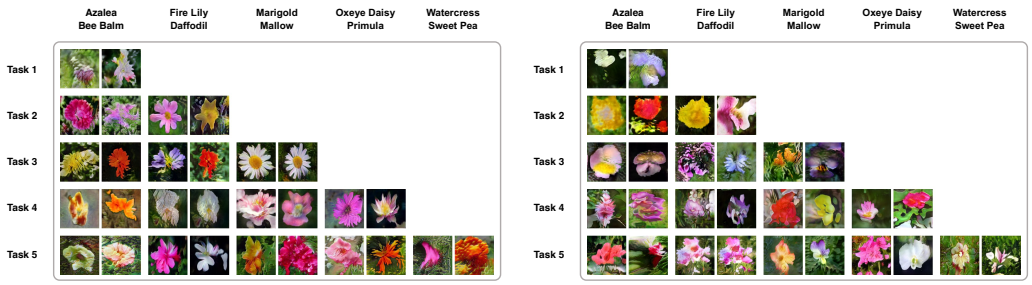


(i) EWC (DDIM)



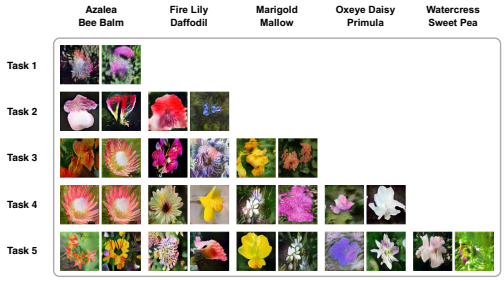
(j) Ensemble (DDIM)

Figure 5: Visualization results of label-conditioned CLoG on the CIFAR-10 [47] dataset.

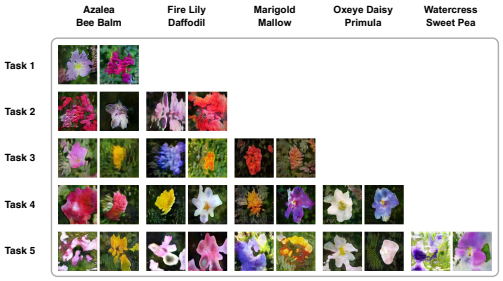


(a) NCL (GAN)

(b) Non-CL (GAN)

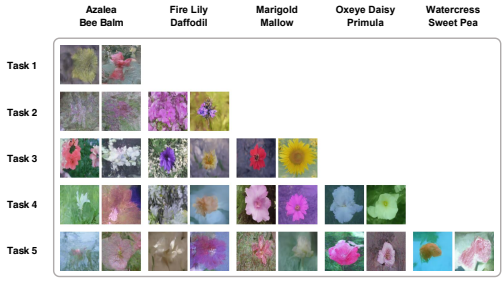


(c) ER (GAN)

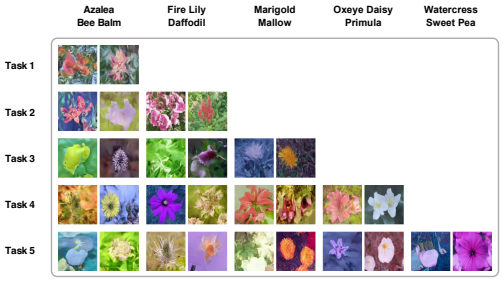


(d) EWC (GAN)

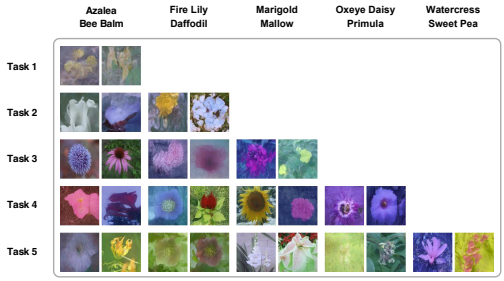
(e) Ensemble (GAN)



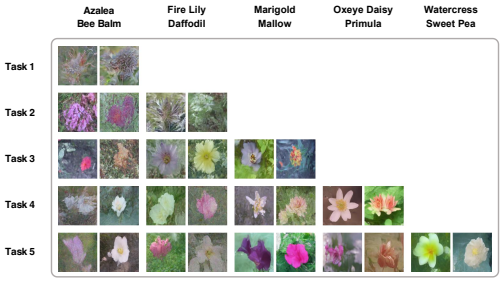
(f) NCL (DDIM)



(g) Non-CL (DDIM)



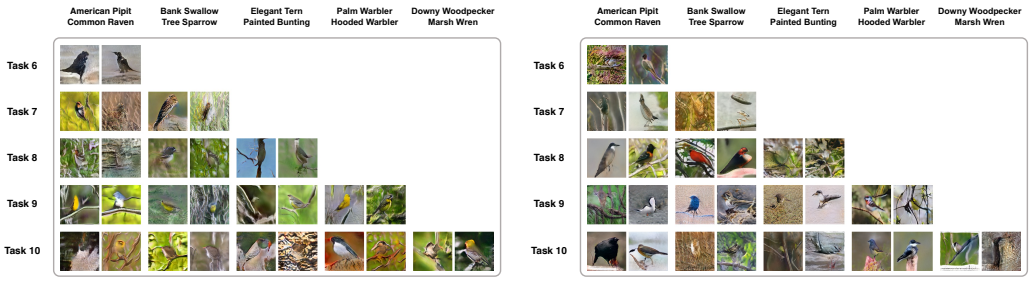
(h) ER (DDIM)



(i) EWC (DDIM)

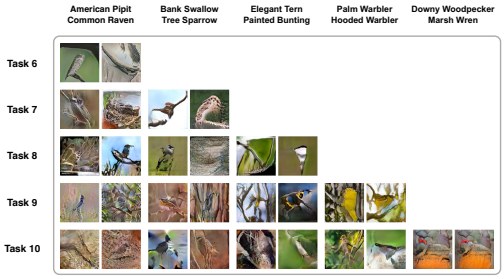
(j) Ensemble (DDIM)

Figure 6: Visualization results of label-conditioned CLoG on the Oxford-Flowers [66] dataset.

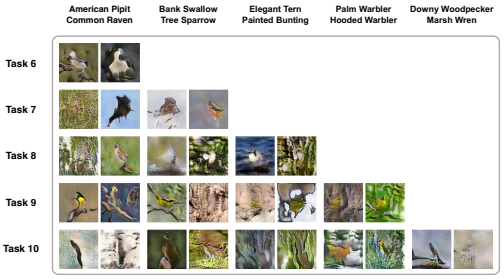


(a) NCL (GAN)

(b) Non-CL (GAN)

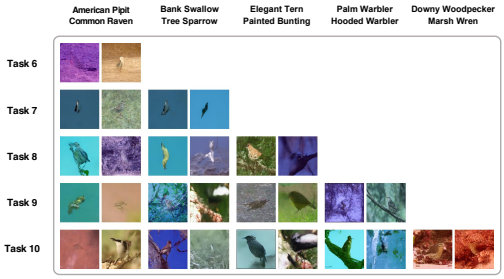


(c) ER (GAN)

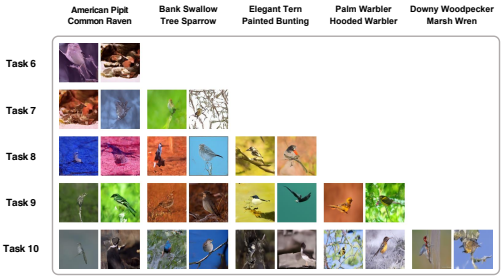


(d) EWC (GAN)

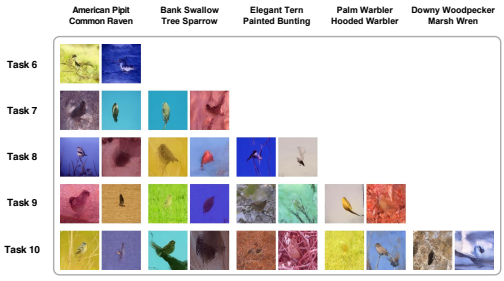
(e) Ensemble (GAN)



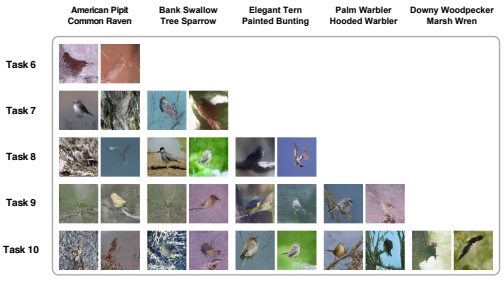
(f) NCL (DDIM)



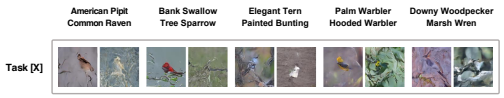
(g) Non-CL (DDIM)



(h) ER (DDIM)

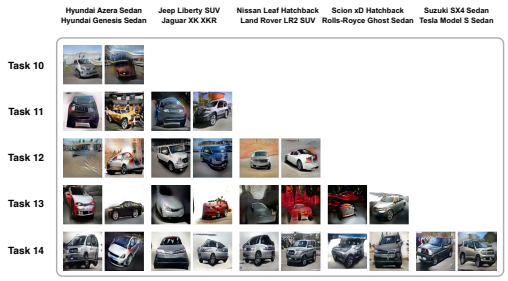


(i) EWC (DDIM)

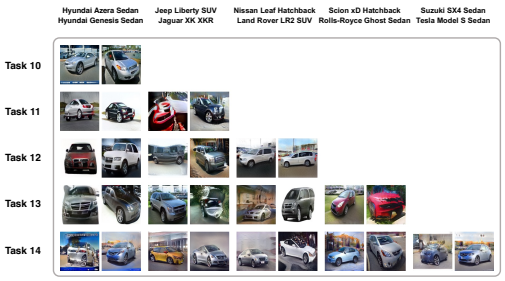


(j) Ensemble (DDIM)

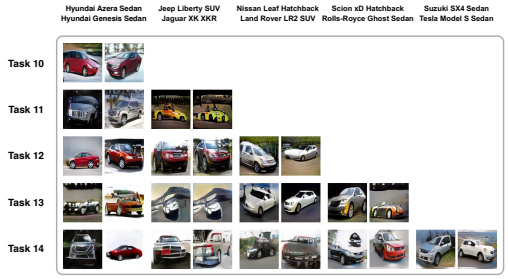
Figure 7: Visualization results of label-conditioned CLoG on the CUB-Birds [108] dataset.



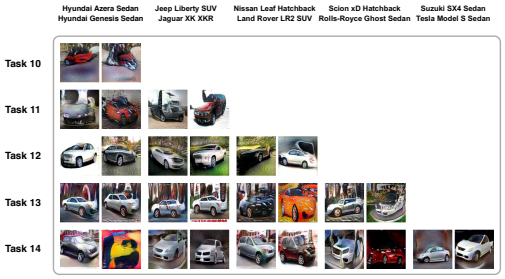
(a) NCL (GAN)



(b) Non-CL (GAN)

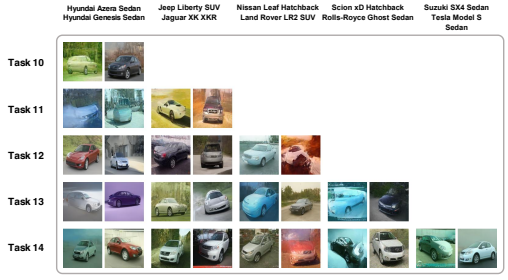


(c) ER (GAN)

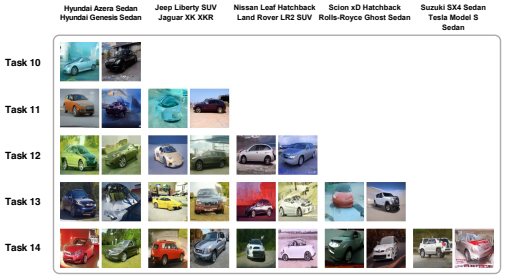


(d) EWC (GAN)

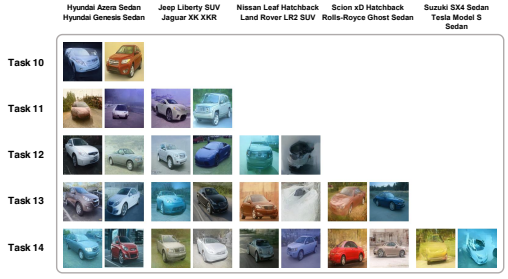
(e) Ensemble (GAN)



(f) NCL (DDIM)



(g) Non-CL (DDIM)

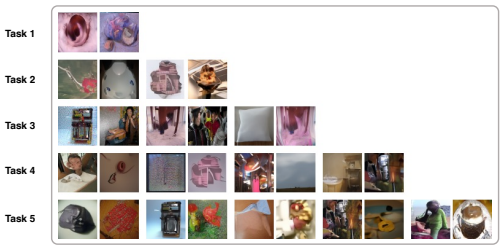


(h) ER (DDIM)

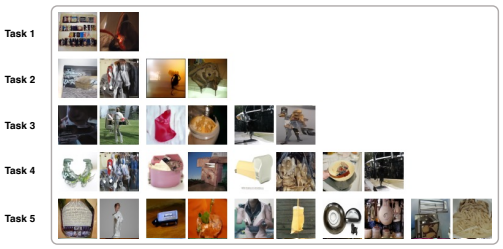
(i) EWC (DDIM)

(j) Ensemble (DDIM)

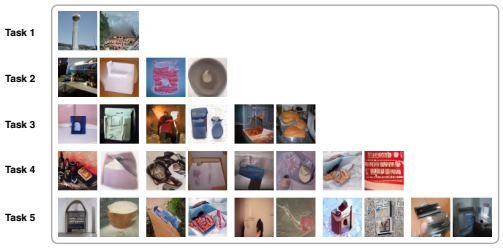
Figure 8: Visualization results of label-conditioned CLoG on the Stanford-Cars [46] dataset.



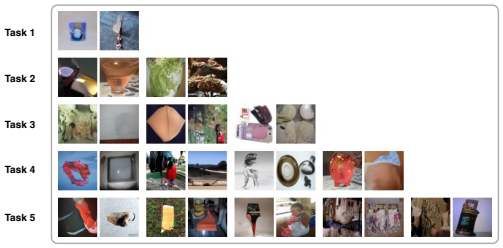
(a) NCL (DDIM)



(b) Non-CL (DDIM)



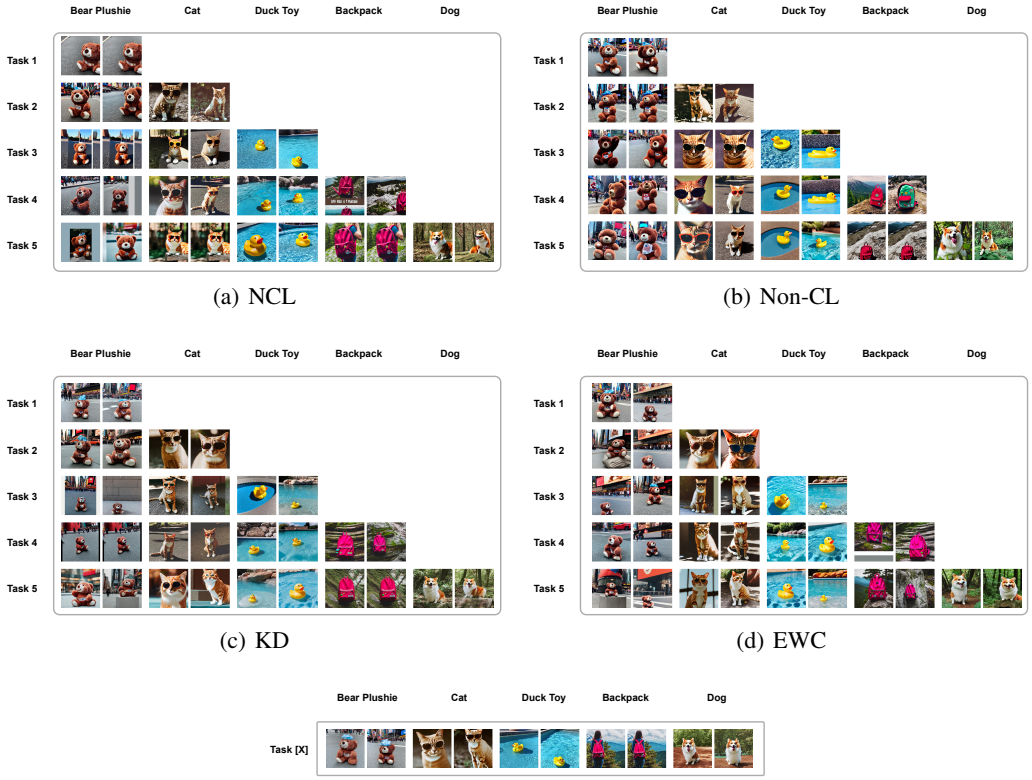
(c) KD (DDIM)



(d) EWC (DDIM)

(e) Ensemble (DDIM)

Figure 9: Visualization results of label-conditioned CLoG on the ImageNet-1k [79] dataset.



(a) NCL (b) Non-CL

(c) KD (d) EWC

(e) Ensemble

Figure 10: Visualization results of concept-conditioned CLoG on the Custom Objects [101] dataset utilizing DreamBooth [78].

(a) NCL (b) Non-CL

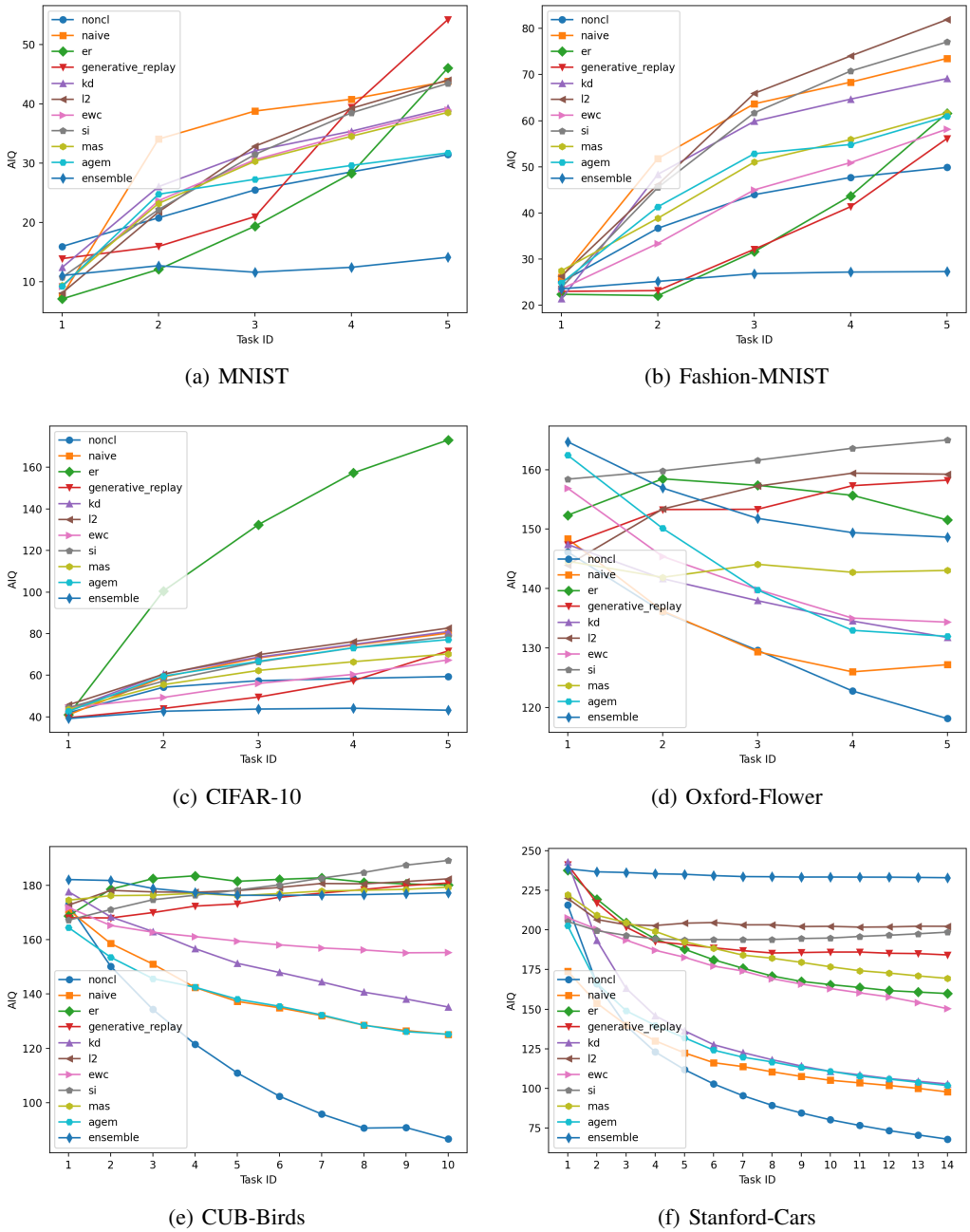
(c) KD (d) EWC

(e) Ensemble

Figure 11: Visualization results of concept-conditioned CLoG on the Custom Objects [101] dataset utilizing Custom Diffusion [49].

E.4 Comprehensive AIQ results for each task

To comprehensively investigate the performance of AIQ when increasing the number of learning tasks, we visualize its evolving curve in Fig. 12 and 13, corresponding to GANs and diffusion models, respectively. Generally, the curve exhibits an upward trend, indicating a tendency to forget the knowledge of previous tasks. However, the AIQ metric gradually decreases on the CUB-Birds, Oxford-Flowers, and Stanford-Cars datasets, demonstrating that incremental learning of similar tasks enhances performance on previous tasks.

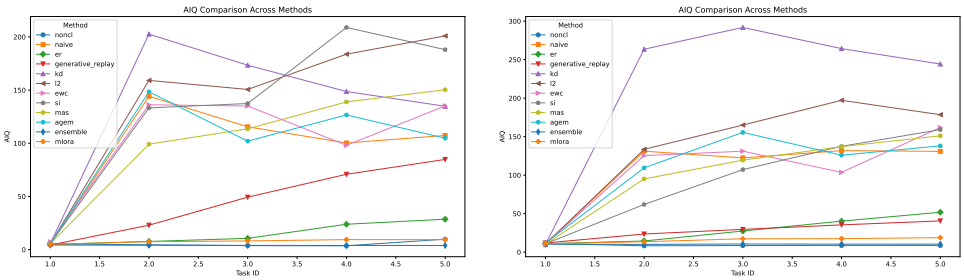


(a) MNIST (b) Fashion-MNIST

(c) CIFAR-10 (d) Oxford-Flower

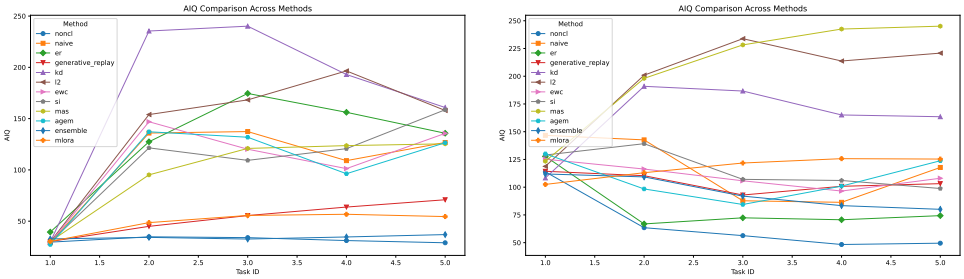
(e) CUB-Birds (f) Stanford-Cars

Figure 12: The evolving performance curve of AIQ across various tasks on label-conditioned CLoG benchmarks. Here GANs are employed as the generator backbone.



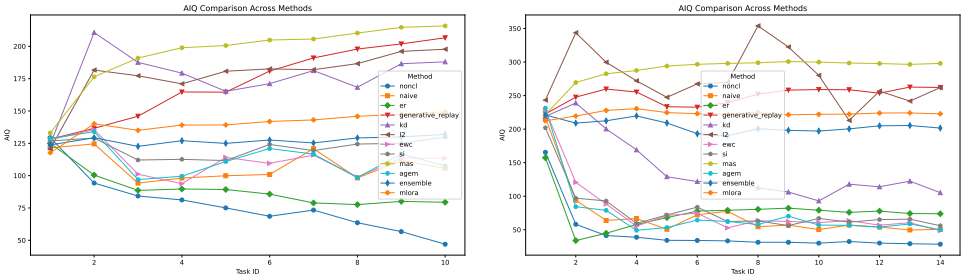
(a) MNIST (DDIM)

(b) Fashion-MNIST (DDIM)



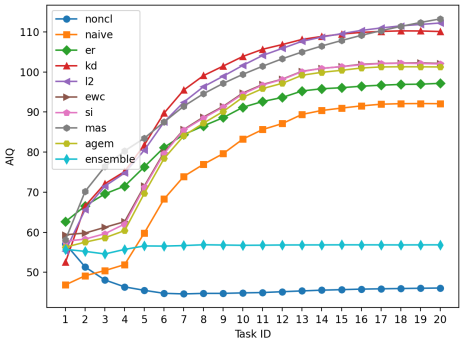
(c) CIFAR-10 (DDIM)

(d) Oxford-Flower (DDIM)



(e) CUB-Birds (DDIM)

(f) Stanford-Cars (DDIM)

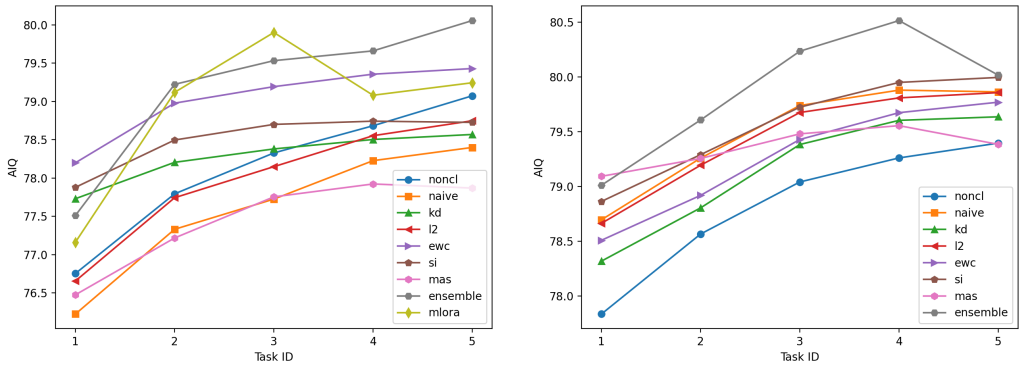


(g) ImageNet-1k (DDIM)

Figure 13: The evolving performance curve of AIQ across various tasks on label-conditioned CLoG benchmarks. Here diffusion models are employed as the generator backbone.

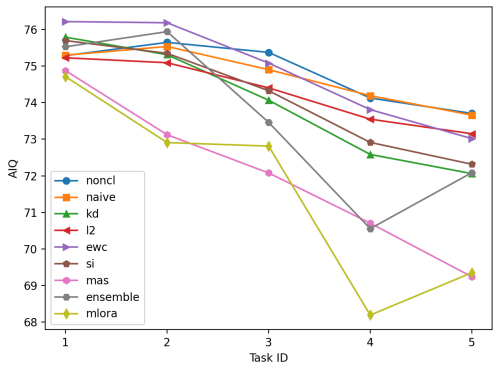
We also visualize evolving curve in Fig. 14 on DreamBooth and Custom Diffusion models, respectively. If we use CLIP avg to calculate AIQ, the curve exhibits an upward trend, indicating a tendency to forget the knowledge of previous tasks. On the other hand, if we use DINO avg to calculate AIQ, the metric gradually decreases for both the DreamBooth and Custom Diffusion Methods. This demonstrates that incremental learning of similar tasks enhances performance on previous tasks,

which is consistent with the actual results of our generated images in Figures 10, and 11. We prefer the AIQ calculated by DINO avg because DINO is not trained to ignore differences between subjects of the same class. Instead, its self-supervised training objective encourages the distinction of unique features of a subject or image.

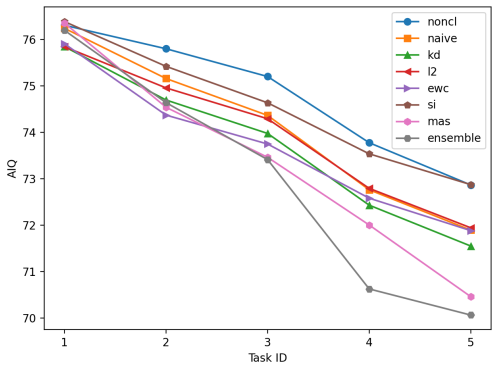


(a) DreamBooth CLIP avg

(b) Custom Diffusion CLIP avg



(c) DreamBooth DINO avg



(d) Custom Diffusion DINO avg

Figure 14: The evolving performance curve of AIQ across various tasks on concept-conditioned CLoG benchmarks. We show the results on the Custom-Objects dataset utilizing DreamBooth [78] and Custom Diffusion [49].

E.5 Computational Budget Analysis

In this section, we present the memory consumption and training time for different baselines. Notice that we use a mix of different types of GPUs for different set of experiments: We use a single NVIDIA V100 GPU for GAN, a single NVIDIA RTX4090 GPU for DDIM, a single NVIDIA A100 GPU for ImageNet-1k, and a single NVIDIA A800 GPU for Costom-Objects.

Now we present a detailed analysis of memory consumption of each baseline method. Methods that require replay samples (ER and A-GEM) introduce an auxiliary replay memory to retain previous data. In addition, all regularization-methods require storing the parameters of a teacher model, doubling the total number of model parameters. Among these techniques, EWC, SI and MAS require additional computation for determining the loss weight of each parameter, resulting in a threefold increase in the model's parameter count. Lastly, the ensemble method increases memory consumption by a factor of T (where T represents the total number of tasks), while C-LoRA introduces T additional trainable weights to facilitate conditional generation.

It is noted that DDIM-based GR is significantly slow, as generating the replay samples for DDIM requires multiple denoising steps (50 in our implementation). It is also noted that the parameter-isolation-based methods have linearly increasing memory consumption when the number of tasks increases, while other methods only consume constant memory budget.

Table 11: Memory Consumption of label-conditioned CLoG benchmarks: Measured in number of parameters (M)

	MNIST	**Fashion-MNIST**	**CIFAR-10**	**CUB-Birds**	**Oxford-Flowers**	**Stanford-Cars**	**ImageNet**
- GAN							
Non-CL	43.30	43.30	43.30	60.55	60.55	60.55	NA
NCL	43.30	43.30	43.30	60.55	60.55	60.55	NA
ER	43.92	43.92	43.92	70.38	70.38	70.38	NA
GR	86.60	86.60	86.60	121.10	121.10	121.10	NA
KD	86.60	86.60	86.60	121.10	121.10	121.10	NA
L2	86.60	86.60	86.60	121.10	121.10	121.10	NA
EWC	129.91	129.91	129.91	181.65	181.65	181.65	NA
SI	129.91	129.91	129.91	181.65	181.65	181.65	NA
MAS	129.91	129.91	129.91	181.65	181.65	181.65	NA
A-GEM	43.92	43.92	43.92	70.38	70.38	70.38	NA
Ensemble	216.52	216.52	216.52	605.49	302.75	847.69	NA
- Diffusion Model							
Non-CL	37.20	37.20	37.20	85.51	85.51	85.51	346.09
NCL	37.20	37.20	37.20	85.51	85.51	85.51	346.09
ER	37.40	37.40	37.40	88.71	88.71	88.71	348.55
GR	74.40	74.40	74.40	171.02	171.02	171.02	NA
KD	74.40	74.40	74.40	171.02	171.02	171.02	692.18
L2	74.40	74.40	74.40	171.02	171.02	171.02	692.18
EWC	111.60	111.60	111.60	256.53	256.53	256.53	778.71
SI	111.60	111.60	111.60	256.53	256.53	256.53	778.71
MAS	111.60	111.60	111.60	256.53	256.53	256.53	778.71
A-GEM	37.40	37.40	37.40	88.71	88.71	88.71	348.55
Ensemble	186.00	186.00	186.00	855.10	427.55	1197.74	6921.84
C-LoRA	43.00	43.00	43.00	103.11	94.31	110.15	476.45

Table 12: Memory Consumption of concept-conditioned CLoG benchmarks: Measured in number of parameters (M)

Metric	Model	**NCL**	**Non-CL**	**KD**	**L2**	**EWC**	**SI**	**MAS**	**Ensemble**	**C-LoRA**
All Params	DreamBooth	1016.84	1016.84	1953.9	1953.9	2890.97	2890.97	2890.97	5084.2	1068.84
	Custom Diffusion	1035.12	1035.12	1990.47	1089.59	1144.06	1144.06	1144.06	5175.6	-
Train Params	DreamBooth	937.06	937.06	937.06	937.06	937.06	937.06	937.06	4685.3	10.4
	Custom Diffusion	54.47	54.47	54.47	54.47	54.47	54.47	54.47	272.35	-

Table 13: **Training Time of different baselines on the label-conditioned CLoG benchmark**: Measured in hours over all tasks

	MNIST	**Fashion-MNIST**	**CIFAR-10**	**CUB-Birds**	**Oxford-Flowers**	**Stanford-Cars**	**ImageNet**
- GAN							
Non-CL	37.65	37.78	32.65	80.32	18.93	79.81	NA
NCL	12.81	12.62	10.82	14.78	6.26	10.64	NA
ER	16.32	15.21	12.66	16.58	7.10	11.81	NA
GR	15.76	14.96	12.78	17.30	7.18	12.20	NA
KD	15.64	15.48	13.38	16.94	7.06	12.07	NA
L2	12.92	12.69	10.74	14.96	6.38	10.51	NA
EWC	15.12	14.92	12.62	20.72	8.54	15.06	NA
SI	16.84	16.52	14.02	25.04	10.14	17.92	NA
MAS	15.16	15.12	12.74	21.17	8.72	14.93	NA
A-GEM	15.68	15.52	13.42	19.19	8.10	13.58	NA
Ensemble	12.64	12.58	10.93	14.51	6.34	10.51	NA
- Diffusion Model							
Non-CL	16.11	15.30	12.88	55.01	13.33	58.33	3953.34
NCL	2.83	3.19	3.17	3.33	3.33	10.04	103.84
ER	2.89	2.83	2.55	9.83	5.61	5.53	104.44
GR	6.9	8.83	10.22	12.64	7.89	16.80	NA
KD	3.56	2.56	3.11	7.12	6.88	8.52	135.67
L2	2.94	3.72	2.73	3.98	3.22	4.89	105.33
EWC	3.44	2.65	2.94	9.09	7.38	8.89	121.86
SI	3.89	4.64	5.05	10.72	6.64	4.59	145.86
MAS	3.96	4.14	2.89	7.37	5.94	13.85	109.44
A-GEM	3.87	3.89	2.87	13.92	12.60	10.50	104.94
Ensemble	3.34	2.67	2.22	6.12	4.94	5.83	102.31
C-LoRA	2.04	2.37	2.72	4.10	2.78	3.29	128.88

Table 14: **Training Time of different baselines on the concept-conditioned CLoG benchmark**: Measured in minutes over all tasks

Model	**NCL**	**Non-CL**	**KD**	**L2**	**EWC**	**SI**	**MAS**	**Ensemble**	**C-LoRA**
DreamBooth	26.76	6.29	32.54	33.09	69.79	172.83	38.68	31.58	25.66
Custom Diffusion	12.40	4.02	16.65	12.49	13.33	15.45	12.25	10.15	-

F Dataset and Task design

F.1 Dataset description

The *training datasets* are summarized in Table 15 and introduced as follows.

MNIST [50] contains 60,000 grayscale images of handwritten digits (0-9) in a 28×28 pixel format. We resize the images to 32×32 resolution for image generation.

FasionMNIST [116] consists of 60,000 grayscale images across 10 fashion categories, such as shirts, dresses, and shoes. The images are also resized from 28×28 to 32×32 for image generation.

CIFAR-10 [47] consists of 60,000 colored images sized at 32×32 pixels, divided into 10 classes including airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck.

ImageNet-1k [79] is the most commonly used subset of ImageNet, which spans 1000 classes and contains 1,281,167 training images. We use the down-sampled 64×64 version following Chrabaszcz et al. [9].

Oxford-Flower [66] consists of 102 flower categories that commonly occur in the United Kingdom. Each class consists of between 40 and 258 images, with a total of 7,169 images. The images are resized to 128×128 for generation.

CUB-Birds [108] contains 11,788 images of 200 subcategories belonging to birds, which is a widely-used dataset for fine-grained visual categorization task. We resize the images to 128×128 for generation.

Stanford-Cars [46] consists of 196 classes of cars with a total of 8,144 images. The images are also resized to 128×128 for image generation.

Custom-Objects [101] contains 5 customized concepts from users with paired text-image demonstrations. Each concept has 5 demonstrations with 512×512 image resolution. The task is to generate the customized concepts given arbitrary text conditions.

Table 15: The detailed configurations of eight CLoG benchmarks studied in this paper.

Dataset	Image Resolution	#Training Images per Task	#Tasks	Description of Each Task
MNIST [50]	32×32	12,000	5	Conditional generation of 2 classes of handwritten digits
FashionMNIST [116]	32×32	12,000	5	Conditional generation of 2 classes of fashion products
CIFAR-10 [47]	32×32	10,000	5	Conditional generation of 2 classes of common objects
ImageNet-1k [79]	64×64	$\sim$64,000	20	Conditional generation of 50 classes of ImageNet images
Oxford-Flower [66]	128×128	$\sim$1,400	5	Conditional generation of 20 categories of flowers
CUB-Birds [108]	128×128	$\sim$1,200	10	Conditional generation of 20 species of birds
Stanford-Cars [46]	128×128	$\sim$600	14	Conditional generation of 14 classes of cars
Custom-Objects [101]	512×512	5	5	Generate a customized object given text conditions

F.2 Task Sequences

We partitioned MNIST, FashionMNIST, and CIFAR-10 into five tasks, assigning two classes to each task. ImageNet-1k was divided into 20 tasks with 50 classes per task, Oxford-Flower into five tasks with 20 categories per task, CUB-Birds into 10 tasks with 20 categories per task, Stanford-Cars into 14 tasks with 14 classes per task, and Custom-Objects into five tasks with one object per task. Following the random class order protocol in Rebuffi et al. [75], we generate five different class orders for each experiment and report their averaged metrics over five random orders. For a fair comparison, the class orderings are fixed in our experiments (see Appendix G.5). It is important to note that one dataset can be segmented into varying numbers of tasks [55] or without requiring uniformity in class [24]. These customized CL settings can be explored in the future, and our current benchmark focuses on addressing more fundamental challenges in CLoG for now.

G Implementation

G.1 Overall description

To ensure the fair comparison across methods, we follow Section 3.4 to use unified settings with common hyperparameters and architecture choices. We follow Heusel et al. [25] to compute FID using the entire training dataset as reference images. To achieve the best training performance, we compute the quality metrics on current task every 500 steps and save the best checkpoint. If the method has CL-related hyper-parameters (e.g., regularization weights), we will search for 8 values across different magnitudes and pick the hyper-parameter based on the quality metrics. We found it's hard to train GAN on the long-sequence and large-scale ImageNet-1k benchmark, so we leave it as "NA" (Not A Number). We didn't implement C-LoRA on GAN and Custom Diffusion as it is not applicable. We follow Lin et al. [55] to use reservoir replay buffer [107] for ER with buffer sizes as 5000 samples for ImageNet-1k, and 200 for the other label-conditioned CLoG benchmarks. Replay-based methods are excluded in Custom-Object as it has very few training samples and thus replay is equivalent to Non-CL.

G.2 Label-conditional CLoG

Implementation Details of StyleGAN2 We employ the official PyTorch implementation of StyleGAN2-ADA [34] as our backbone. The detailed hyperparameters used in our experiments are presented in Table 16. All training runs are performed for 200 epochs using a single NVIDIA Tesla V100 GPU. We utilize six datasets with different image resolutions: 32x32 pixels (MNIST, Fashion-MNIST, CIFAR-10) and 128x128 pixels (CUB-Birds, Oxford-Flowers, Stanford-Cars). Two variants of StyleGAN2 are implemented to generate images at these resolutions, termed Ours-S and Ours-L, respectively.

We use a minibatch size of 64 for Ours-S and 16 for Ours-L. For the replay-based methods in CLoG, we construct a replay memory containing 200 samples from previous tasks, with the replay size set to one-fourth of the minibatch size (16 for Ours-S and 4 for Ours-L). Following the configuration for CIFAR-10 in the original paper [34], we use 512 feature maps for all layers. The weight of the R_1 regularization is set to $\gamma = 0.01$ for Ours-S and $\gamma = 1$ for Ours-L. Additionally, we opt for a more expressive model architecture for the mapping network and the discriminator when synthesizing images at 128x128 pixels. Specifically, we increase the depth of the mapping network from 2 to 8 and enable residual connections in the discriminator. For simplicity, we omit several techniques that are irrelevant to CL capability used in the original paper, including adaptive discriminator augmentation (ADA), style mixing, path length regularization, and exponential moving average (EMA).

Implementation Details of DDIM We employ the Huggingface diffuser[2] implementation of DDIM [96] in our codebase. The detailed hyperparameters used in our experiments are presented in Table 17. All training runs are performed for 200 epochs using a single NVIDIA RTX 4090 GPU for MNIST, Fasion-MNIST, CIFAR-10, CUB-Birds, Oxford-Flowers, Stanford-Cars, and a single NVIDIA A100 GPU for the large-scale ImageNet-1k dataset. Three variants of DDIM are implemented to generate images at small resolution (32x32), meddium resolution (64x64), large resolution (128x128), termed Ours-S, Ours-M, Ours-L, respectively. We use a minibatch size of 256 for Ours-S, 320 for Ours-M, 32 for Ours-L. For the replay-based methods, we maintain a replay buffer containing 200 samples from previous tasks with replay size as 64 for Ours-S and Ours-M, and 8 for Ours-L. Following Nichol and Dhariwal [65], we use different numbers of channel and UNet blocks for Ours-S, Ours-M, and Ours-L.

[2] https://huggingface.co/docs/diffusers

Table 16: Hyperparameters of StyleGAN2 [34] used in our CLoG experiments.

Parameter	Ours-S	Ours-L
Resolution	32×32	128×128
Training epochs	200	200
Minibatch size	64	16
Minibatch stddev	32	32
Replay size	64	16
Memory size	200	200
Feature maps	512	512
Learning rate $\eta \times 10^3$	2.5	2.5
R_1 regularizationγ	0.01	1
Mapping net depth	2	8
Resnet D	-	✓

Table 17: Hyperparameters of DDIM [96] used in our CLoG experiments.

Parameter	Ours-S	Ours-M	Ours-L
Resolution	32×32	64×64	128×128
Training epochs	200	100	200
Minibatch size	256	320	32
Replay size	64	64	8
Memory size	200	5000	200
Learning rate $\eta \times 10^3$	2.0	2.0	1.0
Learning rate warm-up steps	500	500	500
Weight decay	0.0	0.0	0.0
# Unet blocks (×2)	4	4	5
Unet blocks dimension (the largest)	256	512	512
Dropout	0.1	0.1	0.1
Time embedding dimension	512	512	512

G.3 Concept-conditional CLoG

Evaluation Metrics For concept-conditioned CLoG, we follow DreamBooth [78] and Custom Diffusion [49] to evaluate the alignment between generated image and the provided concept, and the text prompts, respectively. To assess subject fidelity, we use two metrics: CLIP Image Alignment and DINO Image Alignment. CLIP Image Alignment measures the average pairwise cosine similarity between the CLIP embeddings of generated and real images. Similarly, the DINO metric calculates the average pairwise cosine similarity between the ViT-S/16 DINO embeddings of generated and real images. To evaluate prompt fidelity, we compute the average cosine similarity between the CLIP embeddings of the text prompt and the images, which we refer to as CLIP Text Alignment. The averages of the image alignment and text alignment scores are combined to derive a single quality metric for straightforward comparison, labeled respectively as DINO avg and CLIP avg. We evaluate each task using 20 text prompts, generating 50 samples per prompt. This results in a total of 1,000 images generated for each task.

Implementation Details DreamBooth and Custom Diffusion both utilize generated by initial stable-diffusion-v1-4, rather than real, category images to calculate the prior loss for their training processes. 200 regularization images are preemptively created using a DDPM sampler over 50 steps with the prompt 'photo of a {category}'. We use DDPM sampling with 50 steps and a classifier-free guidance scale of 6 for both DreamBooth and Custom Diffusion. All training runs are performed using a single NVIDIA A800 GPU. More details can be found in Table 18

DreamBooth adheres to the same data augmentation strategies as Custom Diffusion, which will be introduced later, to ensure a balanced comparison. It trains by fine-tuning both a text transformer and a U-net diffusion model. This training uses a batch size of 1 and a learning rate of 2e-6, which is maintained constant regardless of the number of GPUs or batch size. For generating target images, DreamBooth employs a text prompt formatted as 'photo of a [V] {category}', where '[V]' is replaced with a rarely used token from a specific set ('sks', 'phol', 'oxi', 'mth', 'nigh'). Each training task undergoes 800 steps. Conversely, Custom Diffusion uses a slightly different approach by setting the batch size at 2 and a scaled learning rate of 2e-5, adjusted according to the batch size to an effective rate of 4e-5. It trains each task for only 250 steps. During training, target images undergo random resizing: they are enlarged to between 1.2 and 1.4 times their original size every third iteration, with phrases like 'zoomed in' or 'close up' added to the text prompts. Other times, images are resized to between 0.4 and 1.0 times their original size; when the resizing ratio is below 0.6, terms like 'far away' or 'very small' are incorporated into the prompts, focusing loss propagation only within the valid image regions. The training captions, such as 'photo of a V* dog', incorporate a rare token ('ktn', 'pll', 'ucd', 'mth', 'nigh'), with both the token embedding and the cross-attention parameters being optimized during the training process.

Table 18: Hyperparameters used in our Concept-conditional CLoG.

Parameter	DreamBooth	DreamBooth-C-LoRA	Custom Diffsuion
Resolution	512×512	512×512	512×512
Training steps	800	800	250
Minibatch size	1	1	2
Inference steps	50	50	50
Learning rate	2e-6	5e-5	2e-5
Learning rate scheduler	constant	constant	constant
Learning rate warm-up steps	0	0	0
Prior loss	✓	✓	✓
Prior class images	200	200	200
Data Augmentation	✓	✓	✓

G.4 Class description

We list the class description for each label index for each dataset as follows.

- **MNIST** (10 classes)
  - digit ‘0’, digit ‘1’, digit ‘2’, digit ‘3’, digit ‘4’, digit ‘5’, digit ‘6’, digit ‘7’, digit ‘8’, digit ‘9’
- **FasionMNIST** (10 classes)
  - T-shirt/top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, Ankle boot
- **CIFAR-10** (10 classes)
  - airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck
- **ImageNet-1k** (1,000 classes)
  - tench Tinca tinca, goldfish Carassius auratus, great white shark white shark man-eater man-eating shark Carcharodon carcharias, tiger shark Galeocerdo cuvieri, hammerhead hammerhead shark, electric ray crampfish numbfish torpedo, stingray, cock, hen, ostrich, **(... 980 classes are omitted)** coral fungus, agaric, gyromitra, stinkhorn carrion fungus, earthstar, hen-of-the-woods hen of the woods Polyporus frondosus Grifola frondosa, bolete, ear spike capitulum, toilet tissue toilet paper bathroom tissue
- **Oxford-Flowers** (103 classes)
  - alpine sea holly, anthurium, artichoke, azalea, ball moss, balloon flower, barbeton daisy, bearded iris, bee balm, bird of paradise, **(... 980 classes are omitted)**, toad lily, tree mallow, tree poppy, trumpet creeper, wallflower, water lily, watercress, wild pansy, windflower, yellow iris
- **CUB-Birds** (200 classes)
  - Black footed Albatross, Laysan Albatross, Sooty Albatross, Groove billed Ani, Crested Auklet, Least Auklet, Parakeet Auklet, Rhinoceros Auklet, Brewer Blackbird, Red winged Blackbird, **(... 180 classes are omitted)**, Red headed Woodpecker, Downy Woodpecker, Bewick Wren, Cactus Wren, Carolina Wren, House Wren, Marsh Wren, Rock Wren, Winter Wren, Common Yellowthroat
- **Stanford-Cars** (196 classes)
  - AM General Hummer SUV 2000, Acura RL Sedan 2012, Acura TL Sedan 2012, Acura TL Type-S 2008, Acura TSX Sedan 2012, Acura Integra Type R 2001, Acura ZDX Hatchback 2012, Aston Martin V8 Vantage Convertible 2012, Aston Martin V8 Vantage Coupe 2012, Aston Martin Virage Convertible 2012, **(... 176 classes are omitted)** Toyota Camry Sedan 2012, Toyota Corolla Sedan 2012, Toyota 4Runner SUV 2012, Volkswagen Golf Hatchback 2012, Volkswagen Golf Hatchback 1991, Volkswagen Beetle Hatchback 2012, Volvo C30 Hatchback 2012, Volvo 240 Sedan 1993, Volvo XC90 SUV 2007, smart fortwo Convertible 2012
- **Custom-Objects** (5 concepts)
  - dog, duck toy, cat, backpack, bear plushie

G.5 Random class ordering

Table 19 shows the different class orderings we used on different dataset. Due to space limitation, we only show the ordering of datasets with small class sequences. For large sequences, we refer readers to check our supplemental materials for details. The first class sequence is set as the sequence of class ordering from the original dataset, while the other sequences are generated via random shuffling.

Table 19: The random class ordering used in our benchmarks. The full orderings can be found in our supplemental materials.

Dataset	Class order	Class sequence
MNIST, FasionMNIST, CIFAR-10	1	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
	2	3, 9, 1, 8, 0, 2, 6, 4, 5, 7
	3	6, 0, 2, 8, 1, 9, 7, 3, 5, 4
	4	2, 6, 1, 5, 9, 8, 0, 4, 3, 7
	5	1, 5, 7, 2, 0, 3, 4, 6, 8, 9
Custom-Objects	1	0, 1, 2, 3, 4
	2	4, 3, 1, 0, 2
	3	4, 2, 1, 3, 0
	4	1, 4, 0, 2, 3
	5	2, 1, 0, 3, 4

H Impact Statement

Our work is essential as it contributes to the advancement of generative models' continuous learning, potentially benefiting human lives and society. Our method approaches a general problem and will not have any direct negative impact or be misused in specific domains as long as the task itself is safe, ethical, and fair. The risks of these models should be evaluated based on the specific deployment context, including training data, existing guardrails, deployment environment, and authorized access.