

FAIROCL: FAIR GRADIENT AGGREGATION FOR ON-LINE CONTINUAL LEARNING

Anonymous authors

Paper under double-blind review

ABSTRACT

Online continual learning (OCL) aims to enable neural networks to learn sequentially from streaming data while mitigating catastrophic forgetting, a key challenge in which learning new tasks interferes with the retention of previously acquired knowledge. Although most existing approaches rely on memory buffers to replay past samples, training jointly on mixed data from different tasks often leads to gradient conflicts, which undermine model performance. To address this, we propose FairOCL, a framework that draws inspiration from fair resource allocation in communication networks. FairOCL formulates gradient aggregation across tasks as a constrained utility maximization problem and enforces fairness in the optimization process, allowing principled control over task prioritization. Extensive experiments on several standard benchmarks show that FairOCL achieves consistent improvements over state-of-the-art methods. Our code will be released.

1 INTRODUCTION

Continual learning (CL) aims to equip machine learning models with a remarkable capability of human intelligence, *i.e.*, the ability to learn continuously over time, acquiring new knowledge while retaining and integrating prior experience (McCloskey & Cohen, 1989). This capability is crucial for developing general-purpose intelligent systems that can adapt to dynamic and evolving environments without forgetting previously learned information. Traditional CL methods typically operate under the *offline* setting (Parisi et al., 2019), where models have access to the full dataset of each task and are allowed multiple passes over the data. In contrast, *online* continual learning (OCL) presents a more realistic and challenging scenario, where data arrive sequentially in a stream and each sample can be used only once. This streaming constraint closely aligns with practical deployment scenarios such as real-time learning in robotics, edge computing, and user-facing systems (Mai et al., 2022; Gunasekara et al., 2023). In this paper, we focus on the challenging class incremental learning setting of the OCL mode, where the model incrementally learns to recognize all classes observed so far from a data stream in a single pass, and classify test instances without access to task identity.

Experience replay has emerged as the cornerstone of nearly all successful OCL methods (Chaudhry et al., 2019; Guo et al., 2022; Soutif-Cormerais et al., 2023), where a small subset of past data is stored in a memory buffer and interleaved with current samples during training. Earlier approaches imply a *uniform gradient aggregation* strategy and treat gradients from previous tasks equally during joint optimization (Rolnick et al., 2019; Buzzega et al., 2020; Caccia et al., 2022; Wang et al., 2023). However, this simple scheme ignores task heterogeneity, allowing a few dominant tasks to overshadow others and resulting in **unbalanced updates**. *Gradient transformation* approaches modify gradients through projections, constraining gradient updates to be approximately orthogonal to the subspace of past representations to preserve their high-level statistics (Zeng et al., 2019; Saha et al., 2021; Deng et al., 2021; Yoo et al., 2024). However, these methods focus on preserving individual task representations and do not explicitly balance their relative influence, which may result in **uneven retention** of past task knowledge. *Gradient alignment* methods instead reweight gradients according to their conflict with the current task, emphasizing conflicting ones while discarding others (Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2019; Riemer et al., 2019; Guo et al., 2020; Gupta et al., 2020). Though effective at reducing current-past interference, these approaches overlook conflicts among past tasks, which leads to **imbalanced retention**. More recently, Pareto optimization has been explored to capture inter-task relationships more holistically (Wu et al., 2024), but this approach requires complex hyper-gradient computations for stable training.

To tackle the problem of **achieving balanced learning** across past tasks in OCL, we propose FairOCL, a principled framework that mitigates gradient interference in replay-based methods. Our approach builds on an analogy with fair resource allocation in communication networks (Mo & Walrand, 2000). We view each past task as a user competing for resources, where the shared gradient direction serves as the resource to reduce task losses, and the “service quality” of a task is measured by its loss reduction after updating along this direction. Under this perspective, replay-based OCL can be formulated as a utility maximization problem, where each task is assigned an α -fair utility function. Different choices of α correspond to different fairness behaviors, among which certain settings coincide with classical notions such as max-min (Radunovic & Le Boudec, 2007), proportional (Kelly, 1997), and minimum potential delay fairness (Kelly et al., 1998). From a CL standpoint, this fairness-driven formulation enhances stability by enforcing fairness among past task gradients to prevent uneven retention, while preserving plasticity by ensuring sufficient gradient contribution from the current task. In addition, to counteract the task recency bias (Chrysakis & Moens, 2023) inherent in replay-based methods, where predictions are biased toward recently observed data, we incorporate knowledge distillation (Hinton et al., 2015) to stabilize feature representations and mitigate drift in earlier tasks.

Our contributions are three-fold: **i)** We provide a fresh perspective that connects replay-based OCL with fair resource allocation in communication networks, framing gradient aggregation as a fairness-driven problem. **ii)** We propose FairOCL, the first framework to formulate gradient aggregation in replay-based OCL as a fairness-constrained utility maximization problem, enabling dynamic and balanced task contributions during training. **iii)** To further mitigate task recency bias inherent in replay-based methods, we incorporate knowledge distillation to stabilize past representations and improve retention. Extensive experiments on standard benchmarks, including CIFAR-10 Krizhevsky et al. (2009), CIFAR-100 Krizhevsky et al. (2009), TinyImageNet Le & Yang (2015), and ImageNet-100 Hou et al. (2019), demonstrate that FairOCL consistently outperforms competitive state-of-the-art baselines across varying buffer sizes, validating both its effectiveness and generality.

2 RELATED WORK

Online continual learning (OCL) and Replay-based Methods. OCL aims to train models to adapt to non-stationary data streams under a strict single-pass constraint while mitigating catastrophic forgetting (McCloskey & Cohen, 1989). Unlike offline continual learning, which permits multiple passes over task datasets (Parisi et al., 2019), OCL operates under more realistic and restrictive conditions. A foundational strategy in OCL is experience replay (Rolnick et al., 2019), which stores a subset of past data in a memory buffer and interleaves it with incoming samples during training. Recent extensions include DER++ (Buzzega et al., 2020), which distills past experiences from the logits of samples stored in the memory buffer, ER-ACE (Caccia et al., 2022), which introduces asymmetric losses to balance new and replay data, and CBA (Wang et al., 2023), which utilizes stored past samples to correct the recency bias through a bias attractor. Despite their effectiveness, these methods typically aggregate gradients from past and current tasks with uniform weighting, which often produces conflicting gradient directions and ultimately degrades performance.

Gradient Transformation Methods. One way to mitigate gradient conflicts is to apply projection-based transformations, where gradient updates are constrained to be orthogonal to the subspace spanned by past representations so as to preserve their high-level statistics. Representative examples include OGD (Zeng et al., 2019) and GPM (Saha et al., 2021), which build orthogonal subspaces to protect important representations of earlier tasks. Later methods improved estimation of projection subspace either by maintaining averaged mini-batch activations to avoid capacity exhaustion (Zeng et al., 2019; Guo et al., 2022) or by employing low-rank approximations of past activations in an SVD-like manner (Saha et al., 2021; Deng et al., 2021; Lin et al., 2022; Saha & Roy, 2023). Recently, LPR (Yoo et al., 2024) extended this line of work to the online setting, introducing a layerwise proximal point formulation that stabilizes replay-based training by encouraging gradual representation updates. Although effective in preserving knowledge, these methods regulate tasks independently without explicitly balancing their relative influences, potentially leading to uneven retention.

Gradient Alignment Methods. A complementary line of work mitigates gradient conflicts by adjusting update directions according to their compatibility with past task gradients. GEM (Lopez-Paz & Ranzato, 2017) enforces non-negative inner products between the current gradient update direction and past task gradients, ensuring that new learning does not harm prior knowledge. More efficient variants such as A-GEM (Chaudhry et al., 2019) and MEGA (Guo et al., 2020) reduce

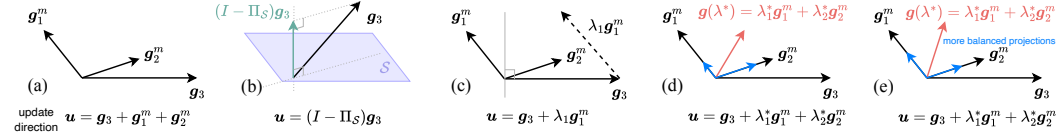


Figure 1: **Conceptual illustration of gradient update strategies.** Here, $\mathcal{S}$ is the subspace spanned by past-task representations. (a) **Uniform gradient aggregation** treat current and past-task gradients equally. (b) **Gradient Transformation** projects g_3 onto the subspace orthogonal complement of $\mathcal{S}$ to preserve prior knowledge. (c) **Gradient Alignment** considers g_1^m that conflicts with g_3 while neglecting g_2 that does not. (d) **POCL** (Wu et al., 2024) obtains weights λ^* via Pareto optimization to flexibly manage inter-task trade-offs. (e) **FairOCL (Ours)** formulates gradient aggregation as an α -fair utility maximization problem and yields λ^* that achieves *more balanced* projections (blue arrows) across tasks. See §3 and §4 for details.

computational cost by averaging historical gradients rather than applying the aforementioned inner project constraint individually. Meta-learning approaches such as MER (Riemer et al., 2019) and La-MAML (Gupta et al., 2020) further adapt update directions through bi-level optimization. Though highlighting the importance of handling gradient conflicts, these methods primarily focus on resolving conflicts between the current task and past tasks, while overlooking relationships among past tasks themselves. More recently, POCL (Wu et al., 2024) employs Pareto optimization to capture inter-task relationships and achieve more holistic gradient balancing, but this approach relies on computationally intensive hyper-gradient calculations for stability. Our method extends this research direction by incorporating fairness-driven constraints, thereby promoting balanced gradients across past tasks without relying on hyper-gradients.

Fairness in Capacity-Constrained Resource Allocation. Fair resource allocation has long been studied in wireless communication, where limited resources such as power and bandwidth must be distributed among users in a manner that balances equity and efficiency, and has more recently also appeared in settings like federated learning (Li et al., 2020; Zhang et al., 2022) and multi-task learning (Ban & Ji, 2024). To formalize this trade-off, researchers have proposed a variety of fairness metrics. Jain’s fairness index (Jain et al., 1984) quantifies the uniformity of allocations, while proportional fairness (Kelly, 1997) strikes a compromise between equity and throughput by allocating resources in proportion to demand. Max-min fairness (Radunovic & Le Boudec, 2007) instead protects the least-served user by maximizing their minimum allocation. These notions are unified under the α -fairness framework (Mo & Walrand, 2000; Lan et al., 2010), where varying α interpolates between utilitarian efficiency ($\alpha \rightarrow 0$) and max-min equity ($\alpha \rightarrow \infty$). In this paper, we bridge OCL with fair resource allocation theory and propose FairOCL, which formulates gradient aggregation as an α -fair utility maximization problem to alleviate gradient interference and mitigate forgetting, while preserving plasticity to learn new tasks.

3 PRELIMINARIES

3.1 GRADIENT AGGREGATION IN REPLAY-BASED OCL

Online continual learning (OCL) aims to incrementally learn a sequence of N tasks from a non-stationary data stream under strict single-pass constraints. Let g_n denote the gradient of the current task n computed from its incoming batch, where $n = 1, \dots, N$. Let g_i^m denote the gradient of a previous task $i < n$ computed from stored samples in the memory buffer $\mathcal{M}$.

Traditional replay-based OCL methods (Rolnick et al., 2019; Buzzega et al., 2020; Caccia et al., 2022) calculate the gradient update direction u by aggregating the current gradient g_n with previous task gradients g_i^m through uniform weighting (cf., Fig. 1 (a)):

$$u = g_n + \sum_{i=1}^{n-1} g_i^m. \quad (1)$$

While this *uniform gradient weighting* reduces catastrophic forgetting, it often suffers from gradient conflicts due to its naive averaging scheme. To mitigate such conflicts, *gradient transformation* methods (Zeng et al., 2019; Saha et al., 2021; Deng et al., 2021; Yoo et al., 2024) project the current gradient g_n onto the orthogonal complement of $\mathcal{S}$ to preserve prior representations (cf., Fig. 1 (b)),

where $\mathcal{S}$ is the subspace spanned by past-task representations. *Gradient alignment* methods (Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2019; Guo et al., 2020) instead determine $\mathbf{u}$ by solving:

$$\max_{\mathbf{u}} -\frac{1}{2}\|\mathbf{g}_n - \mathbf{u}\|_2^2, \quad \text{s.t. } \langle \mathbf{u}, \mathbf{g}_i^m \rangle \geq 0, \quad i = 1, \dots, n-1. \quad (2)$$

This formulation ensures that $\mathbf{u}$ remains close to the current task gradient $\mathbf{g}_n$ to preserve the performance of the current task, while maintaining non-negative alignment with past task gradient $\mathbf{g}_i^m$ to prevent negative backward transfer (cf., Fig. 1 (c)). However, the solution implicitly prioritizes past tasks that conflict with the current gradient, while neglecting inter-task relationships.

Building on these formulations, Wu et al. (2024) (cf., Fig. 1 (d)) derived a ***hierarchical gradient aggregation framework***, reformulating the update direction $\mathbf{u}$ as:

$$\mathbf{u} = \mathbf{g}_n + \mathbf{g}(\boldsymbol{\lambda}) = \mathbf{g}_n + \sum_{i=1}^{n-1} \lambda_i \mathbf{g}_i^m, \quad (3)$$

where $\mathbf{g}_n$ is the current task gradient, and $\mathbf{g}(\boldsymbol{\lambda})$ aggregates past gradients with weights λ_i . This reveals that replay-based methods inherently prioritize the current task gradient, while the aggregation of past task gradients primarily serve as a regularization to mitigate forgetting. Thus, effective OCL requires not only preserving the current task gradient but also fairly balancing past task gradients.

3.2 FAIRNESS IN RESOURCE ALLOCATION

An analogous challenge arises in communication networks, where fairness is a central principle for allocating limited resources (e.g., channel bandwidth, transmission rate) among K users. A common approach formulates this as a utility maximization problem under link capacity constraints:

$$\max_{x_1, \dots, x_K \in C} \sum_{i \in [K]} u(x_i) := (1 - \alpha)^{-1} x_i^{1-\alpha}, \quad \alpha \in [0, 1) \cup (1, +\infty), \quad (4)$$

where x_i denotes the allocation for user i (e.g., transmission rate), $u(\cdot)$ is the utility (e.g., throughput), and C represents the convex set of feasible allocations. The function $(1-\alpha)^{-1} x_i^{1-\alpha}$ is known as the ***α -fair function***. Different choices of α recover widely studied fairness notions. **Proportional fairness** arises as $\alpha \rightarrow 1$, where the utility reduces to $\log x_i$, and the objective becomes the sum of logarithmic utilities. **Minimum potential delay fairness** corresponds to $\alpha = 2$, where the objective to maximize $\sum_i -1/x_i$ can be interpreted as minimizing the total delay for transferring unit-size files. **Max-min fairness** is obtained as $\alpha \rightarrow +\infty$, which protects the least-served user by maximizing their minimum allocation. We refer readers to (Mo & Walrand, 2000) and (Ban & Ji, 2024) for detailed derivations and more discussion. This well-established framework motivates our treatment of gradient aggregation in OCL as a fairness-driven resource allocation problem.

4 FAIROCL: FAIR GRADIENT AGGREGATION IN OCL

Inspired by the principles of fair resource allocation, we propose FairOCL, a framework for fairness-aware and adaptive gradient aggregation in OCL. FairOCL addresses the hierarchical gradient aggregation by regulating past-task gradients to ensure more balanced contributions (cf., Fig. 1 (e)).

4.1 PROBLEM FORMULATION

Recall the hierarchical formulation in Eq. (3), where each $\mathbf{g}_i^m$ denotes the gradient of a past task i . Our goal is to construct an update direction $\mathbf{d} = \mathbf{g}(\boldsymbol{\lambda})$ that fairly aggregates past-task gradients. Following the analogy to communication networks, we interpret $\mathbf{g}_i^{m\top} \mathbf{d}$ as the utility (loss reduction rate) x_i for task i , and larger $\mathbf{g}_i^{m\top} \mathbf{d}$ indicate more favorable optimization for task i .

To promote fairness among $n-1$ past tasks, we adopt the α -fair utility function in Eq. (4), and view each past task as a user in network resource allocation. The utility of user i is calculated as $(1-\alpha)^{-1} x_i^{1-\alpha} = (1-\alpha)^{-1} (\mathbf{g}_i^{m\top} \mathbf{d})^{1-\alpha}$, where $\alpha \in [0, 1) \cup (1, +\infty)$. We then formulate:

$$\max_{\mathbf{d} \in B_\epsilon} \sum_{i=1}^{n-1} \frac{(\mathbf{g}_i^{m\top} \mathbf{d})^{1-\alpha}}{1-\alpha}, \quad \text{s.t. } \mathbf{g}_i^{m\top} \mathbf{d} \geq 0, \quad (5)$$

where B_ϵ is a ball of radius ϵ centered at the origin, enforcing a bound on the magnitude of the update direction $\mathbf{d}$, similar to the link capacity constraints in network resource allocation. The convexity of the constraint set of $\mathbf{d}$ ensures the tractability of the problem. This formulation offers a principled and effective perspective on fairness-aware gradient aggregation for OCL, enabling FairOCL to adaptively modulate task contributions while maintaining optimization efficiency under constraints.

Algorithm 1 FairOCL Training Procedure

Require: Initial student parameters θ_0 ; initialize teacher parameters as $\theta_0^{\text{tea}} \leftarrow \theta_0$, fairness parameter α , memory buffer $\mathcal{M}$, learning rate η , number of tasks N

- 1: **for** task $n = 1$ to N **do** ▷ Sequential task learning
- 2: Initialize task model: $\theta_n^{(0)} \leftarrow \theta_{n-1}$, $\theta_n^{\text{tea}(0)} \leftarrow \theta_{n-1}^{\text{tea}}$
- 3: **for** each batch $\mathcal{B}_n^{(b)}$ in task n **do** ▷ Online updates
- 4: Compute current task gradient: $\mathbf{g}_n \leftarrow \nabla_{\theta} \mathcal{L}(\theta_n^{(b-1)}, \theta_n^{\text{tea}(b-1)}; \mathcal{B}_n^{(b)})$ ▷ Eq. (9)
- 5: Compute past task gradients: $[\mathbf{g}_1^m, \dots, \mathbf{g}_{n-1}^m] \leftarrow \nabla_{\theta} \mathcal{L}(\theta_n^{(b-1)}, \theta_n^{\text{tea}(b-1)}; \mathcal{M})$ ▷ Eq. (9)
- 6: Solve Eq. (7) to obtain weights $\lambda \leftarrow w^*$
- 7: Update direction: $\mathbf{u} = \mathbf{g}_n + \sum_{i=1}^{n-1} \lambda_i \mathbf{g}_i^m$
- 8: Update student: $\theta_n^{(b)} = \theta_n^{(b-1)} - \eta \mathbf{u}$
- 9: Update teacher via EMA ▷ Eq. (8)
- 10: Update memory buffer $\mathcal{M}$ via reservoir sampling (Vitter, 1985)
- 11: **end for**
- 12: Final task model: $\theta_n \leftarrow \theta_n^{(b)}$, $\theta_n^{\text{tea}} \leftarrow \theta_n^{\text{tea}(b)}$
- 13: **end for**

4.2 OPTIMIZATION METHOD

Since the objective in Eq. (5) is monotonically non-decreasing in $\mathbf{d}$, the optimal solution lies on the boundary of B_c . At the optimum, the KKT conditions (Boyd & Vandenberghe, 2004) imply:

$$\sum_i \mathbf{g}_i^m (\mathbf{g}_i^{m\top} \mathbf{d})^\alpha = c \mathbf{d}, \quad c > 0. \quad (6)$$

Following (Ban & Ji, 2024), we set $c = 1$ for simplicity. Assume that the task gradients $\{\mathbf{g}_i^m\}$ are linearly independent, we can express $\mathbf{d}$ as a non-negative combination $\mathbf{d} = \sum_i \omega_i \mathbf{g}_i^m$. Substituting yields the relation $(\mathbf{g}_i^{m\top} \mathbf{d})^\alpha = \omega_i$, which gives the system:

$$\mathbf{G}^\top \mathbf{G} \omega = \omega^{-1/\alpha}, \quad \alpha \neq 0, \quad (7)$$

where $\mathbf{G} = [\mathbf{g}_1 \cdots \mathbf{g}_n]$ is a gradient matrix and the exponent $-1/\alpha$ is applied elementwise. When $\alpha = 0$, this reduces to uniform weighting where $\omega_i = 1$ for all i . Following (Ban & Ji, 2024), we solve Eq. (7) by using a constrained nonlinear least squares formulation. Since FairOCL performs a single backward pass over the current batch combined with memory samples, its computational cost does not increase with the number of tasks. The overall training budget remains comparable to other replay-based baselines (see §A.4).

4.3 ALLEVIATING FEATURE DRIFT WITH KNOWLEDGE DISTILLATION

While fair gradient aggregation ensures balanced updates across tasks, replay-based OCL still suffers from task recency bias (Chrysakis & Moens, 2023), which can compromise gradient aggregation as it undermines the stability of representations. To mitigate this, we integrate knowledge distillation with an Exponential Moving Average (EMA) teacher (Tarvainen & Valpola, 2017), where the teacher parameters are updated as:

$$\theta_n^{\text{tea}(b)} = \gamma \theta_n^{(b)} + (1 - \gamma) \theta_n^{\text{tea}(b-1)}, \quad n = 1, \dots, N. \quad (8)$$

Here, $\theta_n^{(b)}$ denotes the student model parameters at task n , batch b , and $\gamma \in [0, 1)$ is the momentum coefficient, which is set to 0.1 empirically in our experiments. The overall training loss integrates standard cross-entropy with a distillation term:

$$\mathcal{L}(\theta, \theta^{\text{tea}}; \mathcal{D}) = \mathcal{L}_{\text{CE}}(\theta; \mathcal{D}) + \beta \mathcal{L}_{\text{KL}}(\theta, \theta^{\text{tea}}, \tau; \mathcal{D}), \quad (9)$$

where $\mathcal{D} = \{\mathcal{B}_n^{(b)}, \mathcal{M}\}$ is the current training data, $\mathcal{L}_{\text{CE}}$ is the cross-entropy loss, $\mathcal{L}_{\text{KL}}$ is the KL divergence with temperature τ , and β controls the relative contribution of the distillation term. The overall training pipeline is detailed in Alg. 1. At inference, we average the final weights of teacher and student to obtain a stable model estimate following (Tarvainen & Valpola, 2017).

Table 1: **Average Accuracy (Acc) (%)**, $\uparrow$ on four benchmark datasets with different memory buffer size $|\mathcal{M}|$. Displayed values are the mean and standard deviation computed over 5 runs. See §5.2.

Dataset	CIFAR-10			CIFAR-100			TinyImageNet			ImageNet-100		
$ \mathcal{M} $	200	500	1000	1000	2000	5000	2000	5000	10000	2000	5000	10000
ER (NeurIPS'19)	46.33 $\pm$ 2.42	55.73 $\pm$ 2.04	62.99 $\pm$ 2.10	23.00 $\pm$ 3.80	31.55 $\pm$ 1.27	38.05 $\pm$ 1.08	12.29 $\pm$ 0.44	19.64 $\pm$ 1.14	26.92 $\pm$ 1.89	19.06 $\pm$ 0.90	29.74 $\pm$ 1.34	36.72 $\pm$ 1.09
DER++ (NeurIPS'20)	47.07 $\pm$ 0.97	55.53 $\pm$ 1.05	58.51 $\pm$ 0.68	22.80 $\pm$ 1.80	25.89 $\pm$ 1.46	25.71 $\pm$ 2.40	16.10 $\pm$ 1.23	18.49 $\pm$ 1.08	18.06 $\pm$ 1.30	15.36 $\pm$ 3.04	19.19 $\pm$ 1.55	20.48 $\pm$ 4.67
DVC (CVPR'22)	48.08 $\pm$ 4.27	58.72 $\pm$ 2.03	61.11 $\pm$ 2.97	18.66 $\pm$ 2.54	22.73 $\pm$ 2.90	28.47 $\pm$ 3.95	14.04 $\pm$ 0.75	18.48 $\pm$ 1.49	17.95 $\pm$ 2.93	14.54 $\pm$ 5.15	21.88 $\pm$ 3.45	28.50 $\pm$ 2.93
GSA (CVPR'23)	48.90 $\pm$ 3.38	61.45 $\pm$ 1.95	67.63 $\pm$ 1.24	29.68 $\pm$ 1.54	36.96 $\pm$ 0.79	45.86 $\pm$ 1.89	19.08 $\pm$ 0.18	27.22 $\pm$ 0.95	32.94 $\pm$ 0.75	24.29 $\pm$ 0.59	33.47 $\pm$ 1.18	40.18 $\pm$ 0.93
PCR (CVPR'23)	52.20 $\pm$ 0.66	60.61 $\pm$ 2.23	61.66 $\pm$ 1.86	30.68 $\pm$ 0.81	38.63 $\pm$ 1.01	45.27 $\pm$ 0.78	19.55 $\pm$ 0.54	27.02 $\pm$ 0.37	33.34 $\pm$ 0.22	19.89 $\pm$ 6.24	31.35 $\pm$ 3.01	36.99 $\pm$ 4.70
POCL* (ICML'24)	35.69 $\pm$ 3.82	48.08 $\pm$ 3.72	56.97 $\pm$ 3.88	16.48 $\pm$ 1.45	24.84 $\pm$ 0.89	32.16 $\pm$ 2.44	11.59 $\pm$ 0.48	19.45 $\pm$ 1.72	24.38 $\pm$ 0.98	—	—	—
ER-MKD (ICML'24)	57.54 $\pm$ 2.55	68.48 $\pm$ 0.92	74.33 $\pm$ 0.68	38.50 $\pm$ 0.50	45.20 $\pm$ 0.20	52.10 $\pm$ 0.50	25.03 $\pm$ 0.57	33.32 $\pm$ 0.50	39.69 $\pm$ 0.55	30.67 $\pm$ 0.46	39.71 $\pm$ 1.07	44.92 $\pm$ 0.98
FairOCL	58.65$\pm$0.85	68.71$\pm$0.89	74.83$\pm$0.90	39.47$\pm$0.34	46.53$\pm$1.03	53.39$\pm$0.34	26.92$\pm$0.38	34.95$\pm$0.46	40.52$\pm$0.61	31.19$\pm$1.78	41.18$\pm$1.55	47.22$\pm$0.93

* denotes results obtained by running the official code.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUPS

Datasets. Following (Michel et al., 2024; Wu et al., 2024), we consider four representative benchmarks for evaluating OCL. *CIFAR-10* (Krizhevsky et al., 2009) consists of 10 classes with 50,000 training and 10,000 test images, which is partitioned into five disjoint tasks with two classes each. *CIFAR-100* (Krizhevsky et al., 2009) includes 100 classes with the same number of training and test images as CIFAR-10, which is organized into 10 disjoint tasks of 10 classes each. *TinyImageNet* (Le & Yang, 2015) contains 200 classes, with 100,000 training and 10,000 test images, which is split into 20 non-overlapping tasks of 10 classes each. *ImageNet-100* (Hou et al., 2019) is a 100-class subset of ImageNet-1K (Deng et al., 2009), which is divided into 20 tasks, each comprising 5 classes.

Baselines. To contextualize the performance of our method, we compare it against a diverse set of state-of-the-art replay-based OCL baselines. ER (Rolnick et al., 2019) is a foundational experience replay method that trains the model using cross-entropy loss on a combination of current and buffered samples. DER++ (Buzzega et al., 2020) enhances ER by introducing a logit-level consistency regularization via knowledge distillation. DVC (Gu et al., 2022) augments replay with a dual-view contrastive loss that maximizes mutual information between different augmentations of input data. GSA (Guo et al., 2023) deal with cross-task class discrimination caused by data imbalance by introducing a self-adaptive loss based on gradients. PCR (Lin et al., 2023) adopts a proxy-based contrastive loss to enhance discriminative feature learning across tasks. POCL (Wu et al., 2024) incorporates Pareto optimization to balance trade-offs among tasks with a hyper-gradient-based implementation. ER-MKD is a basic instantiation of MKD-OCL (Michel et al., 2024), which incorporates a momentum-based knowledge distillation strategy into ER, producing a simple yet competitive baseline. We adopt ER-MKD as it allows us to evaluate the core mechanism of MKD-OCL in isolation from additional complexities introduced when combining it with stronger baselines.

Evaluation Metrics. We adopt two standard metrics in the continual learning literature: Average Accuracy and Average Forgetting. Let $A_{i,j}$ denote the test accuracy on task i after training on task j . *Average Accuracy (Acc)* measures the overall model performance, and is computed as the average accuracy after learning all the N tasks: $Acc = \frac{1}{N} \sum_{i=1}^N A_{i,N}$. *Average Forgetting (F)* quantifies forgetting of previously learned tasks, which is computed as the average of the maximum performance drop of each task after learning the final task: $F = \frac{1}{N-1} \sum_{i=1}^{N-1} \max_{t \in \{1, \dots, N-1\}} (A_{i,t} - A_{i,N})$. Together, they comprehensively evaluate both plasticity (via Acc) and stability (via F), providing a balanced view of continual learning performance.

Implementation Details. Following (Guo et al., 2023; Michel et al., 2024), we use a standard ResNet-18 (He et al., 2016) (not pre-trained) as the backbone for all methods. The memory buffer is maintained using reservoir sampling (Vitter, 1985). Our method is trained using Adam (Kingma & Ba, 2015) with an initial learning rate of 0.0005. The input batch size is set to 10 for CIFAR-10, and 16 for all other datasets. For replay, the memory sampling batch size is 64/80/160 for CIFAR-10 with $|\mathcal{M}| = 200/500/1000$, 256 for CIFAR-100 and TinyImageNet, and 128 for ImageNet-100. Each experiment is run on a single NVIDIA GeForce RTX 4090 GPU. The fairness parameter α is set to 0.5 for CIFAR-10, and 1.5 for all other three datasets. For knowledge distillation, we use a temperature $\tau = 3.0$ and a distillation loss weight $\beta = 5.5$.

Table 2: **Average Forgetting (F) (%)** ($\downarrow$) on four benchmark datasets with different memory buffer size $|\mathcal{M}|$. Displayed values are the mean and standard deviation computed over 5 runs. See §5.2.

Dataset	CIFAR-10			CIFAR-100			TinyImageNet			ImageNet-100		
$ \mathcal{M} $	200	500	1000	1000	2000	5000	2000	5000	10000	2000	5000	10000
ER (NeurIPS'19)	52.90 $\pm$ 3.54	34.07 $\pm$ 3.13	23.43 $\pm$ 3.41	33.04 $\pm$ 1.75	22.84 $\pm$ 1.27	14.55 $\pm$ 2.73	40.68 $\pm$ 0.72	28.43 $\pm$ 1.14	20.78 $\pm$ 1.00	43.69 $\pm$ 2.12	28.04 $\pm$ 1.11	0 $\pm$ 3.37
DER++ (NeurIPS'20)	36.93 $\pm$ 5.86	28.26 $\pm$ 5.06	19.47 $\pm$ 3.20	33.01 $\pm$ 4.85	29.31 $\pm$ 3.65	30.10 $\pm$ 3.56	32.28 $\pm$ 7.37	33.56 $\pm$ 3.80	33.42 $\pm$ 4.83	27.68 $\pm$ 1.04	17.53 $\pm$ 2.51	19.04 $\pm$ 0.57
DVC (CVPR'22)	45.36 $\pm$ 4.08	30.64 $\pm$ 2.16	24.21 $\pm$ 3.18	41.24 $\pm$ 2.28	36.20 $\pm$ 2.54	33.26 $\pm$ 3.13	38.20 $\pm$ 2.49	39.01 $\pm$ 5.46	40.71 $\pm$ 7.99	53.97 $\pm$ 2.25	41.88 $\pm$ 4.36	34.86 $\pm$ 4.66
GSA (CVPR'23)	35.85 $\pm$ 4.05	23.74 $\pm$ 2.47	16.93 $\pm$ 2.58	30.65 $\pm$ 0.61	20.78 $\pm$ 1.89	8.15 $\pm$ 0.37	36.98 $\pm$ 0.55	22.72 $\pm$ 0.92	13.66 $\pm$ 0.84	48.30 $\pm$ 1.26	32.14 $\pm$ 0.82	19.15 $\pm$ 0.81
PCR (CVPR'23)	30.90 $\pm$ 4.50	26.92 $\pm$ 5.01	19.57 $\pm$ 3.35	34.45 $\pm$ 1.67	25.41 $\pm$ 1.71	15.03 $\pm$ 1.97	27.88 $\pm$ 2.10	16.86 $\pm$ 0.60	9.78 $\pm$ 0.64	37.93 $\pm$ 1.84	27.17 $\pm$ 3.37	19.04 $\pm$ 2.38
POCL* (ICML'24)	55.01 $\pm$ 4.87	34.82 $\pm$ 6.14	24.78 $\pm$ 4.92	31.73 $\pm$ 0.66	19.31 $\pm$ 1.06	11.94 $\pm$ 2.12	36.04 $\pm$ 0.84	20.18 $\pm$ 2.20	13.53 $\pm$ 1.56	—	—	—
ER-MKD (ICML'24)	39.95 $\pm$ 3.43	21.48 $\pm$ 1.38	10.96 $\pm$ 1.73	16.62 $\pm$ 1.10	10.73 $\pm$ 0.89	5.31 $\pm$ 1.20	25.21 $\pm$ 1.25	16.49 $\pm$ 0.97	10.02 $\pm$ 0.97	25.22 $\pm$ 2.54	17.00 $\pm$ 1.94	11.91 $\pm$ 0.57
FairOCL	32.11 $\pm$ 2.00	20.82 $\pm$ 1.77	13.16 $\pm$ 0.64	15.80 $\pm$ 1.40	10.71 $\pm$ 1.65	6.51 $\pm$ 0.98	19.40 $\pm$ 1.09	14.00 $\pm$ 0.79	9.80 $\pm$ 0.81	23.46 $\pm$ 1.54	14.83 $\pm$ 1.16	11.03 $\pm$ 0.91

* denotes results obtained by running the official code.

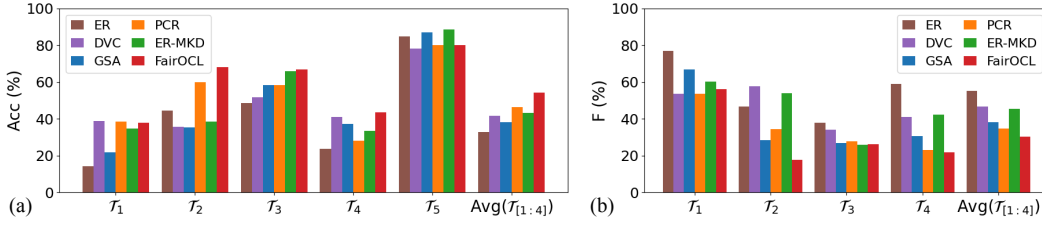


Figure 2: (a) Average Accuracy (Acc) and (b) Average Forgetting (F) of different OCL methods on CIFAR-10 with $N=5$ and buffer size $|\mathcal{M}|$ as 200. Here, $\mathcal{T}_1\text{--}\mathcal{T}_4$ denotes the previously learned tasks, $\mathcal{T}_5$ is the current task, and $\text{Avg}(\mathcal{T}_{1:4})$ means the average performance of past tasks. See §5.3.

5.2 COMPARISON RESULTS

Average Accuracy (Acc) Results. Table 1 reports the mean and standard deviation of Average Accuracy (Acc) over 5 runs on four benchmarks with varying buffer sizes $|\mathcal{M}|$. FairOCL consistently achieves the best performance across all settings. On CIFAR-10, it yields 58.65%, 68.71%, and 74.83% at buffer sizes 200, 500, and 1000, corresponding to relative improvements of 1.93%, 0.34%, and 0.67% over the strongest baseline ER-MKD. On CIFAR-100, FairOCL reaches 39.47%, 46.53%, and 53.39% for buffer sizes 1000, 2000, and 5000, outperforming ER-MKD by 3.22%, 2.94%, and 2.48%. On TinyImageNet, it achieves 26.92%, 34.95%, and 40.52% with buffer sizes 2000, 5000, and 10000, surpassing ER-MKD by 7.55%, 4.89%, and 2.09%. Finally, on ImageNet-100, FairOCL obtains 31.19%, 41.18%, and 47.22% under buffer sizes 2000, 5000, and 10000, with relative gains of 1.70%, 3.70%, and 5.12%. These results demonstrate that FairOCL achieves robust improvements across both simpler and more challenging benchmarks, highlighting its effectiveness in balancing learning and retention. Runtime analysis in §A.4 confirms that FairOCL remains comparable in cost to other replay-based methods.

Average Forgetting (F) Results. Table 2 presents Average Forgetting (F) across the same benchmarks, where lower values indicate better knowledge retention. FairOCL consistently achieves the lowest or near-lowest forgetting across most buffer sizes, confirming its resistance to catastrophic forgetting. On CIFAR-10 and CIFAR-100, it performs best in low-resource regimes, while slightly trailing ER-MKD at larger buffer sizes, though with narrow margins. More importantly, on the more challenging TinyImageNet and ImageNet-100, FairOCL delivers the lowest forgetting across all buffer sizes. For example, it reduces forgetting on TinyImageNet to 19.40%, 14.00%, and 9.80% (buffers 2000, 5000, 10000), and on ImageNet-100 to 23.46%, 14.83%, and 11.03%, outperforming ER-MKD and other strong baselines. Overall, these results confirm that FairOCL not only improves accuracy but also provides superior retention in the OCL setting.

5.3 VERIFICATION ON FAIROCL

Past Task Knowledge Preservation. We first evaluate how well different methods preserve past knowledge by computing Average Accuracy (Acc) on all previously encountered tasks after learning each new task. As shown in Fig. 2 (a), FairOCL maintains consistently high accuracy across tasks, particularly for earlier ones ($\mathcal{T}_1\text{--}\mathcal{T}_4$). While it is not the best on the most recent task $\mathcal{T}_5$, FairOCL

Table 3: **Mean Average Accuracy (Acc)** on previous tasks $\mathcal{T}_{[1:n-1]}$ and on all tasks $\mathcal{T}_{[1:n]}$ for CIFAR-10 with $|\mathcal{M}|=200$ and $N=5$, where $n = 2, \dots, N$. See §5.3 for details.

Methods	$n = 2$		$n = 3$		$n = 4$		$n = 5$	
	$\text{Avg}(\mathcal{T}_1)$	$\text{Avg}(\mathcal{T}_{[1:2]})$	$\text{Avg}(\mathcal{T}_{[1:2]})$	$\text{Avg}(\mathcal{T}_{[1:3]})$	$\text{Avg}(\mathcal{T}_{[1:3]})$	$\text{Avg}(\mathcal{T}_{[1:4]})$	$\text{Avg}(\mathcal{T}_{[1:4]})$	$\text{Avg}(\mathcal{T}_{[1:5]})$
ER (NeurIPS'19)	55.98 $\pm$ 1.97	72.34 $\pm$ 0.71	49.36 $\pm$ 6.67	60.38 $\pm$ 4.81	39.45 $\pm$ 4.76	50.62 $\pm$ 3.22	32.98 $\pm$ 1.80	43.73 $\pm$ 1.14
DVC (CVPR'22)	49.29 $\pm$ 17.36	68.28 $\pm$ 6.29	51.50 $\pm$ 3.78	60.55 $\pm$ 1.62	45.12 $\pm$ 2.96	53.99 $\pm$ 1.19	39.17 $\pm$ 5.77	46.97 $\pm$ 3.30
GSA (CVPR'23)	73.05 $\pm$ 10.29	74.12 $\pm$ 4.11	58.85 $\pm$ 3.37	63.36 $\pm$ 2.34	49.88 $\pm$ 3.28	57.07 $\pm$ 1.57	42.53 $\pm$ 4.13	49.98 $\pm$ 3.20
PCR (CVPR'23)	66.38 $\pm$ 11.02	75.70 $\pm$ 3.87	62.31 $\pm$ 7.83	65.83 $\pm$ 3.01	54.33 $\pm$ 7.13	57.42 $\pm$ 4.25	47.16 $\pm$ 3.88	52.86 $\pm$ 1.73
ER-MKD (ICML'24)	71.54 $\pm$ 9.25	80.12 $\pm$ 3.51	63.09 $\pm$ 3.09	70.23 $\pm$ 2.65	57.21 $\pm$ 3.39	63.40 $\pm$ 3.60	47.34 $\pm$ 2.97	55.35 $\pm$ 1.91
FairOCL	77.43 $\pm$ 7.17	81.44 $\pm$ 3.40	62.82 $\pm$ 4.09	69.54 $\pm$ 2.74	59.14 $\pm$ 3.97	63.95 $\pm$ 2.63	52.94 $\pm$ 2.09	58.65 $\pm$ 0.85

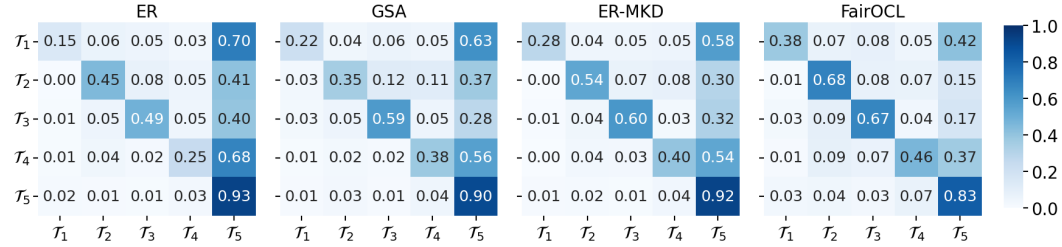


Figure 3: Task-based confusion matrix of various gradient-alignment-based OCL methods on the CIFAR-10 dataset with $|\mathcal{M}|=200$. See §5.3 for details.

achieves the highest average accuracy over past tasks ($\text{Avg}(\mathcal{T}_{[1:4]})$), demonstrating its ability to retain historical knowledge and avoid overfitting to the most recent task. In addition, Fig. 2 (b) shows the Average Forgetting (F) per task. FairOCL consistently exhibits lower forgetting across all tasks, with strong performance on $\text{Avg}(\mathcal{T}_{[1:4]})$, underscoring its ability to mitigate catastrophic forgetting and ensure preservation of earlier task knowledge.

Table 3 further details accuracy on past tasks ($\text{Avg}(\mathcal{T}_{[1:n-1]})$) and all tasks ($\text{Avg}(\mathcal{T}_{[1:n]})$) as the task sequence grows from $n = 2$ to $n = 5$. FairOCL consistently surpasses all baselines in past-task knowledge retention. For instance, at $n = 5$, it achieves 52.94% on past tasks and 58.65% overall, compared to ER-MKD (47.34% / 55.35%) and PCR (46.13% / 52.86%). This demonstrates that FairOCL’s fairness-driven aggregation directly translates into improved overall performance.

Task Interrelationships Analysis. Beyond average accuracy on past tasks, we also examine task interrelationships using the confusion matrices. An effective method should display strong diagonal values, indicating accurate performance on each task, and low off-diagonal entries, reflecting limited interference across tasks. As shown in Fig. 3, FairOCL achieves the most uniform and elevated diagonal values, particularly on earlier tasks $\mathcal{T}_1$ – $\mathcal{T}_4$, showing that it preserves knowledge more evenly throughout the task sequence. In contrast, ER, ER-MKD, and GSA all exhibit stronger bias toward the most recent task $\mathcal{T}_5$ while showing weaker retention of earlier tasks. Overall, the inter-task analysis confirms that FairOCL maintains more balanced knowledge preservation across tasks, mitigating forgetting while sustaining strong performance on current tasks.

5.4 ABLATION STUDY

Component Analysis. To evaluate the contribution of each component in FairOCL, we perform an ablation study on CIFAR-10 ($|\mathcal{M}| = 1000$), CIFAR-100 ($|\mathcal{M}| = 2000$), and TinyImageNet ($|\mathcal{M}| = 5000$). As shown in Table 4, both the fair gradient aggregation strategy (“Fair”) and the recency-bias mitigation via knowledge distillation (“RB-KD”) contribute substantially, but serve complementary purposes. Fairness alone delivers consistent improvements across datasets, confirming its role in balancing past-task contributions and alleviating gradient interference, an aspect that RB-KD cannot address. While RB-KD yields larger standalone improvements in some cases, it mainly stabilizes representations against drift and does not address imbalanced gradient updates. The best performance is consistently obtained when both components are combined, indicating that fairness provides the principled foundation for balanced continual learning, while RB-KD acts as an auxiliary stabilizer that further enhances retention. Together, they form a synergistic framework that surpasses either component in isolation and is essential to the full effectiveness of FairOCL.

Table 4: **Component analysis on FairOCL.** Here, “Fair” denotes the fair gradient aggregation strategy, and “RB-KD” denotes the task recency bias alleviation via knowledge distillation. Results show that both components contribute complementary benefits, and the best performance is consistently achieved when both are combined. See §5.4 for details.

Fair	RB-KD	CIFAR-10 ($ \mathcal{M} =1000$)		CIFAR-100 ($ \mathcal{M} =2000$)		TinyImageNet ($ \mathcal{M} =5000$)	
		Acc ($\uparrow$)	F ($\downarrow$)	Acc ($\uparrow$)	F ($\downarrow$)	Acc ($\uparrow$)	F ($\downarrow$)
$\times$	$\times$	56.84	38.88	31.21	23.14	19.64	29.55
$\checkmark$	$\times$	62.00	27.66	40.00	17.08	26.88	20.82
$\times$	$\checkmark$	67.95	20.73	44.78	11.21	33.16	16.05
$\checkmark$	$\checkmark$	68.44	19.38	46.43	9.02	34.51	14.08

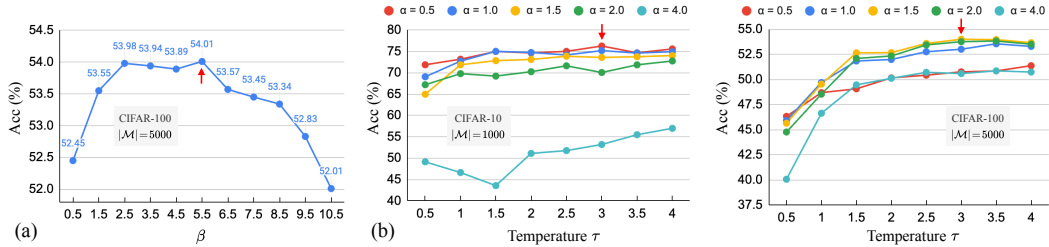


Figure 4: **Ablation study on hyperparameters.** (a) **Effect of loss weight β** on CIFAR-100 ($|\mathcal{M}|=5000$). (b) **Effect of fairness parameter α and temperature τ** on CIFAR-10 ($|\mathcal{M}|=1000$) and CIFAR-100 ($|\mathcal{M}|=5000$). FairOCL shows stable performance across a broad range of α values, indicating robustness to this parameter. See §5.4 for details.

Hyperparameters. We first determine the loss weight β in Eq. (9). Fig. 4 (a) reports the Average Accuracy (Acc) on CIFAR-100 with $|\mathcal{M}|=5000$ under different values of β . The best performance is obtained at $\beta=5.5$, which we adopt as the default for all experiments.

We then study the effects of the fairness parameter α and the distillation temperature τ . Fig. 4 (b) shows results on CIFAR-10 with $|\mathcal{M}|=1000$ (left) and CIFAR-100 with $|\mathcal{M}|=5000$ (right). On CIFAR-10, the best performance occurs at $\alpha=0.5$ and $\tau=3.0$, which we use for all CIFAR-10 settings. On CIFAR-100, the optimal configuration is $\alpha=1.5$, $\tau=3.0$, which also performs well on ImageNet-100 and TinyImageNet, and we apply it directly to these two datasets without further tuning. More broadly, Fig. 4 (b) shows that FairOCL is robust across a wide range of α values, with performance varying only gradually. Thus, α can be treated as a principled control parameter for fairness preferences rather than a fragile hyperparameter requiring extensive search.

6 CONCLUSION

This study targets a key limitation of current replay-based online continual learning (OCL) methods, namely, the gradient conflicts that arise when jointly training on mixed-task data, which hinder the retention of previously acquired knowledge. Our proposed FairOCL framework mitigates this challenge by formulating gradient aggregation as a constrained utility maximization problem inspired by fair resource allocation principles in communication networks. This formulation promotes fair and adaptive task-level gradient updates, thereby effectively alleviating interference among tasks. To further mitigate the task-recency bias caused by class imbalance, we incorporate a knowledge distillation mechanism that stabilizes representations of earlier tasks. Extensive experiments on standard OCL benchmarks show that FairOCL consistently outperforms state-of-the-art baselines, establishing it as a principled and effective solution for OCL.

Limitations and Future Work. While FairOCL shows competitive results in balancing stability and plasticity through gradient fairness, future work could explore mechanisms to automatically adapt the fairness parameter α during training. Another promising direction is to extend the fairness-based formulation to offline continual learning, where multiple passes over data are possible, to further assess its benefits under less restrictive conditions.

REFERENCES

- Hao Ban and Kaiyi Ji. Fair resource allocation in multi-task learning. In *ICML*, pp. 2715–2731, 2024.
- Stephen P Boyd and Lieven Vandenbergh. *Convex optimization*. Cambridge University Press, 2004.
- Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. *NeurIPS*, 33:15920–15930, 2020.
- Lucas Caccia, Rahaf Aljundi, Nader Asadi, Tinne Tuytelaars, Joelle Pineau, and Eugene Belilovsky. New insights on reducing abrupt representation change in online continual learning. In *ICLR*, 2022.
- Arslan Chaudhry, Marc’Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. In *ICLR*, 2019.
- Aristotelis Chrysakis and Marie-Francine Moens. Online bias correction for task-free continual learning. In *ICLR*, 2023.
- Danruo Deng, Guangyong Chen, Jianye Hao, Qiong Wang, and Pheng-Ann Heng. Flattening sharpness for dynamic gradient projection memory benefits continual learning. *NeurIPS*, 34:18710–18721, 2021.
- Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pp. 248–255, 2009.
- Yanan Gu, Xu Yang, Kun Wei, and Cheng Deng. Not just selection, but exploration: Online class-incremental continual learning via dual view consistency. In *CVPR*, pp. 7442–7451, 2022.
- Nuwan Gunasekara, Bernhard Pfahringer, Heitor Murilo Gomes, and Albert Bifet. Survey on online streaming continual learning. In *IJCAI*, pp. 6628–6637, 2023.
- Yiduo Guo, Bing Liu, and Dongyan Zhao. Online continual learning through mutual information maximization. In *ICML*, pp. 8109–8126, 2022.
- Yiduo Guo, Bing Liu, and Dongyan Zhao. Dealing with cross-task class discrimination in online continual learning. In *CVPR*, pp. 11878–11887, 2023.
- Yunhui Guo, Mingrui Liu, Tianbao Yang, and Tajana Rosing. Improved schemes for episodic memory-based lifelong learning. *NeurIPS*, 33:1023–1035, 2020.
- Gunshi Gupta, Karmesh Yadav, and Liam Paull. Look-ahead meta learning for continual learning. *NeurIPS*, 33:11588–11598, 2020.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pp. 770–778, 2016.
- Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.
- Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *CVPR*, pp. 831–839, 2019.
- Rajendra K Jain, Dah-Ming W Chiu, William R Hawe, et al. A quantitative measure of fairness and discrimination. *Eastern Research Laboratory, Digital Equipment Corporation, Hudson, MA*, 21 (1), 1984.
- Frank Kelly. Charging and rate control for elastic traffic. *ETT*, 8(1):33–37, 1997.
- Frank P Kelly, Aman K Maulloo, and David Kim Hong Tan. Rate control for communication networks: shadow prices, proportional fairness and stability. *JORS*, 49(3):237–252, 1998.

- Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.
- Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.
- Tian Lan, David Kao, Mung Chiang, and Ashutosh Sabharwal. *An axiomatic theory of fairness in network resource allocation*. IEEE, 2010.
- Yann Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.
- Tian Li, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith. Fair resource allocation in federated learning. In *ICLR*, 2020.
- Huiwei Lin, Baoquan Zhang, Shanshan Feng, Xutao Li, and Yunming Ye. Pcr: Proxy-based contrastive replay for online class-incremental continual learning. In *CVPR*, pp. 24246–24255, 2023.
- Sen Lin, Li Yang, Deliang Fan, and Junshan Zhang. Trgp: Trust region gradient projection for continual learning. In *ICLR*, 2022.
- David Lopez-Paz and Marc’Aurelio Ranzato. Gradient episodic memory for continual learning. *NeurIPS*, 30, 2017.
- Zheda Mai, Ruiwen Li, Jihwan Jeong, David Quispe, Hyunwoo Kim, and Scott Sanner. Online continual learning in image classification: An empirical survey. *Neurocomputing*, 469:28–51, 2022.
- Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of Learning and Motivation*, volume 24, pp. 109–165. 1989.
- Nicolas Michel, Maorong Wang, Ling Xiao, and Toshihiko Yamasaki. Rethinking momentum knowledge distillation in online continual learning. In *ICML*, pp. 35607–35622, 2024.
- Jeonghoon Mo and Jean Walrand. Fair end-to-end window-based congestion control. *IEEE/ACM TON*, 8(5):556–567, 2000.
- German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019.
- Bozidar Radunovic and Jean-Yves Le Boudec. A unified framework for max-min and min-max fairness with applications. *IEEE/ACM TON*, 15(5):1073–1083, 2007.
- 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. In *ICLR*, 2019.
- David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *NeurIPS*, 32, 2019.
- Gobinda Saha and Kaushik Roy. Continual learning with scaled gradient projection. In *AAAI*, volume 37, pp. 9677–9685, 2023.
- Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. In *ICLR*, 2021.
- Albin Soutif-Cormerais, Antonio Carta, Andrea Cossu, Julio Hurtado, Vincenzo Lomonaco, Joost Van de Weijer, and Hamed Hemati. A comprehensive empirical evaluation on online continual learning. In *ICCV*, pp. 3518–3528, 2023.
- Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *NeurIPS*, 30, 2017.
- Jeffrey S Vitter. Random sampling with a reservoir. *ACM TOMS*, 11(1):37–57, 1985.

- Quanzhang Wang, Renzhen Wang, Yichen Wu, Xixi Jia, and Deyu Meng. Cba: Improving online continual learning via continual bias adaptor. In *ICCV*, pp. 19082–19092, 2023.
- Yichen Wu, Hong Wang, Peilin Zhao, Yefeng Zheng, Ying Wei, and Long-Kai Huang. Mitigating catastrophic forgetting in online continual learning by modeling previous task interrelations via pareto optimization. In *ICML*, 2024.
- Jason Yoo, Yunpeng Liu, Frank Wood, and Geoff Pleiss. Layerwise proximal replay: a proximal point method for online continual learning. In *ICML*, pp. 57199–57216, 2024.
- Guanxiong Zeng, Yang Chen, Bo Cui, and Shan Yu. Continual learning of context-dependent processing in neural networks. *Nature Machine Intelligence*, 1(8):364–372, 2019.
- Guojun Zhang, Saber Malekmohammadi, Xi Chen, and Yaoliang Yu. Proportional fairness in federated learning. *TMLR*, 2022.

A APPENDIX

In this section, we provide additional information to support the main content of the paper. §A.1 documents our use of large language models (LLMs). §A.2 presents a toy multi-objective example to illustrate the effect of different fairness parameters. §A.3 offers a detailed derivation of Eq. (6). §A.4 reports the computational efficiency of FairOCL compared to other baseline methods. Finally, §A.5 analyzes the effect of class coverage in replay batches.

A.1 USE OF LARGE LANGUAGE MODELS (LLMs)

We used LLMs only for polishing the text, specifically, improving grammar, wording, clarity, and fluency. We did not use LLMs for generating core scientific content, defining the methodology, deriving proofs, designing experiments, interpreting results, or any other technical components. We take full responsibility for correctness, factual claims, and attribution.

A.2 ILLUSTRATION OF DIFFERENT FAIRNESS PARAMETERS

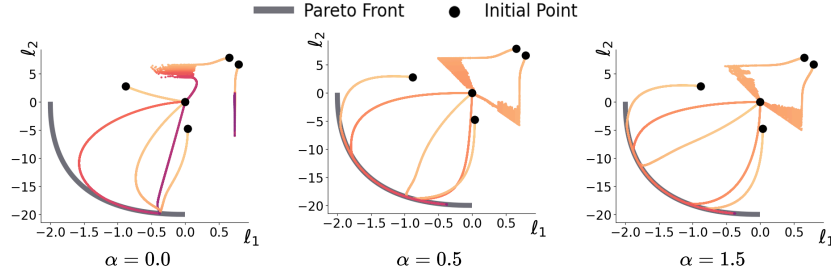


Figure 5: Illustrative of convergence for a two-objective optimization example. Here shows trajectories (color gradient from orange to purple) from five distinct initializations (black dots •) under varying fairness parameters $\alpha \in \{0.0, 0.5, 1.5\}$. The Pareto front is highlighted, demonstrating how α affects the trade-off between objectives L_1 and L_2 . See §A.2 for details.

To demonstrate the effect of the fairness parameter α , we adopt a two-objective optimization problem from (Ban & Ji, 2024), with two optimization objectives $L_1(\mathbf{x})$ and $L_2(\mathbf{x})$ are defined for $\mathbf{x} = (x_1, x_2)^\top \in \mathbb{R}^2$:

$$L_1(\mathbf{x}) = 0.1 \cdot (f_1(\mathbf{x})g_1(\mathbf{x}) + f_2(\mathbf{x})h_1(\mathbf{x})), \quad (10)$$

$$L_2(\mathbf{x}) = f_1(\mathbf{x})g_2(\mathbf{x}) + f_2(\mathbf{x})h_2(\mathbf{x}), \quad (11)$$

where the component functions are given by:

$$f_1(\mathbf{x}) = \max(\tanh(0.5x_2), 0), \quad (12)$$

$$f_2(\mathbf{x}) = \max(\tanh(-0.5x_2), 0), \quad (13)$$

$$g_1(\mathbf{x}) = \log(\max(|0.5(-x_1 - 7) - \tanh(-x_2)|, 0.000005)) + 6, \quad (14)$$

$$g_2(\mathbf{x}) = \log(\max(|0.5(-x_1 + 3) - \tanh(-x_2) + 2|, 0.000005)) + 6, \quad (15)$$

$$h_1(\mathbf{x}) = ((-x_1 + 7)^2 + 0.1(-x_1 - 8)^2)/10 - 20, \quad (16)$$

$$h_2(\mathbf{x}) = ((-x_1 - 7)^2 + 0.1(-x_1 - 8)^2)/10 - 20. \quad (17)$$

The gradient magnitude disparity ($\|L_2(\mathbf{x})\| > \|L_1(\mathbf{x})\|$) poses challenges for multi-objective balancing. We investigate five initial points $\{(-8.5, 7.5), (0, 0), (9.0, 9.0), (-7.5, -0.5), (9.0, -1.0)\}$ using Adam optimizer with a learning rate of 1×10^{-3} . The training process stops when the Pareto front is reached. Fig. 5 shows the optimization trajectories under different fairness regimes. When $\alpha = 0$ (i.e., without fairness), the algorithm may fail to converge to a Pareto stationary point. In contrast, settings with fairness ($\alpha > 0$) lead to convergence to the Pareto front, with the degree of fairness shaped by the choice of α . Specifically, $\alpha = 0.5$ is used in our CIFAR-10 experiments, while $\alpha = 1.5$ is adopted for CIFAR-100, TinyImageNet and ImageNet-100. These results demonstrate that incorporating fairness not only ensures convergence but also allows for flexible control over fairness behavior across tasks.

A.3 DERIVATION OF EQ. (6)

We first show that the optimal solution lies on the boundary of the Euclidean ball B_ϵ , *i.e.*, $\|\mathbf{d}\|_2 = \epsilon$. This follows from the fact that the objective is monotonically increasing in the norm of $\mathbf{d}$, under the constraint $\mathbf{g}_i^\top \mathbf{d} \geq 0$ for all i . Specifically:

- The objective in Eq. (5) is a sum of terms of the form $\frac{(\mathbf{g}_i^\top \mathbf{d})^{1-\alpha}}{1-\alpha}$.
- If $\alpha < 1$, the function $(\mathbf{g}_i^\top \mathbf{d})^{1-\alpha}$ is a monotonically increasing in $\mathbf{g}_i^\top \mathbf{d}$ (since the exponent $1-\alpha > 0$).
- If $\alpha > 1$, the term $\frac{(\mathbf{g}_i^\top \mathbf{d})^{1-\alpha}}{1-\alpha}$ can be rewritten as $-\frac{(\mathbf{g}_i^\top \mathbf{d})^{1-\alpha}}{\alpha-1}$, which is also a monotonically increasing function of $\mathbf{g}_i^\top \mathbf{d}$ (since the exponent $1-\alpha < 0$, but the negative sign flips the monotonicity).

Since $\mathbf{g}_i^\top \mathbf{d}$ is non-decreasing with $\|\mathbf{d}\|_2$ (due to the constraint), the objective is non-decreasing in $\mathbf{d}$, and is therefore maximized when $\|\mathbf{d}\|_2 = \epsilon$.

Next, to derive Eq. (6), we apply the Karush-Kuhn-Tucker (KKT) conditions (Boyd & Vandenberghe, 2004) to the constrained optimization in Eq. (5). The Lagrangian is:

$$\mathcal{L}(\mathbf{d}, c, \{\mu_i\}) = \sum_{i=1}^{n-1} \frac{(\mathbf{g}_i^\top \mathbf{d})^{1-\alpha}}{1-\alpha} - \frac{c}{2}(\mathbf{d}^\top \mathbf{d} - \epsilon^2) + \sum_{i=1}^{n-1} \mu_i (\mathbf{g}_i^\top \mathbf{d}). \quad (18)$$

The KKT conditions yield:

- **Stationarity:** $\nabla_{\mathbf{d}} \mathcal{L} = \sum_{i=1}^{n-1} (\mathbf{g}_i^\top \mathbf{d})^{-\alpha} \mathbf{g}_i - c\mathbf{d} + \sum_{i=1}^{n-1} \mu_i \mathbf{g}_i = 0$.
Rearranging gives: $\sum_{i=1}^{n-1} [(\mathbf{g}_i^\top \mathbf{d})^{-\alpha} + \mu_i] \mathbf{g}_i = c\mathbf{d}$.
- **Primal Feasibility:** $\|\mathbf{d}\|_2^2 \leq \epsilon^2$ and $\mathbf{g}_i^\top \mathbf{d} \geq 0$.
- **Dual Feasibility:** $c \geq 0, \mu_i \geq 0$.
- **Complementary Slackness:** $c(\|\mathbf{d}\|_2^2 - \epsilon^2) = 0, \mu_i (\mathbf{g}_i^\top \mathbf{d}) = 0, \forall i$.

At the optimum, $\|\mathbf{d}\|_2 = \epsilon$, implying $c > 0$.

Since $\mathbf{g}_i^\top \mathbf{d} \geq 0$, we get $\mu_i = 0$ for all i .

Substituting into the stationarity condition gives: $\sum_{i=1}^{n-1} ((\mathbf{g}_i^\top \mathbf{d})^{-\alpha}) \mathbf{g}_i = c\mathbf{d}, c > 0$, which is Eq. (6).

A.4 COMPUTATIONAL EFFICIENCY

To assess the computational efficiency of FairOCL, we compare its training time with five replay-based OCL methods, namely, DVC (Gu et al., 2022), GSA (Guo et al., 2023), PCR (Lin et al., 2023), POCL (Wu et al., 2024), and ER-MKD (Michel et al., 2024). All experiments are conducted on CIFAR-10 with a buffer size of $\mathcal{M} = 200$. As shown in Table 5, FairOCL achieves the highest average accuracy (58.65%) while maintaining competitive training time (545.2s). Notably, FairOCL is 3× faster than the POCL (1628.2s), which relies on computationally intensive hyper-gradient optimization. Moreover, FairOCL exhibits similar or better efficiency compared to DVC (552.2s) and PCR (547.4s), despite outperforming them in terms of accuracy. These results show that FairOCL delivers strong performance without incurring additional computational overhead.

Table 5: Comparison of training time and average accuracy for different OCL methods on CIFAR-10 ($\mathcal{M} = 200$). See §A.4 for details.

Method	DVC	GSA	PCR	POCL	ER-MKD	FairOCL
Average Accuracy (% , $\uparrow$)	48.08	48.90	52.20	35.69	57.54	58.65
Training Time (s)	552.2	400.6	547.4	1628.2	355.4	545.2

A.5 EFFECT OF CLASS COVERAGE

FairOCL does not explicitly enforce the presence of all previously seen classes in each replay batch. To evaluate whether such coverage improves performance, we conducted experiments on CIFAR-10. As shown in Table 6, enforcing class coverage did not lead to noticeable accuracy gains, but in our cases, slightly degraded performance. This may be because enforcing such class coverage can disrupt the natural sampling distribution maintained by reservoir sampling. More importantly, our method focuses on mitigating *task-level* interference. Although individual classes may be temporarily absent from a batch, they are unlikely to remain absent across many consecutive batches, and it is also rare for all classes from a task to be absent simultaneously. Thus, FairOCL can still compute meaningful task-level gradients and apply fairness-aware updates effectively. Over time, most classes can be sampled with sufficient frequency to support stable learning dynamics. Experimental results in Fig. 2 confirm that FairOCL can mitigate task interference effectively without requiring strict class coverage in each batch compared with other methods.

Table 6: Effect of class coverage. Here presents the Average Accuracy (Acc, %, $\uparrow$) on CIFAR-10 with different buffer sizes $|\mathcal{M}|$. See §A.5 for details.

Class Coverage?	$ \mathcal{M} = 200$	$ \mathcal{M} = 500$	$ \mathcal{M} = 1000$
Yes	57.41	69.43	74.12
No	59.22	69.94	76.26