ViTReplay: Continual Learning with Vision Transformers via Conditional GAN Replay

Zhicong Zhu[1,*], Kun Zhou[2,*], Bo Tang[1]
[1]Worcester Polytechnic Institute, Worcester, MA, USA
[2]Linyi University, Linyi, China

***Abstract*—Continual learning aims to learn a stream of tasks while mitigating catastrophic forgetting. With the rapid progress of Vision Transformers (ViTs), they have achieved strong performance on a wide range of vision tasks and have attracted increasing attention in continual learning. However, most ViT-based continual learning approaches rely on exemplar buffers for rehearsal, and representation diversity often degrades when the buffer is small or unrepresentative; replacing exemplars with generative replay can reduce storage, but low-fidelity synthesis may introduce distribution shift. To address this, we propose a ViT-based continual learning framework that integrates class-conditional generative adversarial networks (cGANs) with ViTs and employs bidirectional alignment, improving both diversity and alignment of distributions and semantics. We define forward alignment as using replay samples synthesized by a cGAN from previous tasks to regularize the ViT, thereby mitigating representation drift and promoting sample diversity. Conversely, we define reverse alignment as using the updated ViT to provide semantic guidance to the cGAN by enforcing classification consistency together with feature and prediction-level alignment, thereby improving the semantic fidelity of generated samples. Experimental results indicate that, regardless of whether original samples are stored, this approach significantly enhances model robustness and representation diversity in multi-task environments, thus achieving more efficient and effective continual learning.**

***Index Terms*—Conditional Generative Adversarial Network, Continual Learning, Vision Transformer, Bidirectional alignment, Generative replay**

I. Introduction

In traditional deep learning paradigms, models are typically trained once on a fixed dataset and remain static thereafter. When encountering new tasks or shifts in data distributions [1], they often suffer from catastrophic forgetting, necessitating costly retraining. Continual learning or lifelong learning aims to address this limitation by enabling models to continuously integrate new knowledge while preserving previously acquired information [2]. Meanwhile, as Vision Transformers (ViTs) have become increasingly prominent in vision tasks, ViT-based training strategies have also been introduced into continual learning frameworks [3]. However, for such ViT-based approaches, handling historical data is an unavoidable challenge: relying on stored exemplars raises privacy and memory-budget concerns [4], and when the buffer is small or unrepresentative, it may further degrade generalization and reduce feature/representation diversity.

Recent work, such as DyTox [5], has demonstrated the potential of ViT-based continual learning by integrating task tokens and knowledge distillation [6], achieving promising results in practical scenarios. DyTox relies on a relatively simple distillation procedure [7] and, in its original design, still requires storing historical samples for rehearsal. Such a "limited-buffer + simplified distillation" setting may fall short of capturing the complex dynamics induced by continuously evolving tasks; consequently, it can still suffer from representation drift and reduced representation diversity, especially under strict privacy or memory budgets and long task sequences. As an alternative, generative replay introduces generative models (e.g., GANs [8]) to approximate old-task data, thereby reducing the dependence on raw exemplar storage and partially alleviating representation drift and diversity degradation. However, naively replacing stored exemplars with replayed synthetic data is non-trivial: since the fidelity and distributional consistency of synthesized samples are difficult to guarantee, it remains challenging to achieve robust feature alignment and to maintain a rich representation space as tasks evolve.

To address these challenges, we propose a continual learning framework that integrates class-conditional generative adversarial networks (cGANs) [9] with Vision Transformers (ViTs) [10] [11] via a bidirectional alignment mechanism. Unlike traditional approaches that rely solely on storing raw training samples, our method assigns a dedicated cGAN to each newly introduced task. By imposing complementary constraints, including classification consistency, feature-level alignment, and diversity regularization, these constraints encourage synthesized samples to be both representative and diverse, preventing mode collapse and homogeneous representations, while providing stable replay signals that mitigate discriminative representation drift and improve generalization of the ViT. Conversely, the ViT's strong feature extraction capability offers semantic and classification guidance to the cGAN, thereby improving the semantic fidelity and overall quality of generated samples. By establishing a bidirectional coupling between the cGAN and the ViT, our framework aligns the generative distribution with the discriminative semantics throughout task evolution, leveraging the complementary strengths of both components. Moreover, to accommodate diverse deployment requirements, our method can optionally retain a small subset of historical samples for higher accuracy; when storage is infeasible or privacy constraints prohibit sample retention, relying solely on cGAN-synthesized replay still achieves competitive performance. Overall, this design fully leverages the ViT's feature extraction capabilities, ensur-

*Zhicong Zhu and Kun Zhou contributed equally to this work.

ing the seamless integration of previously learned knowledge and newly acquired information into a unified, continuously evolving representation space.

II. Background

In continual or incremental learning, deep models are required to adapt to novel tasks or classes that arrive sequentially, while mitigating catastrophic forgetting of previously acquired knowledge [12] [13]. Some large-scale continual learning strategies rely on rehearsal, where a fixed-size memory buffer stores a subset of past tasks' samples to be replayed alongside new data during subsequent training stages [14]. Some methods compress or prune these buffers to reduce storage costs [15]. Additionally, an increasing number of studies leverage Generative Adversarial Networks (GANs) [8] in incremental learning to enable generative replay by synthesizing data resembling previously learned tasks. However, without appropriate alignment to the current model, the generator may itself suffer from forgetting or mode collapse over multiple stages of training.

Following the original introduction of the Transformer [16] in machine translation, its application has expanded into computer vision. Specifically, the Vision Transformer (ViT) [17] segments an image into a series of patch tokens and employs self-attention mechanisms to achieve superior performance on various vision benchmarks. In incremental learning, earlier research predominantly employed convolutional neural networks (CNNs), applying rehearsal or regularization-based approaches to alleviate forgetting. More recently, researchers have begun exploring the use of Transformers for continual learning [18]. For instance, an incremental learning method called LVT [19] incorporates inter-task attention mechanisms and dual-classifier structures to balance the retention of old knowledge with the plasticity for new knowledge, thereby mitigating catastrophic forgetting. Another example is DyTox [5]—a dynamic and expandable Transformer framework specifically designed for incremental learning scenarios. Its core idea is to insert a lightweight Task-Attention Block at the end of a ViT and add a vectorized Task Token for each new task. When new tasks or classes arrive, DyTox expands the model parameters merely by adding the corresponding task token, thereby avoiding a large increase in the overall parameter size and significantly alleviating catastrophic forgetting.

While these approaches effectively mitigate catastrophic forgetting to some extent, challenges remain in terms of privacy, storage costs, and adaptation to large-scale tasks. To address these issues, we build upon the DyTox architecture by introducing a novel solution that combines conditional GANs (cGANs) [9] with dual-directional alignment. In contrast to traditional methods, our bidirectional collaboration approach not only accommodates memory-sensitive scenarios by enabling high-quality replay and superior adaptability to new classes without storing original data, but also allows the storage of raw samples from previous tasks for enhanced accuracy in continual learning. Furthermore, the diversity of the generated samples strengthens class discriminability, providing a more efficient solution for large-scale incremental learning with lengthy task sequences.

III. Methodology

In this section, we present our newly proposed framework for continual learning. The specific framework is shown in the Figure 1. We first review the foundations of the DyTox [5] framework and Task Tokens and then introduce our dual-directional alignment approach, which establishes a closed-loop collaboration between the ViT and class-conditional GANs.

A. Training Procedure

In our incremental task-learning scenario, the model transitions from task $t-1$ to task t, striving to assimilate new knowledge from the incoming task while preserving recognition of previously acquired information. The detailed procedure is shown in Algorithm 1. We consider the setting where no real samples from previous tasks are stored; replay samples are fully synthesized. Real data from the current task $\mathcal{D}^t$ are still used for learning task t. If a subset of real samples is stored, one can combine them with the generated samples to further enhance performance. First, we align the frozen model f_θ^{t-1} with the current ViT f_θ^t by leveraging replay samples synthesized by the generators stored in memory $\mathcal{G}^{(t-1)}$. At task t, we leverage all previously learned generators in memory, $\mathcal{G}^{(t-1)} = \{G^1, \ldots, G^{t-1}\}$, to synthesize replay samples. For each $\tau \in \{1, \ldots, t-1\}$, we sample $z \sim p_z$ and $c \sim \mathrm{Uniform}(\mathcal{C}_\tau)$, and generate $x_{\mathrm{fake}}^\tau = G^\tau(z, c)$. In each rehearsal mini-batch, we generate class-balanced replay (i.e., the same number of samples per class). Second, we align from the f_θ^t back to the cGAN (G^t, D^t), aligning the generator's synthetic data with the evolving feature space of f_θ^t. This cyclical process forms a closed-loop wherein the ViT benefits from synthesized old-class samples for knowledge distillation, and the cGAN is refined by the ViT's classification feedback and feature matching. Specifically, as new tasks arrive sequentially, we alternate between two principal steps:

1) **Stage I (Forward alignment):** Leveraging replay samples $\{x_{\mathrm{fake}}^\tau\}_{\tau=1}^{t-1}$ synthesized by all generators in memory $\mathcal{G}^{(t-1)}$, we minimize the training loss for the new task t as well as the knowledge distillation loss for previously learned tasks. Through this approach, the model integrates the newly introduced task while preserving recognition performance on earlier tasks.
2) **Stage II (Reverse alignment):** We initialize and optimize the generator and discriminator losses—including adversarial, classification consistency, diversity, and feature matching objectives—to ensure that the generated samples faithfully reflect the distribution of the current task t, exhibit sufficient diversity, and align with the ViT's decision boundaries.

Accordingly, for each task t, we complete a single closed-loop update consisting of one forward alignment step to update the ViT and one reverse alignment step to update the cGAN. The resulting system seamlessly balances the learning of novel

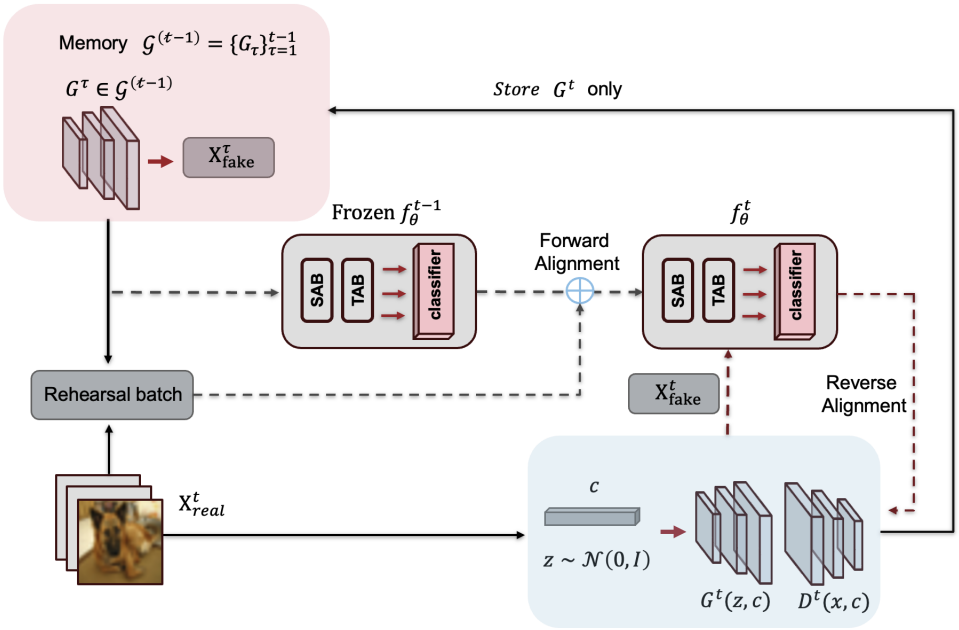

Fig. 1. Illustration of the training pipeline at incremental step t. The memory stores a set of previously learned generators $\mathcal{G}^{(t-1)}$, which synthesize replay samples x^τ_{fake} ($\tau \leq t-1$). These generated samples are combined with the current real data x^t_{real} to form rehearsal batches for updating the student ViT f^t_θ. *Forward alignment (gray dashed arrows)* from the frozen teacher f^{t-1}_θ constrains the student f^t_θ to preserve previously acquired knowledge. Subsequently, the updated ViT f^t_θ performs *reverse alignment* (red dashed arrows) to guide the training of the class-conditional GAN (G^t, D^t) based on the samples synthesized by the generator $G^t(z, c)$. Finally, only the generator G^t is stored in memory for future tasks.

information and the preservation of knowledge from previous tasks, providing an efficient and scalable solution.

B. Overview of DyTox

DyTox [5] first employs a Vision Transformer (ViT) [10] to segment the input image into multiple patch tokens, which are subsequently passed through a series of Self-Attention Blocks (SABs) to extract shared visual features. In the final Task-Attention Block (TAB), the corresponding Task Token is concatenated or fused with these generic features, yielding a task-specific representation that is then fed into its dedicated classifier (i.e., a task-specific head) for prediction. As new tasks arrive, DyTox only requires adding a new Task Token and an associated independent classifier for each newly introduced task, while knowledge distillation techniques are used to preserve recognition of previously learned classes.

When learning task t, DyTox allocates a newly created Task Token u_t and updates it via the classification loss L_{clf} in order to acquire the discriminatory ability for new classes. Meanwhile, to mitigate forgetting of old classes, DyTox retains a frozen snapshot f^{t-1}_θ from the end of the previous task and stores a subset of past training samples for knowledge distillation. Consequently, during the training phase of task t, a knowledge distillation term L_{kd} is applied to these buffered samples to preserve prior knowledge. Moreover, to ensure that the Task Tokens across different tasks remain disentangled, DyTox incorporates a diversity loss L_{div}, which enforces a certain distance among these tokens in the latent space, thereby alleviating task-to-task interference. The overall DyTox loss function is formulated as

$$L_{\text{ViT}} = (1-\alpha)\, L_{\text{clf}} + \alpha\, L^{\text{ViT}}_{\text{kd}} + \lambda\, L_{\text{div}} \tag{1}$$

In this formulation, $(1-\alpha)\, L_{\text{clf}}$ targets the newly introduced classes $\mathcal{C}_t$, enforcing the model to focus on learning them. Meanwhile, $\lambda\, L_{\text{div}}$ helps regulate task-token diversity, preventing the parameters allocated for each task from collapsing into excessively similar regions. This approach effectively alleviates catastrophic forgetting. Building upon the ViT architecture and Task Token mechanism of DyTox, we further propose a generative replay strategy and a bidirectional alignment scheme. Note that having one classifier head per task does not imply task-incremental evaluation. We follow the standard class-incremental protocol: the task identity is not provided at test time.

C. Dual-Directional Knowledge Distillation

Update ViT using cGANs: In the original DyTox framework, the knowledge distillation loss $L^{\text{ViT}}_{\text{kd}}$ primarily relies on stored samples from previous tasks. Specifically, a subset of historical data is fed into both the current model and the frozen snapshot to compute their output distributions; the Kullback–Leibler (KL) [20] divergence between these distributions is then minimized. Moreover, $L^{\text{ViT}}_{\text{kd}}$ is typically combined with

Algorithm 1 ViTReplay Algorithm

Input: Data streams $\{\mathcal{D}^t\}_{t=1}^{T}$, $\Lambda_{\mathrm{ViT}} = (\alpha, \lambda)$, $\Lambda_{\mathrm{G}} = (\lambda_{\mathrm{cls}}, \lambda_{\mathrm{div}}, \mu)$
For a new task t: $\mathcal{D}^t$
if $t = 1$ **then**
  Initialize generator memory: $\mathcal{G}^{(0)} \leftarrow \emptyset$
  Train ViT for the first time:
    $f_\theta^t \leftarrow (\mathcal{D}^t, \Lambda_{\mathrm{ViT}})$
else
  Generate replay samples from memory $\mathcal{G}^{(t-1)}$:
    For all $G^\tau \in \mathcal{G}^{(t-1)}$
    $x_{\mathrm{fake}}^\tau \leftarrow G^\tau(\cdot)$
    $x_{\mathrm{real}}^t \leftarrow \mathrm{sampleReal}(\mathcal{D}^t)$
  Forward alignment to update f_θ^t:
    $\mathrm{rehearsalBatch} \leftarrow (x_{\mathrm{real}}^t, x_{\mathrm{fake}}^\tau)$
    $f_\theta^t \leftarrow (f_\theta^{t-1}, \mathrm{rehearsalBatch}, \Lambda_{\mathrm{ViT}})$
end if
Train the cGAN for current task t:
  $(G^t, D^t) \leftarrow \mathrm{initCGAN}()$
Reverse alignment to update (G^t, D^t):
repeat
  update D^t by minimizing L_{D^t} (Eq. (3))
  update G^t by minimizing L_{G^t} (Eq. (9))
until converged
store $\mathcal{G}^{(t)} \leftarrow \mathcal{G}^{(t-1)} \cup \{G^t\}$
Freeze f_θ^t for the next task

Output: f_θ^t, $\mathcal{G}^{(t)}$

the classification loss L_{clf} and the diversity regularization L_{div} to form the full loss function (as shown in Eq. 1).

For the ViTReplay framework, we propose leveraging class-conditional GANs (cGANs) to synthesize replay samples for rehearsing previously learned tasks. Upon the arrival of task t, the model has access to a frozen snapshot f_θ^{t-1} from task $t-1$ and a set of previously learned generators stored in memory, denoted as $\mathcal{G}^{(t-1)}$. Specifically, we used all $G^\tau \in \mathcal{G}^{(t-1)}$ with $\tau \le t-1$ to synthesize replay samples $x_{\mathrm{fake}}^\tau = G^\tau(z, c)$. Let $p^{t-1}(\cdot \mid x) = \mathrm{softmax}\big(f_\theta^{t-1}(x)\big)$. In class-incremental learning, we distill only on old classes. Let $\mathcal{C}_{1:t-1} = \bigcup_{k=1}^{t-1} \mathcal{C}_k$ denote the old classes. Here $f_\theta^t(x)$ denotes the unified logit vector over $\mathcal{C}_{1:t} = \bigcup_{k=1}^{t} \mathcal{C}_k$, obtained by concatenating the logits from all task-specific heads learned up to step t, and define $p_{\mathrm{old}}^t(\cdot \mid x) = \mathrm{softmax}\Big(f_\theta^t(x)\big|_{\mathcal{C}_{1:t-1}}\Big)$. To preserve knowledge of previously learned classes, we minimize:

$$L_{\mathrm{kd}}^{\mathrm{ViT}} = \mathbb{E}_{x=x_{\mathrm{fake}}^\tau}\Big[D_{\mathrm{KL}}\big(p^{t-1}(\cdot \mid x) \,\big\|\, p_{\mathrm{old}}^t(\cdot \mid x)\big)\Big]. \quad (2)$$

where $D_{\mathrm{KL}}(\cdot\|\cdot)$ denotes the KL divergence. Intuitively, $D_{\mathrm{KL}}\big(p^{t-1}(\cdot \mid x) \,\big\|\, p_{\mathrm{old}}^t(\cdot \mid x)\big)$ measures the discrepancy between the teacher distribution $p^{t-1}(\cdot \mid x)$ and the student distribution $p_{\mathrm{old}}^t(\cdot \mid x)$ on the same replay sample x. Minimizing this term encourages f_θ^t to preserve the predictive behavior of f_θ^{t-1} on previously learned classes.

In our newly proposed algorithm, we still adhere to the overall structure of Eq. 1. However, by leveraging the old-class data generated by the cGAN, we can align the class distributions between the old and new models. This allows knowledge distillation to be carried out even under data-scarce conditions or when historical samples cannot be retained. Additionally, the redesigned cGAN enhances a certain degree of data diversity. And it is also possible to store the original data and combine these stored samples with newly generated ones for training, which would substantially improve performance.

Update cGANs using ViT: In conventional ViT-based replay strategies, the model absorbs prior knowledge only through a one-way pathway. In contrast, our bidirectional alignment introduces an additional reverse pathway, where the updated ViT f_θ^t further supervises the training of the task-specific cGAN (G^t, D^t) [9]. Concretely, as the representation space of f_θ^t evolves over incremental tasks, independently optimizing f_θ^t and the cGAN can easily lead to distributional drift, thereby reducing the compatibility between the synthesized samples and the decision boundaries of f_θ^t and ultimately degrading performance. We therefore propose this second alignment path to ensure that the generated samples remain aligned with the latest decision boundaries and to maximize effectiveness. To this end, the ViT [10] provides not only classification-level feedback (to enforce correct labeling of generated samples) but also feature-level alignment signals, enabling the generator to adapt to the updated discriminative criteria in higher-level representations. This design mitigates mode collapse and distributional drift that may otherwise occur when transitioning to new tasks.

In summary, this dual-directional mechanism introduces a closed-loop interplay between the current ViT and the cGAN. On one hand, the ViT relies on the cGAN's synthetic old data for knowledge distillation, reinforcing previously learned knowledge. On the other hand, the ViT's evolving decision boundaries and feature representations continuously refine the generator, ensuring it produces samples that conform to the latest model expectations. The detailed cGAN implementation appears in the subsequent equations.

D. Class-Conditional GANs and Loss Functions

We associate each task t with a class-conditional GAN (G^t, D^t). The generator G^t takes (z, c) as inputs, where $z \sim p_z$ is a noise vector, c denotes the class label, which corresponds to the specific category information in the current classification task. Let $\mathcal{C}_t$ denote the class set of task t, and let $p_{\mathrm{data}}^t(c)$ denote the class prior over $\mathcal{C}_t$ (we use the empirical prior or a uniform distribution over $\mathcal{C}_t$).

(a) Adversarial loss. Let $(x, c) \sim p_{\mathrm{data}}^t(x, c)$ be a real image-label pair from the current task t. The discriminator D^t aims to distinguish real from synthetic data:

$$\begin{aligned} L_{D^t} = &- \mathbb{E}_{(x,c)\sim p_{\mathrm{data}}^t}\big[\log D^t(x, c)\big] \\ &- \mathbb{E}_{z\sim p_z,\, c\sim p_{\mathrm{data}}^t(c)}\Big[\log\big(1 - D^t(G^t(z, c), c)\big)\Big]. \end{aligned} \quad (3)$$

The generator's adversarial loss tries to fool D^t:

$$L_{\mathrm{GAN}}^{G^t} = -\,\mathbb{E}_{z\sim p_z,\; c\sim p_{\mathrm{data}}^t(c)}\Big[\log D^t(G^t(z,c),c)\Big]. \quad (4)$$

(b) Classification consistency. Let $p^t(\cdot \mid x) = \mathrm{softmax}(f_\theta^t(x))$, and let $p^t(c \mid x)$ denote the probability assigned to class c by this distribution. To ensure that synthesized samples of class c remain recognizable to the current ViT f_θ^t, we introduce:

$$L_{\mathrm{cls}} = \mathbb{E}_{z\sim p_z,\; c\sim p_{\mathrm{data}}^t(c)}\Big[-\log p^t\big(c \,\big|\, G^t(z,c)\big)\Big], \quad (5)$$

Minimizing L_{cls} aligns G^t with the updated decision boundaries of the current ViT.

(c) Feature matching for distributions. We further align generated samples with the f_θ^t via feature matching. Let $Z^t(\cdot)$ be the feature extractor associated with f_θ^t We define:

$$L_{\mathrm{fm}}^{G^t} = \mathbb{E}_{(x,c)\sim p_{\mathrm{data}}^t,\; z\sim p_z}\Big[\big\|Z^t(G^t(z,c)) - Z^t(x)\big\|^2\Big], \quad (6)$$

By minimizing this loss, the generator strives to produce samples that remain aligned with the feature representation of f_θ^t.

(d) Diversity loss with $z_i \neq z_j$. To avoid mode collapse and encourage intra-class diversity, we measure the distance between pairs of samples generated from two distinct noise vectors $z_i \neq z_j$. Let $F_s(\cdot)$ be a feature mapping (e.g., an intermediate layer of ViT). We define:

$$\Delta x \;=\; F_s\big(G^t(z_i,c)\big) \;-\; F_s\big(G^t(z_j,c)\big) \quad (7)$$

$$L_{\mathrm{divGan}} \;=\; \mathbb{E}_{c\sim p_{\mathrm{data}}^t(c),\; z_i,z_j\sim p_z,\; z_i\neq z_j}\Big[\frac{1}{\|\Delta x\|^2+\epsilon}\Big]. \quad (8)$$

where ϵ is a small constant to avoid division by zero. We only consider pairs (z_i, z_j) with $z_i \neq z_j$ to ensure meaningful distance comparisons. This objective encourages G^t to increase the pairwise distance in the feature space.

(e) Overall generator objective. Combining the above terms, the generator loss is given by:

$$L_{G^t} \;=\; L_{\mathrm{GAN}}^{G^t} \;+\; \lambda_{\mathrm{cls}}\, L_{\mathrm{cls}} \;+\; \lambda_{\mathrm{div}}\, L_{\mathrm{divGan}} \;+\; \mu\, L_{\mathrm{fm}}^{G^t} \quad (9)$$

Here, λ_{cls}, λ_{div}, and μ control the influence of classification consistency, diversity, and feature matching, respectively.

IV. Experiments

In this section, we present a comprehensive experimental setup designed to evaluate the proposed framework. Our goal is to ensure a fair, robust and multifaceted evaluation. In this study, we conduct a systematic comparison on CIFAR-100 [21] and ImageNet-100 [22] under multiple incremental learning configurations to evaluate the effectiveness of our approach. To disentangle the effect of rehearsal memory under different budget regimes, we report results in two complementary scenarios. First, to specifically assess ViTReplay under strict memory constraints, we additionally evaluate the 5-step (20 classes/step) and 10-step (10 classes/step) streams with two small-buffer configurations, i.e., buffer_size=0 (no real-data storage; shown as – in the table) and buffer_size=200 (limited real-data storage), as summarized in Table I. Second, in the main configurations, we use the DyTox protocol with buffer_size=2000 as the broader-storage baseline for comparison. Specifically, CIFAR-100 is incrementally partitioned into scenarios introducing 10, 5, or 2 new classes at each step, yielding 10-step, 20-step, and 50-step settings, respectively. The overall experimental design is based on our newly proposed model and leverages the original DyTox-based continual learning framework. Additionally, DyTox [5] serves as the primary baseline for comparison. To ensure a fair and controlled comparison with DyTox, we maintain parameters and training configurations as closely aligned as possible, modifying or extending only the aspects crucial to our approach. Readers may refer to DyTox's source code for additional settings not explicitly specified here. In this experiment, we set all advanced data augmentation parameters associated with DyTox+, such as reprob, mixup=0.0, and cutmix=0.0, to 0.0 to disable those augmentation techniques. The objective of this study is to compare models under the most basic training paradigm, we do not enable these advanced augmentations; consequently, we only use DyTox as the comparison baseline instead of DyTox+.

We employ several metrics to comprehensively assess continual learning performance, including the last accuracy (Last) and the average accuracy (Avg) at the completion of all incremental steps, forward transfer (FWT) and backward transfer (BWT). BWT [23] measures the impact of learning new tasks on the performance of previously learned tasks and FWT evaluates the extent to which knowledge from earlier tasks facilitates the learning of new tasks. Higher values for these metrics generally represent better performance or reduced forgetting, thereby offering a holistic depiction of the model's continual learning capabilities. Moreover, we evaluate all methods under the standard class-incremental learning (Class-IL) setting. At test time, the task identity is unknown and the model must classify among the union of all seen classes (i.e., no task-ID-based head selection).

In selecting hyperparameters, we consider all loss function coefficients, including $\lambda_{\mathrm{div}}, \lambda_{\mathrm{cls}}, \lambda, \mu$, and α. For instance, when incrementally learning 10 stages on CIFAR-100, selecting $\alpha = 0.5$, $\lambda = 0.1$, $\lambda_{\mathrm{cls}} = 1$, $\mu = 1$, and $\lambda_{\mathrm{div}} = 0.1$ strikes a balance between adversarial intensity and feature alignment, while maintaining sufficient class distinction and diversity in the generator. For larger-scale datasets, such as ImageNet-100, we make minor adjustments to certain loss weights and learning-rate schedules (e.g., raising λ_{div} to 0.3 or 0.5) to accommodate higher resolutions and more classes. Employing learning rates and epoch settings (e.g., 500 epochs per incremental task) comparable to DyTox ensures a more direct and equitable comparison. Overall, the selection of these hyperparameters follows a similar principle.

A. Baselines Comparison

From Table II, it can be observed that ViTReplay consistently achieves higher average accuracy (Avg) than traditional incremental learning approaches under the 10-, 20-, and 50-step settings. Moreover, when compared to DyTox, which is also based on a ViT backbone, ViTReplay exhibits an additional 2%–3% improvement. Notably, ViTReplay, benefiting from generative replay and the bidirectional alignment strategy, continues to maintain high accuracy on previously learned classes, even in longer incremental task sequences. When extended to 50 incremental steps, ViTReplay's advantage over DyTox becomes more evident, reflecting its strong ability to suppress catastrophic forgetting over extended sequences.

Table I further examines ViTReplay's performance in scenarios with no original data storage (*Buffer Size* = 0) and in settings where a limited number of real samples (*Buffer Size* = 200) are retained, compared against various no-buffer methods as well as DyTox. Under the no-buffer configuration, ViTReplay demonstrates significantly higher accuracy than DDGR and BIR in both 5-step and 10-step incremental settings. Furthermore, with a buffer of only 200 real samples, ViTReplay achieves higher average accuracy and last-step accuracy than DyTox. This indicates that, even in a hybrid setting that combines a small buffer with cGAN-generated data, ViTReplay effectively mitigates forgetting and maintains robust new-task learning over long task sequences, demonstrating strong scalability and adaptability.

Compared with DyTox, ViTReplay not only inherits the global self-attention and Task Token mechanisms of ViT, but also refines generator-based replay to better align the old and new feature spaces. In the multi-task incremental setting of ImageNet-100, ViTReplay continues to exhibit superior performance, as shown in Table III. Overall, ViTReplay demonstrates superior accuracy compared with most baseline methods across various incremental steps and effectively reduces forgetting at the final stage.

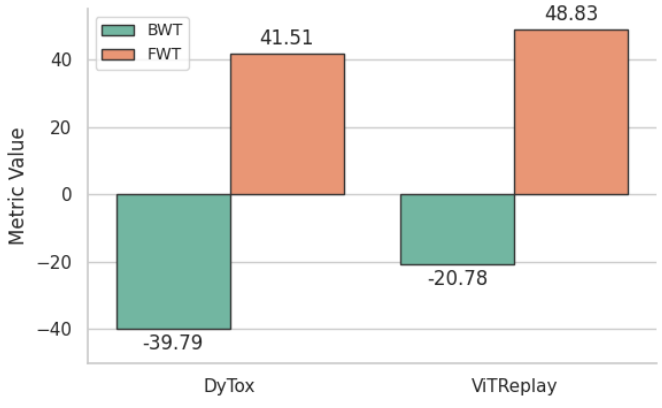

Fig. 2. Comparison of the baseline and the proposed ViTReplay method on the CIFAR-100 dataset under a 10-step incremental learning setting. The performance is evaluated using the BWT (Backward Transfer) and FWT (Forward Transfer) metrics.

From Figure 2, one can directly observe the Backward Transfer (BWT) and Forward Transfer (FWT) [23] metrics for DyTox and ViTReplay. In terms of preserving knowledge from previously learned classes, DyTox exhibits a BWT of approximately -39.79, indicating a substantial negative shift and thus pronounced forgetting of older tasks. Conversely, ViTReplay achieves a BWT of -20.78, still negative, yet clearly reflecting a lower degree of forgetting and stronger retention of old-class performance. Regarding FWT, DyTox attains 41.51, demonstrating a moderate degree of positive influence from old tasks on new-task learning. In comparison, ViTReplay yields an FWT of 48.83, suggesting that its more diverse synthesized samples more effectively facilitate the establishment of robust classification boundaries for incoming classes. These results indicate that ViTReplay not only mitigates forgetting to a greater extent but also leverages previously acquired knowledge more effectively in learning new classes, rendering it a more balanced and efficient approach to incremental learning.

In summary, it outperforms various competing methods, including DyTox, in multi-stage incremental learning. Even under prolonged task sequences, ViTReplay maintains stable accuracy on old classes, highlighting the benefits of continuously updating old-class representations. Moreover, the complementary roles of the diversity loss, class-consistency loss, and dual-directional alignment further enhance the model's resilience in aligning new and old feature spaces and mitigating forgetting, thereby achieving higher average and final accuracies.

B. Ablation Study

In the ablation study, we define four modules, namely, the feature matching loss, the diversity loss, the class-consistency loss, and forward alignment, and test various combinations of these components. As shown in Table IV, ViTReplay exhibits substantial differences in both Last Accuracy (Last Acc) and Average Accuracy (Avg Acc) when compared to the baseline method. When all the enhanced strategies are enabled, ViTReplay achieves a Last Acc of 50.07% and an Avg Acc of 67.74%, demonstrating that the synergy among these modules maximizes the alleviation of forgetting while ensuring high-quality generation and classification precision.

Once any component is removed, the model performance declines to varying degrees. Notably, eliminating reverse alignment (diversity loss, class-consistency loss, feature matching) leads to the most pronounced drop. This suggests that reverse alignment plays a critical role in boosting the generator's effectiveness and preserving previously learned classes.

Under the same experimental conditions, DyTox attains a Last Acc of only 45.61% and an Avg Acc of 64.82%, both substantially lower than ViTReplay with its complete configuration. These results suggest that dual-directional alignment substantially improves performance. Consequently, these strategies are highly effective in preventing mode collapse, aligning the current decision boundaries, and retaining previously learned classes. Compared with earlier approaches, they

TABLE I

COMPARISON OF DIFFERENT METHODS UNDER 5-STEP AND 10-STEP INCREMENTAL SETTINGS ON CIFAR-100. "BUFFER SIZE" INDICATES WHETHER A METHOD STORES REAL SAMPLES (200) OR RELIES SOLELY ON SYNTHESIZED DATA (–).

Methods	**Buffer Size**	**5 Steps**		**10 Steps**	
		Avg	**Last**	**Avg**	**Last**
DDGR [24]	-	28.11±2.58	-	15.99±1.08	-
BIR [25]	-	21.75±0.08	-	15.26±0.49	-
ViTReplay	-	31.56±0.85	20.59±0.74	20.87±0.45	12.98±0.27
DyTox	200	57.24±0.56	42.34±0.23	50.08±0.78	33.42±0.98
ViTReplay	200	**61.96±0.34**	**44.87±0.63**	**55.65±1.12**	**37.59±0.57**

TABLE II

COMPARISON OF VARIOUS METHODS ON THE CIFAR-100 DATASET UNDER DIFFERENT INCREMENTAL LEARNING SETUPS (10 STEPS, 20 STEPS, AND 50 STEPS). NOW EACH SETUP REPORTS #P (NUMBER OF PARAMETERS), THE AVERAGE ACCURACY (AVG) AND THE LAST ACCURACY (LAST). MISSING VALUES ARE REPLACED WITH "-".

	10 steps			20 steps			50 steps		
Methods	**#P**	**Avg**	**Last**	**#P**	**Avg**	**Last**	**#P**	**Avg**	**Last**
ResNet18 Joint	11.22	–	80.41	11.22	–	81.49	11.22	–	81.74
Transf. Joint	10.72	–	76.12	10.72	–	76.12	10.72	–	76.12
iCaRL [26]	11.22	65.27±1.02	50.74	11.22	61.20±0.83	43.75	11.22	56.08±0.83	36.62
UCIR [27]	11.22	58.66±0.71	43.39	11.22	58.17±0.30	40.63	11.22	56.86±0.83	37.09
BiC [28]	11.22	68.80±1.20	53.54	11.22	66.48±0.32	47.02	11.22	62.09±0.85	41.04
WA [29]	11.22	69.46±0.29	53.78	11.22	67.33±0.15	47.31	11.22	64.32±0.28	42.14
PODNet [30]	11.22	58.03±1.27	41.05	11.22	53.97±0.85	35.02	11.22	51.19±1.02	32.99
RPSNet [31]	56.5	68.60	57.05	-	–	–	-	–	–
DyTox [5]	10.73	71.50	57.76	10.74	68.86	51.47	10.77	64.82	45.61
ViTReplay	22.36	**73.27±0.33**	**59.67±0.29**	35.97	**71.08±0.26**	**54.94±0.35**	69.23	**67.74±0.51**	**50.07±0.43**

TABLE III

COMPARISON ON IMAGENET-100 OVER 10 INCREMENTAL STEPS. #P DENOTES THE NUMBER OF PARAMETERS (IN MILLIONS). BOTH TOP-1 AND TOP-5 RESULTS ARE REPORTED.

Methods	**#P**	**top-1**		**top-5**	
		Avg	**Last**	**Avg**	**Last**
ResNet18 joint	11.22	-	-	-	95.10
Transf. joint	11.00	-	79.12	-	93.48
DyTox	11.01	71.85	57.94	90.72	83.52
ViTReplay	23.12	**73.24**	**59.89**	**92.26**	**85.36**

TABLE IV

THE ABLATION STUDY RESULTS ON THE CIFAR-100 UNDER A 50-STEP INCREMENTAL LEARNING SETTING ARE PRESENTED, REPORTING BOTH THE AVERAGE ACCURACY AND THE LAST ACCURACY.

FORWARD ALIGNMENT	DIVERSITY LOSS	CLASS-CONSISTENCY	FEATURE MATCHING	LAST ACC	AVG ACC
ViTReplay				**50.07**	**67.74**
✓	✗	✓	✓	49.11	66.82
✓	✗	✗	✓	48.72	66.24
✓	✓	✗	✗	47.91	66.71
✓	✗	✓	✗	48.19	66.59
✓	✓	✓	✗	49.71	67.27
✓	✓	✗	✓	48.97	67.09
✓	✗	✗	✗	47.35	65.78
DyTox				45.61	64.82

more effectively mitigate forgetting and enhance incremental learning performance.

C. Computational Resources and Memory Usage

In multi-step incremental learning scenarios, an important measure of success is striking a balance between controlling computational resources and memory overhead during training/inference. ViTReplay adopts distinct trade-offs and optimizations in this regard. First, compared to DyTox, the increase in parameter count of ViTReplay primarily arises from the inclusion of class-conditional GANs mechanism. And ViTReplay utilizes the same Transformer backbone as DyTox for classification, therefore, the auxiliary generative modules are excluded from the classification and inference pipeline. Consequently, the deployed inference model has an almost identical parameter footprint to DyTox (10.73), indicating that parameter storage and deployment are not a practical concern. In this work, we primarily focus on storage and inference efficiency; we note that training the task-specific cGAN introduces additional training-time cost, while not increasing inference-time overhead. Overall, ViTReplay effectively balances performance and resource usage in multi-stage incremental learning. It remains a viable option for scenarios where hardware resources are limited yet stable accuracy is still desired.

Moreover, for memory-constrained environments, ViTReplay offers the option to set the replay buffer (Buffer Size) to 0, thereby entirely avoiding stored memory for old classes and relying on a generator (cGAN) to dynamically synthesize old-task samples. Since there is no need to load or manage a large number of real images during training, this zero-

buffer configuration can greatly alleviate resource usage in cases where hardware is tight or storing old-class data is impractical. Therefore, the proposed method is particularly well suited to privacy- and security-sensitive settings where storing real samples is infeasible, thereby offering substantial practical value in real-world applications.

V. Conclusion and Future Work

This paper presents a ViT-based continual learning framework that integrates class-conditional generative adversarial networks (cGANs) with Vision Transformers (ViTs) through a bidirectional alignment strategy. Structurally, the framework establishes a two-way alignment between the cGAN and the ViT: replayed images help regularize the ViT when new tasks arrive, mitigating representation drift and promoting sample diversity, while the ViT provides semantic guidance that improves the fidelity of the generated samples. Moreover, experimental results demonstrate that the proposed method exhibits robust performance in both data-free environments and scenarios where real data can be retained, highlighting its adaptability in various continual learning applications.

Despite these promising findings, several directions remain for future work. We can explore more advanced generative models (e.g., diffusion-based generators) and develop alignment strategies tailored to the characteristics of different generators. Furthermore, investigating interpretability techniques designed for Transformer architectures may enable a more systematic understanding of the model's adaptive dynamics and attention allocation across tasks, thereby guiding more principled refinements to the replay process.

Acknowledgment

This material is based upon work supported in part by NSF under Award IIS-2325863. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the NSF.

References

[1] H. Qu, H. Rahmani, L. Xu, B. Williams, and J. Liu, "Recent advances of continual learning in computer vision: An overview," 2024. [Online]. Available: https://arxiv.org/abs/2109.11369

[2] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 7, pp. 3366–3385, 2021.

[3] L. Wang, X. Zhang, H. Su, and J. Zhu, "A comprehensive survey of continual learning: theory, method and application," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[4] B. Ermis, G. Zappella, M. Wistuba, A. Rawal, and C. Archambeau, "Memory efficient continual learning with transformers," 2023. [Online]. Available: https://arxiv.org/abs/2203.04640

[5] A. Douillard, A. Ramé, G. Couairon, and M. Cord, "Dytox: Transformers for continual learning with dynamic token expansion," 2022. [Online]. Available: https://arxiv.org/abs/2111.11326

[6] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015. [Online]. Available: https://arxiv.org/abs/1503.02531

[7] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *International Journal of Computer Vision*, vol. 129, no. 6, pp. 1789–1819, 2021.

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[9] M. Mirza, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[10] A. Dosovitskiy, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[11] K. Lee, H. Chang, L. Jiang, H. Zhang, Z. Tu, and C. Liu, "Vitgan: Training gans with vision transformers," *arXiv preprint arXiv:2107.04589*, 2021.

[12] Y. Chang, W. Li, J. Peng, B. Tang, Y. Kang, Y. Lei, Y. Gui, Q. Zhu, Y. Liu, and H. Li, "Reviewing continual learning from the perspective of human-level intelligence," 2021. [Online]. Available: https://arxiv.org/abs/2111.11964

[13] B. Tang, *Information Theory for Modern Machine Learning*, 2025.

[14] X. Li, B. Tang, and H. Li, "Adaer: An adaptive experience replay approach for continual lifelong learning," *Neurocomputing*, vol. 572, p. 127204, Mar. 2024. [Online]. Available: http://dx.doi.org/10.1016/j.neucom.2023.127204

[15] R. Aljundi, M. Lin, B. Goujaud, and Y. Bengio, "Gradient based sample selection for online continual learning," *Advances in neural information processing systems*, vol. 32, 2019.

[16] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[17] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, Z. Yang, Y. Zhang, and D. Tao, "A survey on vision transformer," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 87–110, 2023.

[18] M. Xue, H. Zhang, J. Song, and M. Song, "Meta-attention for vit-backed continual learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 150–159.

[19] Z. Wang, L. Liu, Y. Duan, Y. Kong, and D. Tao, "Continual learning with lifelong vision transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 171–181.

[20] J. Shlens, "Notes on kullback-leibler divergence and likelihood," 2014. [Online]. Available: https://arxiv.org/abs/1404.2000

[21] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," *Technical Report*, 2009.

[22] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[23] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," 2022. [Online]. Available: https://arxiv.org/abs/1706.08840

[24] R. Gao and W. Liu, "Ddgr: Continual learning with deep diffusion-based generative replay," in *International Conference on Machine Learning*. PMLR, 2023, pp. 10 744–10 763.

[25] G. M. Van de Ven, H. T. Siegelmann, and A. S. Tolias, "Brain-inspired replay for continual learning with artificial neural networks," *Nature communications*, vol. 11, no. 1, p. 4069, 2020.

[26] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "icarl: Incremental classifier and representation learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[27] S. Hou, X. Pan, C. Change Loy, Z. Wang, and D. Lin, "Learning a unified classifier incrementally via rebalancing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[28] Y. Wu, Y. Chen, L. Wang, Y. Ye, Z. Liu, Y. Guo, and Y. Fu, "Large scale incremental learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[29] B. Zhao, X. Xiao, G. Gan, B. Zhang, and S. Xia, "Maintaining discrimination and fairness in class incremental learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[30] A. Douillard, M. Cord, C. Ollion, T. Robert, and E. Valle, "Podnet: Pooled outputs distillation for small-tasks incremental learning," in *Proceedings of the IEEE European Conference on Computer Vision (ECCV)*, 2020.

[31] J. Rajasegaran, M. Hayat, S. Khan, F. Shahbaz Khan, L. Shao, and M.-H. Yang, "An adaptive random path selection approach for incremental learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.