ONLINE CONTINUAL LEARNING OF DIFFUSION MODELS: MULTI-MODE ADAPTIVE GENERATIVE DISTILLATION

Rui Yang, Matthieu Grard, Emmanuel Dellandréa, Liming Chen

ONLINE CONTINUAL LEARNING OF DIFFUSION MODELS: MULTI-MODE ADAPTIVE GENERATIVE DISTILLATION

Rui YANG[1], Matthieu Grard[2] , Emmanuel Dellandrea[1] & Liming CHEN[1]

[1]Ecole Centrale de Lyon, CNRS, Universite Claude Bernard Lyon 1
INSA Lyon, Université Lumière Lyon2, LIRIS, UMR5205, 69130 Ecully, France
[2]Siléane, 17 rue Descartes, 42 Saint-Etienne, France

ABSTRACT

Continual learning typically relies on storing real data, which is impractical in privacy-sensitive settings. Generative replay with diffusion models offers a high-fidelity alternative. However, in online continual learning (OCL), these models struggle with catastrophic forgetting and incur high computational costs from frequent updates and sampling. Existing distillation methods reduce generation steps but rely on a fixed teacher model, limiting their effectiveness as data distributions evolve. To address these, we introduce Multi-Mode Adaptive Generative Distillation (MAGD), which incorporates two innovative techniques: Noisy Intermediate Generative Distillation (NIGD) and SNR-Guided Generative Distillation (SGGD). **NIGD** leverages intermediate noisy images, created during the reverse process rather than by adding noise post-generation, to enhance knowledge transfer. **SGGD** uses a signal-to-noise ratio (SNR) based threshold to optimize the sampling of time steps, reducing unnecessary generation. Guided by an Exponential Moving Average (EMA) teacher, MAGD effectively mitigates catastrophic forgetting as it adapts to new data streams. Experiments on Fashion-MNIST, CIFAR-10, and CIFAR-100 show that MAGD reduces generation overhead by up to 25% relative to standard generative distillation and 92% compared to DDGR-1000, while maintaining generating quality. Furthermore, in class-conditioned diffusion models, MAGD outperforms memory-based methods in terms of classification accuracy.

Index Terms— Continual Learning, Online Learning, Diffusion Model

1. INTRODUCTION

Continual learning focuses on developing models that adapt to an evolving data stream without forgetting previously acquired knowledge[1]. A common strategy to manage this catastrophic forgetting is replay-based learning, where the model rehearses old examples as it encounters new ones. In principle, storing past data [2, 3] can effectively mitigate forgetting. However, such an approach faces privacy restrictions and storage limitations in real-world domains like healthcare, finance, and robotics. As an alternative, generative replay synthesizes representative samples from earlier distributions [4, 5], thus removing the need for retaining raw data and satisfying strict data-protection requirements.

Diffusion models have recently emerged as powerful generators for high-resolution, photorealistic images [6, 7], surpassing earlier generative paradigms in many settings. When applied to class-incremental learning [8, 9, 10], diffusion models can effectively recreate data from previously learned classes. However, these techniques typically presume well-defined tasks: each time a new task appears, the model generates replay samples for that specific task. This rigid structure overlooks two major challenges in online continual learning (OCL), where data arrive as a continuous, unordered stream without clear task boundaries. First, diffusion models generally rely on iterative denoising processes, sometimes requiring hundreds or thousands of steps per sample. In an online continual learning (OCL) setting, where replay could be triggered by each data batch, this expense rapidly becomes prohibitive. Second, the diffusion model can suffer catastrophic forgetting if they cannot revisit earlier distributions. The problem is exacerbated by the absence of explicit task labels or boundaries, forcing the model to update continually without full knowledge of when or how the data distribution might shift.

To reduce the computational cost in the generation of diffusion models, several distillation-based approaches aim to decrease the number of generation steps[11, 12, 13]. These methods transfer knowledge from a pretrained teacher model to a diffusion model with fewer steps, effectively lowering computation for offline generative tasks. However, these approaches presume a fixed teacher and continuous access to original data. These assumptions are impractical in Online Continual Learning (OCL), where the teacher must adapt to new distributions without revisiting older data.

Building on the need to address both the computational overhead of diffusion-based generative replay and the constraints of Online Continual Learning (OCL), we pose a key question: How can we effectively distill knowledge into a continually evolving diffusion model while avoiding the stor-

age of past data and prohibitive generation costs? To tackle this challenge, we introduce Online Multi-Mode Adaptive Generative Distillation (MAGD), a framework specifically designed to support the continual training of diffusion models under online continual learning conditions. Our main contributions are as follows

- We introduce a novel strategy, **Noisy Intermediate Generative Distillation (NIGD)**, to distill knowledge using intermediate noisy images directly generated from the reverse process (not obtained by adding noise to the final image), thereby reducing redundant computation compared to full denoising steps.
- We propose **SNR-Guided Generative Distillation (SGGD)**, which uses an SNR-based threshold to dynamically select among current data, generated samples, or Gaussian noise, minimizing the frequency of full generation cycles and cutting computational costs.
- Empirical evaluations on Fashion-MNIST, CIFAR-10, and CIFAR-100 show that MAGD reduces overall generation steps (e.g., 10 for Fashion-MNIST, 25 for CIFAR) while preserving or even improving performance. Notably, it achieves a 25% reduction in computation compared to standard distillation and a 92% saving over methods using 1000-step denoising (DDGR-1000), all while producing higher-quality generated samples and strong classification performance.

2. RELATED WORK

Online Continual Learning (OCL): OCL handles data arriving in sequential, small batches, often without access to previous batches, introducing challenges such as new classes (Online Class Incremental, OCI) or variations like background shifts (Online Domain Incremental, ODI) [14, 15, 2, 16]. Traditional methods, using exemplars and contrastive learning loss, often struggle with privacy and storage limitations. To address these issues, researchers have utilized Generative Adversarial Networks (GANs) [17] and Variational Autoencoders (VAEs) [18] to synthesize past data, avoiding storage of actual data but facing challenges in maintaining image quality. Our approach leverages diffusion models to distill knowledge from generated noisy images, enhancing the sustainability of high-quality data generation. This method improves computational and memory efficiency in OCL, ensuring adaptability and robustness across varying conditions.

Diffusion Models in Continual Learning: diffusion models are renowned for their strong performance in various benchmarks [7], but they require substantial computational resources. Techniques such as DDIM [19], progressive distillation [11], and consistency models [12] have been developed to reduce these demands. Our research employs DDIM for its efficient conversion of noise to data, which is advantageous for continual learning applications. In continual learning, recent approaches [20, 21, 8] have used diffusion models to replace traditional replay buffers. For example, SDDR [10] uses a pretrained Stable Diffusion model as a static buffer, and DGR-distill[21] employs generative distillation for noise prediction instead of synthesized images. While these methods are innovative, they often overlook class balance and detailed knowledge from the teacher model's reverse diffusion process. Diffclass[9] provides optimal results by using separate models for each task, but this reduces efficiency. Additionally, current methods do not fully tackle the complexities of online continual learning, a critical area for practical application.

3. METHODOLOGY

3.1. Problem formulation

In our study, we explore online continual learning where a model learns from a data stream, presenting each batch only once. We define a data stream at time step k as $\mathcal{B}^k = \{(x_i, y_i)\}_{i=1}^{N_b}$, with each pair (x_i, y_i) representing an input data point and its label, and N_b represents the batch size. The noise prediction model is denoted by ϵ_θ. We define the online continual learning algorithm A as follows:

$$A^k : (\epsilon_{\theta^{k-1}}, \mathcal{B}^k) \rightarrow (\epsilon_{\theta^k}) \tag{1}$$

At each training step k, the model $\epsilon_{\theta^{k-1}}$ receives a small batch $\mathcal{B}^k$ and updates its parameters accordingly, resulting in the new model ϵ_{θ^k}.

3.2. Generative replay and Generative distillation

We consider both Generative Replay (DGR) and Generative Distillation (DGR-distill) [21] as our baselines, which are among the most commonly used strategies applying generative models to continual learning algorithms. Before training on a new batch $\mathcal{B}^k$, they first use the previous noise prediction model $\epsilon_{\theta^{k-1}}$ to generate a memory batch $\mathbf{X}_r$. We then add noise ϵ_r corresponding to the diffusion step t_r to obtain the noisy images $\mathbf{X}_r$. The two methods differ only in the target used for the distillation loss. In DGR, it uses the known added noise as the prediction target : $\mathcal{L}_{DGR} = MSE(\epsilon_r, \epsilon_{\theta^k}(\widetilde{\mathbf{X}}_r, t_r))$. However, in DGR-distill, it uses the previous model's output as target : $\mathcal{L}_{DGR-distill} = MSE(\epsilon_{\theta^{k-1}}(\widetilde{\mathbf{X}}_r, t_r), \epsilon_{\theta^k}(\widetilde{\mathbf{X}}_r, t_r))$

3.3. Noisy Intermediate Generative Distillation (NIGD)

To efficiently generate images, we utilize a DDIM scheduler [19], which operates over a selected subset of steps $\{\tau_1, \tau_2, \ldots, \tau_s\}$ from the total number of steps T, thereby reducing the number of necessary steps to S. As discussed

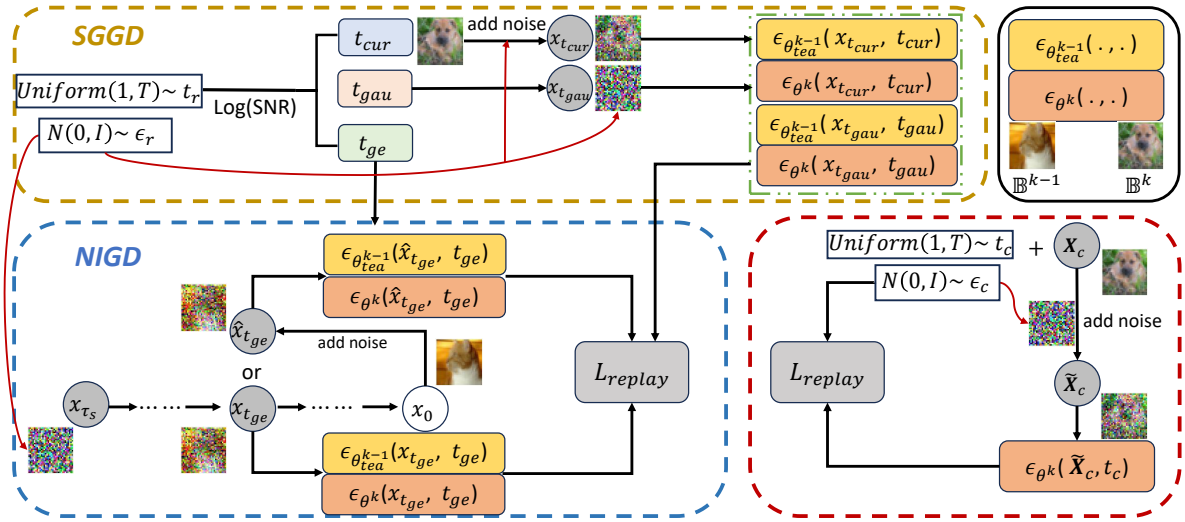

Fig. 1. Illustration of Our Method. The **yellow region** represents **SGGD**, the **blue region** denotes **NIGD**, and the **red region** corresponds to training on the current batch $\mathbb{B}^k$. $\epsilon_{\theta^{k-1}_{tea}}$ is our EMA-teacher, and ϵ_{θ^k} is our current model.

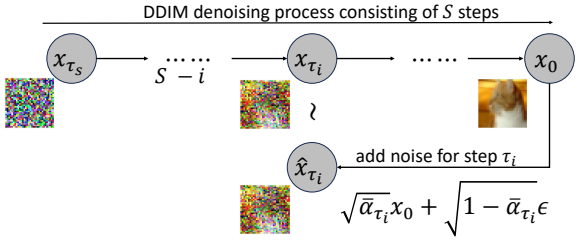

Fig. 2. An illustration of the DDIM denoising process with S steps shows two approaches: using $S-i$ steps to directly generate $\boldsymbol{x}_{\tau_i}$, or first generating the original images $\boldsymbol{x}_0$, followed by adding noise to produce the noisy image $\hat{\boldsymbol{x}}_{\tau_i}$ at step τ_i.

in Sec. 3.2, current continual learning methods that apply diffusion models use either DGR or DGR-distill. Both approaches initially generate the original images $\boldsymbol{x}_0$ using the full S steps, then add noise to produce $\hat{\boldsymbol{x}}_{\tau_i}$ at specified time steps τ_i, as illustrated in Fig. 2. However, if we utilize only $\hat{\boldsymbol{x}}_{\tau_i}$ for distillation during the generation process, the diffusion model can directly generate noisy images at time step denoted by $\boldsymbol{x}_{\tau_i}$. These noisy images can also be useful for distillation as they require only $S-i$ generation steps. We then compute the differences between them and explain their utility for distillation.

The reverse process in DDIM is described by:

$$\boldsymbol{x}_{\tau_i} = \sqrt{\bar{\alpha}_{\tau_i}} * \frac{\boldsymbol{x}_{\tau_{i+1}} - \sqrt{1-\bar{\alpha}_{\tau_{i+1}}}\epsilon_\theta(\boldsymbol{x}_{\tau_{i+1}})}{\sqrt{\bar{\alpha}_{\tau_{i+1}}}} + \sqrt{1-\bar{\alpha}_{\tau_i}}\epsilon_\theta(\boldsymbol{x}_{\tau_{i+1}}) \quad (2)$$

From this, the generated image $\boldsymbol{x}_0$ is obtained after S steps. The noisy images $\hat{\boldsymbol{x}}_{\tau_i}$ are derived from $\boldsymbol{x}_0$ as follows:

$$\hat{\boldsymbol{x}}_{\tau_i} = \sqrt{\bar{\alpha}_{\tau_i}}\boldsymbol{x}_0 + \sqrt{(1-\bar{\alpha}_{\tau_i})}\epsilon \quad (3)$$

We can then derive the difference between $\hat{\boldsymbol{x}}_{\tau_i}$ and $\boldsymbol{x}_{\tau_i}$ as shown in Fig. 2:

$$\hat{\boldsymbol{x}}_{\tau_i} - \boldsymbol{x}_{\tau_i} = \sum_{j=i}^{1}(\boldsymbol{r}_j\epsilon_\theta(\boldsymbol{x}_{\tau_j})) \quad (4)$$

$$\boldsymbol{r}_j = \sqrt{\bar{\alpha}_{\tau_i}}\left(\sqrt{\frac{1-\bar{\alpha}_{\tau_{j-1}}}{\bar{\alpha}_{\tau_{j-1}}}} - \sqrt{\frac{1-\bar{\alpha}_{\tau_j}}{\bar{\alpha}_{\tau_j}}}\right) \quad (5)$$

From Eq. (4), the difference between the noisy image $\hat{\boldsymbol{x}}_{\tau_i}$ (derived by adding noise to $\boldsymbol{x}_0$) and the directly generated noisy image $\boldsymbol{x}_{\tau_i}$ is determined by the generation steps from τ_i to τ_1. Based on our experience, ther residual component is relatively weaker compared to the noisy image $\boldsymbol{x}_{\tau_i}$.

In our continual learning setup, we use the previously trained model, $\epsilon_{\theta^{k-1}}$, as the teacher model to update the new model ϵ_{θ^k} by distilling knowledge for each τ_i. We require:

$$\epsilon_{\theta^k}(\boldsymbol{x}_{\tau_{i-1}}|\boldsymbol{x}_{\tau_i}, \tau_i) = \epsilon_{\theta^{k-1}}(\boldsymbol{x}_{\tau_{i-1}}|\boldsymbol{x}_{\tau_i}, \tau_i) \quad (6)$$

During the generation process, the model generates $\boldsymbol{x}_{\tau_i}$ without accessing $\hat{\boldsymbol{x}}_{\tau_i}$, yet $\boldsymbol{x}_{\tau_i}$ is essential for distilling localized information from the previous model. In a S-step DDIM generation, both the directly generated noisy image $\boldsymbol{x}_{\tau_i}$ and the noisy image $\hat{\boldsymbol{x}}_{\tau_i}$ created by adding noise to $\boldsymbol{x}_0$ are crucial for distillation. The former retains localized details, while the latter preserves global information. This study suggests two methods of obtaining a noised image, for any given diffusion step, τ_i as shown in Fig. 2:

1. **Two-Stage Approach**: Generate $\boldsymbol{x}_0$ using S steps, then add noise for step τ_i to get $\hat{\boldsymbol{x}}_{\tau_i}$.

2. **Direct Approach**: Directly generate $\boldsymbol{x}_{\tau_i}$ using $S - i$ steps.

In practice, we distill knowledge from both the intermediate noisy images and the two-stage noisy images produced during the inverse process.

3.4. SNR-Guided Generative Distillation (SGGD)

Research by [22] identifies two functional phases of diffusion models: initial denoising of corrupted images for refining final samples when t is small, and generating images from noise when t is large, demonstrating strong generalization across datasets like CIFAR-10 and CelebA in early diffusion stages ($t/T < 0.1$).

In continual learning, using solely generated images for training leads to image quality degradation [5, 4, 8]. We propose leveraging early-stage denoising capabilities of diffusion models for direct knowledge distillation from current training data, offering improved image clarity, knowledge preservation, and reduced computational costs by bypassing initial image generation.

To find the turning point t_c of the time step before which current training data can be effectively used, we calculate the Signal-to-Noise Ratio (SNR) along with the time step. This measurement assesses the relative amplitude of the added noise compared to the original image. We use the same formula as in [22]:

$$SNR(\boldsymbol{x}_0, t) = \frac{\bar{\alpha}_t \boldsymbol{x}_0^2}{1 - \bar{\alpha}_t} \tag{7}$$

We set $log(SNR) = 3$ as a threshold, as it maintains favorable FID scores. For Fashion-MNIST and CIFAR-10, the identified time steps, t_{low}, are 50 and 35, respectively. As $log(SNR)$ drops to -9, we reach a point where the model's input approximates Gaussian noise, marking the t_{high} thresholds at 878 and 848 for the respective datasets.

In our workflow, **images for distillation are selected based on the training step** t_r: 1. If $t_r < t_{low}$, use images from the **current batch**. 2. If $t_r > t_{high}$, use images generated from **Gaussian noise**. 3.Otherwise, **generate** noisy images from the previous model.

We adaptively adjust t_{low} and t_{high} using a moving average formula, minimizing manual tuning and reducing the need for generated images by about 20% without affecting performance.

3.5. EMA in Online Continual Learning

In an Online Continual Learning scenario, where no clear task boundaries exist as in classic class-incremental learning, we must update our teacher model $\epsilon_{\theta_{tea}}$ dynamically. We use the Exponential Moving Average (EMA) method [23, 24] to achieve this:

$$\epsilon_{\theta^k_{tea}} = (1 - \lambda)\epsilon_{\theta^{k-1}_{tea}} + \lambda \epsilon_{\theta^k} \tag{8}$$

Here, ϵ_{θ^k} represents our current model, updated with the latest batch k. We set the update rate λ to 0.01, allowing the teacher model to slowly assimilate new knowledge while ensuring stability over time.

3.6. Workflow and overall objective

Our method's workflow is illustrated in Fig. 1. At the training step k, we first initialize the current model ϵ_{θ^k} using the parameters from the previous model $\epsilon_{\theta^{k-1}}$. We then update ϵ_{θ^k} by processing the current batch B^k and distilling knowledge from the EMA-based teacher $\epsilon_{\theta^{k-1}_{tea}}$. Because the previous batch B^{k-1} is not retained, it is unavailable for training at this step. The current batch B^k comprises images ($\mathbf{X}_c$), labels ($\mathbf{Y}_c$).

The process starts by randomly selecting time steps t_r, and generating Gaussian noise ϵ_r. According to the resampling guidelines in **SNR-Guided Generative Distillation (SGGD)** Sec. 3.4, we categorize t_r into three types: t_{cur} for replaying current images $\boldsymbol{x}_{t_{cur}}$, t_{gau} for replaying Gaussian noise $\boldsymbol{x}_{t_{gau}}$, and t_{ge} for replaying generated noisy images. For instances categorized under t_{ge}, we employ **Noisy Intermediate Generative Distillation (NIGD)** to produce half of the images as directly noisy images $\boldsymbol{x}_{t_{ge}}$ and the other half as two-stage noisy images $\hat{\boldsymbol{x}}_{t_{ge}}$. By combining all types of noisy images for replay, we can construct our noisy memory batch as $\widetilde{\mathbf{X}}_r$. The replay loss is then calculated based on these categorizations.

$$\mathcal{L}_{replay} = MSE(\epsilon_{\theta^k}(\widetilde{\mathbf{X}}_r, t_r), \epsilon_{\theta^{k-1}_{tea}}(\widetilde{\mathbf{X}}_r, t_r)) \tag{9}$$

Next, we sample time steps t_c and noise ϵ_c. We then pass the current noisy training data ($\widetilde{\mathbf{X}}_c, \epsilon_c$) through our current model to obtain:

$$\mathcal{L}_{current} = MSE(\epsilon_{\theta^k}(\mathbf{X}_c, t_c), \epsilon_c) \tag{10}$$

Finally, the overall objective is formulated as:

$$\mathcal{L}_{total} = \alpha \mathcal{L}_{current} + (1 - \alpha)\mathcal{L}_{replay} \tag{11}$$

where α is a hyperparameter controls the balance between the current loss and the replay loss. After updating our current model ϵ_{θ^k}, we then use Eq. (8) to update our EMA-based teacher.

4. EXPERIMENTS AND RESULTS

In this paper, we evaluate our method using three popular datasets for online continual learning: Fashion-MNIST, CIFAR-10, and CIFAR-100. For each dataset, we train our

model offline on half of the classes as the initial task, establishing a well-trained baseline. The remaining dataset is then introduced in an online stream.

4.1. Evaluation metrics and Methods

We use the Fréchet Inception Distance (**FID**) to assess image quality against a test set from previously encountered tasks and the Kullback-Leibler Divergence (**KLD**) to evaluate the class distribution balance in generated images. Final classification accuracy (**Acc**) measures the performance of class-conditioned diffusion models. We explore both unconditional and class-conditioned diffusion models, referencing [6, 25]. For the unconditional model, we compare our approach with deep generative methods like **DGR**, **DGR with distillation** [21], and **DDGR** [8], and the memory-based method **ER** [14]. Comparisons include **Fine-tuning (F.T.)** as a lower bound and **Joint-training (J.T.)** as an upper bound, along with **DDGR-1000** for its high computational performance.

For the class-conditioned model, we assess classification accuracy against memory-free methods like **BIR**[26] and **PASS**[27], and the memory-based method **PCR**[28].

Fashion-MNIST employs a small UNet for 10 DDIM steps, while **CIFAR-10** and **CIFAR-100** use a medium-sized UNet for 25 steps. Both **ER** and **PCR** utilize a memory buffer of 1000. All diffusion models implement EMA.

4.2. Overall results

We present results for both unconditional and class-conditioned diffusion models across Tab. 1 and Tab. 2, summarized as mean and standard deviation over five random runs.

Unconditional Diffusion Model Results: In Tab. 1, Our method outperforms DGR-distill, showing improvements of 4.5 to 5.0 in FID scores and superior KLD performance, with a 25% reduction in computational costs. For Fashion-MNIST, it matches DDGR-1000's performance with just 10 generation steps and only 25 steps for CIFAR-10, reducing computational demand by 92% compared to DDGR-1000.

Class-Conditioned Diffusion Model Results: In Tab. 2, when examining the class-conditioned diffusion model, our method does not just compete on FID scores but also demonstrates a clear advantage in classification accuracy. It consistently surpasses the performance of basic DGR-distill and closely approaches, and in some metrics exceeds, that of DDGR-1000. Remarkably, our approach outperforms the memory-based method PCR, even with a significantly larger memory buffer.

5. ABLATION STUDY

Our method incorporates two innovative components: NIGD and SGGD. SGGD utilizes three types of replay images: Gaussian noise, current images, and generative images.

Table 1. Results Presented as Mean and Standard Deviation Over 5 Random Runs, with unconditional diffusion model.

	Fashion-MNIST			CIFAR-10		
	FID↓	KLD↓	Time↓	FID↓	KLD↓	Time↓
F.T.	95.5 ± 10.2	4.75 ±1.81	×0.15	73.5±5.8	3.83±1.15	×0.08
DDGR-1000	19.2 ± 2.5	0.09 ± 0.01	×20.5	37.8 ± 3.4	0.15 ± 0.02	×8.75
J.T.	14.7 ± 1.5	0.07 ± 0.01	×0.15	27.3±2.1	0.11±0.01	×0.08
ER	25.9±3.9	0.38±0.15	×0.15	50.5±5.9	0.35±0.13	×0.08
DGR	90.5 ± 10.5	1.15 ±0.23	×0.91	75.3 ± 6.6	1.55 ± 0.58	×0.95
DGR-distill	24.8 ± 3.4	0.17 ± 0.08	0.8h ×1	46.3 ± 6.0	0.28 ± 0.14	1.5h ×1
Ours	20.3 ± 2.2	0.10 ± 0.04	×0.75	41.3 ± 4.6	0.17 ± 0.09	×0.72

Table 2. Results Presented as Mean and Standard Deviation Over 5 Random Runs, with class-conditioned diffusion model.

	CIFAR-10			CIFAR-100		
	FID↓	Acc↑	Time↓	FID↓	Acc↑	Time↓
F.T.	58.5±7.8	11.2 ± 0.2	× 0.07	65.2 ± 8.9	4.5 ±0.3	× 0.07
DDGR-1000	31.5 ± 1.6	45.7 ± 1.2	× 9.21	35.8 ± 2.3	28.8 ± 0.9	× 9.21
J.T.	26.3±1.8	73.5 ± 0.5	× 0.07	30.5 ±2.2	67.4 ± 0.3	× 0.07
ER	42.5±5.1	31.8 ± 2.5	× 0.07	52.6±5.4	18.9 ± 3.1	× 0.07
DGR-distill	39.3 ± 4.2	38.8 ± 3.8	1.7h × 1	44.5 ± 4.5	24.3 ± 2.4	1.7h × 1
BIR	-	30.5 ± 3.7	-	-	16.3 ± 3.8	-
PASS	-	37.8 ± 2.5	-	-	20.5 ± 2.7	-
PCR	-	40.5 ± 3.2	-	-	25.7 ± 2.2	-
Ours	34.7 ± 3.5	42.1 ± 2.1	× 0.71	38.2 ± 3.7	27.5 ± 1.3	× 0.71

NIGD employs two methods to generate noisy images from the diffusion model: a two-stage approach and a direct approach. We use the DGR-distill backbone with unconditional diffusion models and apply 10 generation steps for Fashion-MNIST. Our results demonstrate that both SGGD and NIGD enhance the quality of the generated images while reducing computational costs.

Table 3. Ablation Study on Fashion-MNIST

	FID↓	KLD↓	Time↓
w. Generative (two-stage)	24.8 ± 3.4	0.17 ±0.08	0.8h × 1
w. Current	56.9 ± 3.8	0.33 ± 0.12	×0.25
w. Gaussian	35.4 ± 2.6	0.76 ± 0.09	×0.25
w. SGGD	22.4±3.8	0.15±0.08	×0.85
w. Generative (direct)	23.8 ± 5.8	0.15 ± 0.09	×0.63
w. NIGD	21.5 ± 2.7	0.13 ±0.05	×0.83
Ours	20.1 ± 2.2	0.10 ± 0.03	×0.75

6. CONCLUSION

We introduced the Multi-Mode Adaptive Generative Distillation (MAGD) approach to mitigate catastrophic forgetting and reduce computational costs in online continuous training of diffusion models. By integrating NIGD, SGGD, and EMA, our method maintains high-quality image generation while reducing computational expenses by up to 25% compared to basic DGR-distill and 92% compared to DDGR-1000. In class-conditioned models, MAGD significantly outperforms basic DGR-distill and surpasses memory-based methods in terms of classification accuracy, demonstrating its potential as a viable alternative to traditional memory buffers.

7. REFERENCES

[1] James K., R. Pascanu, Neil R., Joel V., G. Desjardins, Andrei A. Rusu, K. Milan, J. Quan, T. Ramalho, Agnieszka G., D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[2] J. Wei, Y.and Ye, Z. Huang, J. Zhang, and H. Shan, "Online prototype learning for online continual learning," in *ICCV*, 2023, pp. 18764–18774.

[3] Guo Y., L. Bing, and Zhao D., "Online continual learning through mutual information maximization," in *ICML*. 2022, vol. 162 of *Proceedings of Machine Learning Research*, pp. 8109–8126, PMLR.

[4] T. Lesort, H. Caselles-Dupré, M. G.Ortiz, A. Stoian, and D.Filliat, "Generative models from the perspective of continual learning," *IJCNN*, pp. 1–8, 2018.

[5] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," in *Neur IPS*, 2017.

[6] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *NeurIPS*, 2020.

[7] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *NeurIPS*, vol. abs/2105.05233, 2021.

[8] R. Gao and W. Liu, "Ddgr: Continual learning with deep diffusion-based generative replay," in *ICML*, 2023.

[9] Z. Meng, J. Zhang, C. Yang, Z. Zhan, P. Zhao, and Y. Wang, "Diffclass: Diffusion-based class incremental learning," *ECCV*, vol. abs/2403.05016, 2024.

[10] Q. Jodelet, Xin Liu, Y. Phua, and T. Murata, "Class-incremental learning using diffusion model for distillation and replay," *ICCVW*, pp. 3417–3425, 2023.

[11] Tim S. and J. Ho, "Progressive distillation for fast sampling of diffusion models," in *ICLR*. 2022, OpenReview.net.

[12] Yang S., Prafulla D., Mark Chen, and Ilya S., "Consistency models," *ICML*, 2023.

[13] David B., A. Autef, J. Lin, Dian A., S. Zhai, S. Hu, D. Zheng, W. Talbot, and E. Gu, "Tract: Denoising diffusion models with transitive closure time-distillation," *ICML*, vol. abs/2303.04248, 2023.

[14] A. Chaudhry, M. Rohrbach, M. Elhoseiny, T. Ajanthan, P. K. Dokania, Philip H. S. Torr, and M. Ranzato, "Continual learning with tiny episodic memories," *ICML*, vol. abs/1902.10486, 2019.

[15] M. De Lange and Tinne T., "Continual prototype evolution: Learning online from non-stationary data streams," *ICCV*, pp. 8230–8239, 2020.

[16] Z. Mai, R. Li, H. J. Kim, and S. Sanner, "Supervised contrastive replay: Revisiting the nearest class mean classifier in online class-incremental continual learning," *CVPRW*, pp. 3584–3594, 2021.

[17] Ian J. Goodfellow, Jean P., Mehdi M., B. Xu, David W., S. Ozair, Aaron C. C., and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, pp. 139 – 144, 2014.

[18] D. P. Kingma and Max W., "Auto-encoding variational bayes," *CoRR*, vol. abs/1312.6114, 2013.

[19] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," *ICLR*, vol. abs/2010.02502, 2020.

[20] M. Zajac, K. Deja, A. Kuzina, J. M. Tomczak, T. Trzciński, Florian Shkurti, and Piotr Miłoś, "Exploring continual learning of diffusion models," *arxiv*, 2023.

[21] S. Masip, Pau R., T. Tuytelaars, and Gido M. van de Ven, "Continual learning of diffusion models with generative distillation," *arXiv preprint arXiv:2311.14028*, 2023.

[22] Kamil Deja, A. Kuzina, T. Trzciński, and J. M. Tomczak, "On analyzing generative and denoising capabilities of diffusion-based deep generative models," in *Neurips*, 2022.

[23] J. Grill, F. Strub, F. Altch'e, C.Tallec, P. H. Richemond, Elena B., Carl D., B. Pires, Z. Guo, M. Azar, Bilal Piot, Koray K., R. Munos, and M. Valko, "Bootstrap your own latent: A new approach to self-supervised learning," *NeurIPS*, vol. abs/2006.07733, 2020.

[24] N. Michel, M. Wang, L. Xiao, and T. Yamasaki, "Rethinking momentum knowledge distillation in online continual learning," *ICML*, 2024.

[25] Olaf R., P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *MICCAI*, vol. abs/1505.04597, 2015.

[26] Gido M. van de Ven, Hava T. Siegelmann, and A. S. Tolias, "Brain-inspired replay for continual learning with artificial neural networks," *Nature Communications*, vol. 11, 2020.

[27] Fei Zhu, Xu-Yao Z., Chuang W., Fei Y., and Cheng-Lin L., "Prototype augmentation and self-supervision for incremental learning," in *CVPR*, 2021, pp. 5867–5876.

[28] H. Lin, B. Zhang, S. Feng, X. Li, and Y. Ye, "Pcr: Proxy-based contrastive replay for online class-incremental continual learning," *CVPR*, pp. 24246–24255, 2023.