CLIF: WHEN CONTINUAL LEARNING MEETS INFLUENCE-BASED FORGETTING

Anonymous authors
Paper under double-blind review

ABSTRACT

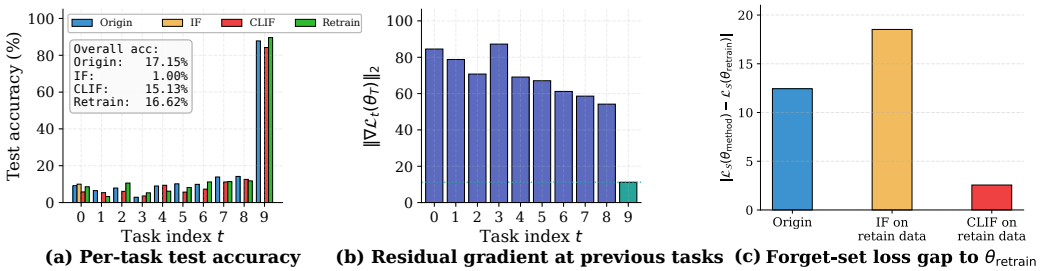
Machine unlearning in continual learning (CL) addresses the growing need to remove parts of a CL model's accumulated knowledge in response to privacy regulations, biases, or noise. Influence functions (IFs) offer an effective, retraining-free route to unlearning in the standard single-task setting, but we show that simply applying IFs to CL models leads to common failure. We identify two core challenges (Cs) as the root causes: **(C1) Non-stationarity**. Sequential training drives the final CL model far from the stationary point of the joint full-history loss that IFs assume. This leaves a residual gradient that standard IFs ignore, biasing the unlearning target. **(C2) Curvature unreliability**. IFs rely on the inverse of Hessian to convert removal gradients into parameter updates. But in CL, Hessian is estimated from small, distributionally skewed available data, yielding ill-conditioned inversions that corrupt parameter updates. To overcome both challenges, we propose **CLIF** (**C**ontinual-**L**earning with **I**nfluence-based **F**orgetting), a novel framework—the first, to our knowledge, to rehabilitate IF-based unlearning for replay-based CL—via three complementary phases. Phase I addresses C1 by introducing a bias-corrected influence target that incorporates residual gradients overlooked by standard IFs. Phase II addresses C2 via a devised Continual Fisher Information preconditioner whose variance-form Fisher provides a full-rank, stably invertible alternative to standard Hessian inverse. Phase III further proposes a forget-subspace projection that confines updates to forget-relevant directions while removing retain-aligned components to reduce performance degradation. Extensive experiments across two unlearning scenarios show that CLIF matches the gold-standard retraining behaviour more closely than standard IFs and state-of-the-art baselines, delivering effective unlearning while retaining strong performance on remaining tasks. Source code and data are available at `https://anonymous.4open.science/r/CLIF-DD56`.

1 INTRODUCTION

Continual learning (CL) (Wang et al., 2024a; Tong et al., 2025; Ma'sum et al., 2025) enables models to acquire knowledge incrementally from a stream of tasks, and is increasingly adopted in real-world applications (e.g., autonomous driving (Yang et al., 2025a) and healthcare (Amrollahi et al., 2022)) where data arrives sequentially and storage of full training histories is impractical (Shi et al., 2026; Yang et al., 2025b). Among the dominant paradigms, replay-based methods (Tong et al., 2025; Cho et al., 2025; Liu et al., 2026), which maintain a small buffer of exemplars from previous tasks, have emerged as particularly effective at mitigating catastrophic forgetting while remaining computationally tractable. In deployed CL systems, certain training samples may later need to be removed to obey privacy regulations (Protection, 2018; Wang et al., 2024b; Shaik et al., 2025), to mitigate biases introduced by prejudicial training data (Chen et al., 2023; Li et al., 2026a), or to improve robustness by excising adversarial, poisoned, or corrupted samples discovered post-deployment (Sheng et al., 2024; Han et al., 2025). These scenarios call for *machine unlearning* (Wang et al., 2024b; Shaik et al., 2025): efficiently updating the model to behave as if the targeted data had never been trained, without the prohibitive cost of retraining the entire CL model from scratch.

Approximate unlearning methods (Guo et al., 2020; Nguyen et al., 2020) offer a practical alternative by computing parameter corrections that approximate the effect of retraining. Among these, influence functions (IFs) (Koh & Liang, 2017; Bae et al., 2022; Chen et al., 2023; Li et al., 2026a; Rao et al.,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Per-task test accuracy **(b) Residual gradient at previous tasks** **(c) Forget-set loss gap to θ_{retrain}**

Figure 1: Observations on the CIFAR-100 dataset[1]. (a) We compare the test accuracy of the original CL model (Tong et al., 2025) (Origin) and unlearning baselines: the standard IF and our proposed CLIF using accessible retain data (buffer and current task training data), and the gold-standard Retrain model. (b) The gradient norm of loss at the final CL model is significantly larger for historical tasks than for the last task, confirming the presence of a residual gradient that violates the stationarity assumption. (c) The loss gap with the retrained model is even larger for the standard IF update computed using only the retain set than Origin, while CLIF closely matches the retrained model.

2025) are a representative and widely studied family that provides closed-form parameter updates based on first-order approximations to the leave-one-out or leave-k-out retraining objectives (Koh & Liang, 2017; Koh et al., 2019; Izzo et al., 2021; Li et al., 2026b). In the single-task setting, this amounts to a single Newton-step correction that is both computationally efficient and theoretically grounded. This has made IFs a natural backbone for retraining-free unlearning methods (Zhang et al., 2025; Chen et al., 2023; Li et al., 2026a; Wu et al., 2023).

This triggers a natural question, *can influence functions be directly applied to the CL setting to achieve the same effective unlearning as in the single-task setting?* We investigated this question on CIFAR-100 (10 tasks) with the goal of unlearning class 0 from a backbone CL model (Tong et al., 2025), and yielded a striking negative result: *standard IF-based unlearning not only underperforms, but also catastrophically destroys the model.* As shown in Figure 1 (a), the IF-updated model retains only 1% overall test accuracy, with retain-set accuracy (tasks 1-9) collapsing to 0%, which is supposed to be preserved. We attribute such a collapse to two fundamental incompatibilities between the underlying assumptions behind influence functions and the realities of continual learning, which correspondingly lead to two key challenges to be addressed.

Challenge 1 (C1): Non-stationarity. The standard IF derivation assumes the trained model as if it sits near a minimum of its training loss. This framing is reasonable for a classical single-task setting, where the model is optimised jointly on all training data. In CL, however, the model is produced by a sequence of optimisations, each targeting a different objective with a different set of training data, leaving a large *residual gradient* that severely misleads the unlearning target. Figure 1 (b) empirically confirms this: the gradient norms on historical tasks (blue bars) are several times larger than on the current task (the green bar), revealing that the model is far from a jointly optimal state.

Challenge 2 (C2): Curvature unreliability from the available data. Even if the influence target were corrected, the IF still requires inverting the Hessian of the training loss. In standard machine unlearning, this Hessian is estimated over the full training dataset, providing a reliable characterisation of the local curvature. However, full historical data is unavailable in CL; only a small replay buffer and the current task's data are accessible. This limited data is insufficient and distributionally skewed, yielding an ill-conditioned curvature estimate. The impact is evidenced in Figure 1(c): the resulting IF update leaves the forget-set loss away from that of the oracle retrained model, even farther than the original model, whereas our CLIF closely matches the retrained model.

Together, C1 and C2 explain the collapse observed in Figure 1 (a): IF computes the wrong direction (due to C1) and scales it by an unstable, noise-amplified factor (due to C2), yielding a massive parameter perturbation in a direction irrelevant to genuine retraining. This results in a degenerate network that, rather than unlearning the target class, has instead unlearned the entire data distribution.

To address these challenges, we propose **CLIF** (**C**ontinual-**L**earning with **I**nfluence-based **F**orgetting), a novel framework that rehabilitates IF-based unlearning for replay-based CL through three com-

[1]Detailed experimental settings are provided in Section 5.1 and Appendix E.1.

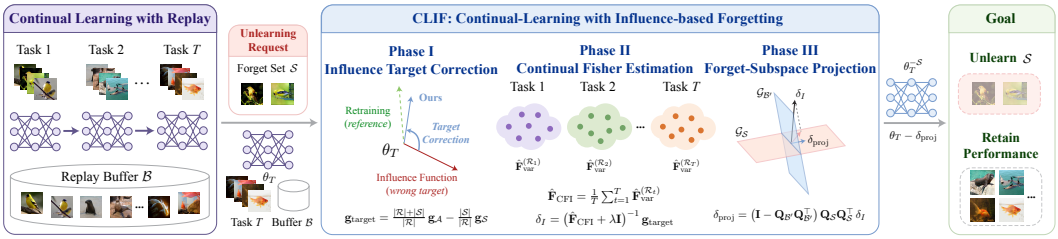


Figure 2: Illustration of CLIF. It first corrects the influence target with the residual gradient. Then, Continual Fisher Information is computed as a stable Hessian approximation. Finally, it confines updates with forget- and retain-space projection to reduce performance degradation.

plementary phases, as illustrated in Figure 2. To address C1, **Phase I** introduces a bias-corrected influence target that explicitly incorporates the residual gradient induced by CL non-stationarity. To address C2, **Phase II** designs a Continual Fisher Information preconditioner to avoid the unreliable Hessian inverse. It removes residual-gradient-induced inflation and balances curvature estimates across tasks, leading to stable per-parameter scaling under limited available data. **Phase III** is devised with a forget-subspace projection to refine the resulting update. By retaining only directions supported by the forget samples and filtering out directions shared with the retain buffer, it localises the parameter perturbation to forgetting-relevant components and reduces performance degradation on previously learned tasks. Remarkably, Figure 1 (c) shows that CLIF, using only the limited retain data, achieves even closer alignment with the oracle retrained model than the standard IF. Our main contributions are as follows:

- We provide the first systematic analysis of why influence-function-based unlearning fails in continual learning, identifying non-stationarity and curvature unreliability as the two root causes for failure induced by the specialised characteristics of the CL setting.
- We propose CLIF, an innovative three-phase framework that adapts influence functions to the replay-based CL regime by correcting the influence target (Phase I), stabilising curvature estimation (Phase II), and constraining the update to forget-relevant subspaces (Phase III).
- We provide theoretical analysis bounding the approximation error of CLIF relative to the gold-standard retrained model under CL-specific constraints with interpretable terms, including buffer proxy gap, retrain residual, curvature estimation, and projection error.
- Extensive experiments across two unlearning scenarios on three widely-used benchmarks demonstrate that CLIF achieves effective unlearning while retaining strong performance on remaining tasks, outperforming representative and state-of-the-art unlearning baselines.

2 Related Work

Continual learning aims to update models incrementally as new tasks arrive without catastrophically forgetting previously learned knowledge (Ke et al., 2021; Lin et al., 2023). Existing approaches broadly fall into three families. **(1) Regularisation-based methods** (Kirkpatrick et al., 2017; Benzing, 2022) add penalty terms that discourage changes to parameters deemed important for previous tasks using different measures. **(2) Replay-based methods** (Riemer et al., 2019a; Rebuffi et al., 2017; Tong et al., 2025) maintain a small buffer of exemplars from previous tasks and interleave them with current-task data during training. **(3) Architecture-based methods** (Gurbuz & Dovrolis, 2022) allocate separate parameters or sub-networks per task (Wortsman et al., 2020). In this work, we focus on widely adopted replay-based methods, as their finite and constrained data availability for historical tasks provides a practical yet challenging testbed for continual learning and unlearning.

Machine unlearning tackles the problem of efficiently removing the influence of specific training data from a trained model (Wang et al., 2024b; Shaik et al., 2025). We organise existing work by the unlearning guarantee. **(1) Exact unlearning** (Cao & Yang, 2015; Bourtoule et al., 2021) ensures the unlearned model is statistically indistinguishable from the retrained one. While principled, they need a splittable training pipeline that is incompatible with the sequential, non-partitioned nature of CL. **(2) Approximate unlearning** (Guo et al., 2020; Li et al., 2026a) seeks efficient parameter updates that

approximate the retrained model within controlled error bounds. While elegant, influence functions have been shown to be fragile in deep learning (Basu et al., 2021), and some variants (Schioppa et al., 2022; Kwon et al., 2024) have been proposed to improve their robustness and scalability. Yet, these methods rely on access to the original training data for gradient and Hessian computation, an assumption incompatible with CL that leads to the failure modes we identify in this work.

Unlearning in CL hasn't yet received enough attention, with only limited related works (Cao et al., 2025; Liu et al., 2022). While there are some works on continual unlearning Wuerkaixi et al. (2025); Gao et al. (2025); Lee et al. (2025a); Shen et al. (2025), they aim to address the problem of sequential unlearning (Xu et al., 2026), i.e., multiple conventional unlearning requests, instead of the unlearning for CL models. Thus, they fall outside the scope of this work. To our knowledge, there is a lack of work studying the unique challenges of influence-based unlearning in the CL scenarios and providing an effective and practical solution. See Appendix G for a detailed review of the all related literature.

3 PRELIMINARIES

Replay-Based Continual Learning. We consider a classic CL setting where a model with parameters $\boldsymbol{\theta}$ is trained sequentially on T tasks with datasets $\mathcal{D}_1, \cdots, \mathcal{D}_T$. At stage t, training starts from the previous model $\boldsymbol{\theta}_{t-1}$ and updates it using the current task data $\mathcal{D}_t$ together with a replay buffer $\mathcal{B}_t$ containing a small subset of exemplars from previous tasks:

$$\boldsymbol{\theta}_t = \mathcal{C}_t\Big(\boldsymbol{\theta}_{t-1}; \widetilde{\mathcal{L}}_t\Big), \qquad \widetilde{\mathcal{L}}_t(\boldsymbol{\theta}) = \alpha_t \mathcal{L}_{\mathcal{D}_t}(\boldsymbol{\theta}) + (1-\alpha_t)\mathcal{L}_{\mathcal{B}_t}(\boldsymbol{\theta}), \tag{1}$$

where $\mathcal{C}_t$ denotes the sequential optimiser used by the CL baseline, and α_t reflects the replay sampling ratio or loss weighting. The empirical loss over a dataset $\mathcal{A}$ is $\mathcal{L}_{\mathcal{A}}(\boldsymbol{\theta}) = \frac{1}{|\mathcal{A}|}\sum_{z\in\mathcal{A}} \mathcal{L}(z, \boldsymbol{\theta})$. The full training data is $\mathcal{D} = \bigcup_{t=1}^{T} \mathcal{D}_t$ with $|\mathcal{D}| = n$. The buffer $\mathcal{B}_T$ at stage T satisfies $|\mathcal{B}_T| \ll n$.

Unlearning in Continual Learning. Given the trained CL model $\boldsymbol{\theta}_T \in \mathbb{R}^p$, the replay buffer $\mathcal{B}$, current-task data $\mathcal{D}_T$, and an unlearning request specifying a forget set $\mathcal{S} = \{z_1, \ldots, z_m\} \subset \mathcal{D}$, the goal is to produce an unlearned model $\boldsymbol{\theta}_T^{-\mathcal{S}}$ that approximates the *gold-standard* retrained model $\boldsymbol{\theta}^{\mathrm{rt}}$, i.e., the model that would have been obtained by re-running the entire CL pipeline from scratch with the *ideal retain set* $\mathcal{D} \setminus \mathcal{S}$.

Beyond the data from CL training ($\mathcal{D}_T$ and $\mathcal{B}$), the unlearning request itself supplies $\mathcal{S}$ to the algorithm, forming a GDPR-style erasure request identifies samples (or a class whose exemplars the requester provides). The full accessible data is therefore $\mathcal{A} = \mathcal{D}_T \cup \mathcal{B} \cup \mathcal{S}$. We further partition $\mathcal{A}$ into two disjoint components that play distinct roles in the method: $\mathcal{A} = \mathcal{R} \sqcup \mathcal{S}$, where $\mathcal{R} := (\mathcal{D}_T \cup \mathcal{B}) \setminus \mathcal{S}$ is the *accessible retain set* $\mathcal{R}$. The ideal retain set $\mathcal{D} \setminus \mathcal{S}$ remains unavailable in CL because past-task data outside the buffer has been discarded. The formal unlearning objective seeks a parameter perturbation $\boldsymbol{\delta}$ that minimises the distance to the retrained model subject to retaining performance:

$$\min_{\boldsymbol{\delta}} \|\boldsymbol{\theta}_T - \boldsymbol{\delta} - \boldsymbol{\theta}^{\mathrm{rt}}\|_2^2 \qquad \text{s.t.} \quad \mathcal{L}_{\mathcal{R}}(\boldsymbol{\theta}_T - \boldsymbol{\delta}) \le \mathcal{L}_{\mathcal{R}}(\boldsymbol{\theta}_T) + \epsilon. \tag{2}$$

Standard Influence Functions. In the classical Empirical Risk Minimisation (ERM) under a single-task setting, the model is trained to converge on a fixed dataset $\mathcal{D}$: $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta})$. The influence of removing a sample z on the optimal parameters is derived via implicit differentiation of the first-order optimality condition (Koh & Liang, 2017; Cook & Weisberg, 1982):

$$\boldsymbol{\theta}^* - \boldsymbol{\theta}^*_{-z} \approx -\frac{1}{n}\mathbf{H}_{\boldsymbol{\theta}^*}^{-1} \nabla_{\boldsymbol{\theta}} \mathcal{L}(z, \boldsymbol{\theta}^*), \tag{3}$$

where $\mathbf{H}_{\boldsymbol{\theta}^*} = \nabla^2_{\boldsymbol{\theta}} \mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta}^*)$ is the Hessian of the empirical risk. This derivation rests on two key assumptions:

Definition 3.1 (Standard IF Assumptions)**. (A1) Stationarity:** The model resides at a (local) minimum of the total training loss with $\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta}^*) = \mathbf{0}$. **(A2) Regularity:** The Hessian is positive definite $\mathbf{H}_{\boldsymbol{\theta}^*} \succ 0$ and the condition number $\kappa(\mathbf{H}_{\boldsymbol{\theta}^*})$ is moderate.

3.1 MOTIVATION: WHY STANDARD IFS COLLAPSE IN CL

Applying the IF derivation to a CL model $\boldsymbol{\theta}_T$ implicitly presumes that $\boldsymbol{\theta}_T$ can be treated as a minimiser of the full-history loss $\mathcal{L}_{\mathcal{D}_{1:T}}$ whose sample removal is being approximated. This presumption fails in two compounding ways, leading to two key challenges to be addressed.

Challenge 1 (C1): Non-stationarity. The standard IF update is the Newton step that connects one stationary point ($\boldsymbol{\theta}^* = \arg\min \mathcal{L}_{\mathcal{D}}$) to a nearby stationary point ($\boldsymbol{\theta}^{*,-\mathcal{S}} = \arg\min \mathcal{L}_{\mathcal{D}\setminus\mathcal{S}}$). The derivation is valid when $\boldsymbol{\theta}^*$ sits near a minimum of $\mathcal{L}_{\mathcal{D}}$. In CL, $\boldsymbol{\theta}_T$ is the endpoint of sequential optimisation on a *different objective at each stage*; at no point during training is the joint loss $\mathcal{L}_{\mathcal{D}_{1:T}}$ optimised directly. The classical premise, therefore, does not stand. Consequently, the Newton step $\mathbf{H}^{-1}\mathbf{g}_{\mathcal{S}}$, where $\mathbf{g}_{\mathcal{S}} = \nabla_\theta \mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta}_T)$, starts from a non-stationary point and heads toward a target that drifts away from the gold-standard retrained model. Figure 1 (b) shows per-task gradient norms at $\boldsymbol{\theta}_T$ ranging from 54.2 to 87.3 on historical tasks, versus 11.2 on the current task. Each historical task's gradient norm is a witness that $\boldsymbol{\theta}_T$ is not at a joint minimum of $\mathcal{L}_{\mathcal{D}_{1:T}}$.

Challenge 2 (C2): Poor curvature estimation. Even granted a correct influence target, the IF update requires inverting a Hessian $\mathbf{H}_{\mathcal{R}}$ estimated from the accessible retain set. With limited and task-skewed accessible data, this estimate can poorly represent the curvature of the ideal retain objective, and regularisation-based strategies cannot recover missing or distorted curvature information. Consequently, the resulting inverse-curvature scaling can still produce large and misaligned parameter updates, potentially damaging the model.

4 Continual Learning with Influence-based Forgetting

We now present CLIF, a three-phase framework for influence-based unlearning in replay-based CL, as illustrated in Figure 2. Each phase targets a specific aspect of the challenges identified in Section 3.1. Phase I corrects the influence target for non-stationarity (C1), Phase II stabilises the curvature estimate with a CL-specialised Fisher preconditioner (C2), and Phase III further constrains the resulting update from Phases I and II to forget-relevant directions, guarding retained knowledge against cross-task interference inherent to shared-representation CL models.

4.1 Phase I: Influence Target Correction

Phase I restructures the influence update to alleviate the non-stationarity challenge in CL. Since $\mathcal{D} \setminus \mathcal{S}$ is unavailable at unlearning time, CLIF derives the perturbation directly as a Newton step on the accessible retain loss $\mathcal{L}_{\mathcal{R}}$. A second-order Taylor expansion of $\mathcal{L}_{\mathcal{R}}$ around $\boldsymbol{\theta}_T$ and first-order optimality in $\boldsymbol{\delta}$ yield the condition $\mathbf{H}_{\mathcal{R}}\boldsymbol{\delta} = \mathbf{g}_{\text{target}}$. We introduce Proposition 4.1 (See Appendix C.1 for the detailed proof) to organise it in terms of the residual gradient $\mathbf{g}_{\mathcal{A}}$ that standard IF is missing and the forget gradient $\mathbf{g}_{\mathcal{S}}$. In the stationary ERM setting ($\mathbf{g}_{\mathcal{A}} = \mathbf{0}$), only the sample-removal term survives and Eq. equation 4 turns to the classical leave-$|\mathcal{S}|$-out (Koh & Liang, 2017).

Proposition 4.1 (Bias-Corrected Influence Target)**.** *Let $\boldsymbol{\theta}_T$ be the CL model, $\mathcal{S}$ the forget set, and $\mathcal{R}$ the accessible retain set. Define $\mathbf{g}_{\mathcal{A}} := \nabla_{\boldsymbol{\theta}}\mathcal{L}_{\mathcal{A}}(\boldsymbol{\theta}_T)$, $\mathbf{g}_{\mathcal{S}} := \nabla_{\boldsymbol{\theta}}\mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta}_T)$, and $\mathbf{H}_{\mathcal{R}} := \nabla^2_{\boldsymbol{\theta}}\mathcal{L}_{\mathcal{R}}(\boldsymbol{\theta}_T)$. The stationary condition of the second-order quadratic approximation of $\mathcal{L}_{\mathcal{R}}(\boldsymbol{\theta}_T - \boldsymbol{\delta})$ is*

$$\mathbf{H}_{\mathcal{R}}\boldsymbol{\delta} = \mathbf{g}_{\text{target}}, \qquad \mathbf{g}_{\text{target}} = \frac{|\mathcal{R}| + |\mathcal{S}|}{|\mathcal{R}|}\mathbf{g}_{\mathcal{A}} - \frac{|\mathcal{S}|}{|\mathcal{R}|}\mathbf{g}_{\mathcal{S}} = \mathbf{g}_{\mathcal{R}}. \tag{4}$$

Building upon this, the parameter update can be formed as $\boldsymbol{\delta} = \mathbf{H}_{\mathcal{R}}^{-1}\mathbf{g}_{\text{target}}$. However, under C2, the standard Hessian inverse is rank-deficient and noise-amplifying. To this end, Phase II is devised with a CL-specialised Fisher preconditioner as a stable alternative to $\mathbf{H}_{\mathcal{R}}^{-1}$.

4.2 Phase II: Continual Fisher Estimation

Directly solving $\mathbf{H}_{\mathcal{R}}\boldsymbol{\delta} = \mathbf{g}_{\text{target}}$ is intractable under C2 and regularisation strategies (Alecsa & László, 2021; Ledoit & Wolf, 2022; Kumar & Shneider, 2016) remain unreliable in CL. With $|\mathcal{R}| \ll p$, the estimated Hessian can be rank-deficient, and its non-zero eigenvalues have high variance, amplifying noise in inversion. Thus, we seek a curvature approximation that is (i) full-rank and well-conditioned, (ii) estimable from $\mathcal{R}$, and (iii) cheap to invert.

Inspired by the Fisher information matrix (Teso et al., 2021; Lehmann & Casella, 1998), we propose the *Continual Fisher Information* (CFI) $\hat{\mathbf{F}}_{\text{CFI}}$, a diagonal preconditioner that satisfies all three criteria while incorporating CL adaptations.

Variance-form Fisher. The empirical diagonal Fisher $[\hat{\mathbf{F}}_{\text{diag}}]_j = \mathbb{E}_{z\in\mathcal{R}}[g_j(z)^2]$ is commonly used as a tractable curvature proxy (Kunstner et al., 2019), where the Fisher coincides with the Hessian under

the model distribution, and can be interpreted as the covariance of gradients when $\mathbb{E}[g_j] = 0$. In CL, as we identified in Phase I, the empirical mean gradient $\mathbb{E}_{z\in\mathcal{R}}[g_j(z)] = [\mathbf{g}_\mathcal{R}]_j$ is generally non-zero. As a result, the empirical Fisher decomposes as $[\hat{\mathbf{F}}_{\text{diag}}]_j = \text{Var}_{z\in\mathcal{R}}[g_j(z)] + \big(\mathbb{E}_{z\in\mathcal{R}}[g_j(z)]\big)^2$, where the second moment contains an additional squared empirical-mean term. We therefore use the within-task variance as a Fisher-inspired scaling statistic:

$$[\hat{\mathbf{F}}_{\text{var}}]_j = \mathbb{E}_{z\in\mathcal{R}}[g_j(z)^2] - \big(\mathbb{E}_{z\in\mathcal{R}}[g_j(z)]\big)^2 = \text{Var}_{z\in\mathcal{R}}[g_j(z)], \tag{5}$$

which removes the squared-mean bias. This estimator is non-negative and coincides with the empirical Fisher when $\mathbf{g}_\mathcal{R} = \mathbf{0}$.

Task-balanced estimation. Since the retain set $\mathcal{R}$ can be severely imbalanced across tasks, usually dominated by $\mathcal{D}_T$, a uniform empirical estimator over $\mathcal{R}$ can systematically misestimate curvature along directions important to historical tasks. Thus, we partition by task, $\mathcal{R}_t$ for $t = 1, \cdots, T$, compute a per-task variance Fisher $\hat{\mathbf{F}}_{\text{var}}^{(\mathcal{R}_t)}$, and average with equal weight:

$$[\hat{\mathbf{F}}_{\text{CFI}}]_j = \frac{1}{T}\sum_{t=1}^{T}[\hat{\mathbf{F}}_{\text{var}}^{(\mathcal{R}_t)}]_j = \frac{1}{T}\sum_{t=1}^{T}\left[\frac{1}{|\mathcal{R}_t|}\sum_{z\in\mathcal{R}_t} g_j(z)^2 - \left(\frac{1}{|\mathcal{R}_t|}\sum_{z\in\mathcal{R}_t} g_j(z)\right)^2\right]. \tag{6}$$

Equal task weighting alleviates the buffer imbalance issue in CL.

Preconditioned update. The Phase II update combines Phase I's target with $\hat{\mathbf{F}}_{\text{CFI}}$ and a damping term $\lambda > 0$ for numerical stability: $\boldsymbol{\delta}_I = \big(\hat{\mathbf{F}}_{\text{CFI}} + \lambda\mathbf{I}\big)^{-1}\mathbf{g}_{\text{target}}$. Since $\hat{\mathbf{F}}_{\text{CFI}}$ is diagonal, inversion is element-wise and the full phase costs $\mathcal{O}(|\mathcal{R}| \cdot p)$, the same order as one gradient pass.

4.3 Phase III: Forget-Subspace Projection

Phases I and II produce a well-scaled influence update $\boldsymbol{\delta}_I$, yet it can cause measurable degradation on historical tasks if $\boldsymbol{\delta}_I$ has a non-negligible component along directions relevant to the retain set. Thus, Phase III first selects forget-supported components and then removes their alignment with the retained-buffer subspace.

Forget and retain gradient subspaces. Let $\{\nabla\mathcal{L}(z, \boldsymbol{\theta}_T) : z \in \mathcal{S}\} \subset \mathbb{R}^p$ be the per-sample forget gradients, and similarly for $\mathcal{B}' = \mathcal{B} \setminus \mathcal{S}$. We compute orthonormal bases $\mathbf{Q}_\mathcal{S} \in \mathbb{R}^{p\times r_f}$ and $\mathbf{Q}_{\mathcal{B}'} \in \mathbb{R}^{p\times r_r}$ of their respective spans via modified Gram–Schmidt, with a fixed deterministic sample order and ranks $r_f \le \min(|\mathcal{S}|, r_{\max})$ and $r_r \le \min(|\mathcal{B}'|, r_{\max})$ truncated by $r_{\max}$ for efficiency.

Target projection and retain-overlap removal. The projected update is

$$\boldsymbol{\delta}_{\text{proj}} = \big(\mathbf{I} - \mathbf{Q}_{\mathcal{B}'}\mathbf{Q}_{\mathcal{B}'}^\top\big)\mathbf{Q}_\mathcal{S}\mathbf{Q}_\mathcal{S}^\top\boldsymbol{\delta}_I. \tag{7}$$

The projector $\mathbf{Q}_\mathcal{S}\mathbf{Q}_\mathcal{S}^\top$ restricts $\boldsymbol{\delta}_I$ to a low-rank approximation of the forget-gradient subspace $\mathcal{G}_\mathcal{S}$; the outer projector $\mathbf{I} - \mathbf{Q}_{\mathcal{B}'}\mathbf{Q}_{\mathcal{B}'}^\top$ then removes components that lie along the retain-buffer subspace $\mathcal{G}_{\mathcal{B}'}$, suppressing cross-task interference. We also provide Proposition C.1 to quantify the first-order deviation introduced by the projection.

4.4 Complete Algorithm

Algorithm 1 returns $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}_T - \boldsymbol{\delta}_{\text{proj}}$, with overall cost $O(|\mathcal{A}| \cdot p + (|\mathcal{S}| + |\mathcal{B}'|) \cdot r_{\max} \cdot p)$. See Section C.4 for the detailed complexity analysis. Let $\mathcal{E} = \mathcal{D} \setminus \mathcal{S}$ and let $\boldsymbol{\theta}^{\text{rt}}$ denote the output of sequential retraining on $\mathcal{E}$. Define $\Gamma = \|\mathbf{g}_\mathcal{R} - \mathbf{H}_\mathcal{R}\boldsymbol{\delta}_I\|_2 + \|\mathbf{H}_\mathcal{R}(\boldsymbol{\delta}_I - \boldsymbol{\delta}_{\text{proj}})\|_2 + \frac{L_H}{2}\|\boldsymbol{\delta}_{\text{proj}}\|_2^2$.

Theorem 4.1 (Conditional residual and loss bounds)**.** *Suppose $\mathcal{L}_\mathcal{R}$ has an L_H-Lipschitz Hessian on a convex neighborhood Ω containing $\boldsymbol{\theta}_T$, $\hat{\boldsymbol{\theta}}$, and $\boldsymbol{\theta}^{\text{rt}}$, and $\sup_{\theta\in\Omega}\|\nabla\mathcal{L}_\mathcal{R}(\boldsymbol{\theta}) - \nabla\mathcal{L}_\mathcal{E}(\boldsymbol{\theta})\|_2 \le \epsilon_B$. Then $\|\nabla\mathcal{L}_\mathcal{E}(\hat{\boldsymbol{\theta}})\|_2 \le \Gamma + \epsilon_B$. If $\mathcal{L}_\mathcal{E}$ additionally satisfies the PL inequality $\|\nabla\mathcal{L}_\mathcal{E}(\boldsymbol{\theta})\|_2^2 \ge 2\mu_{\text{PL}}(\mathcal{L}_\mathcal{E}(\boldsymbol{\theta}) - \inf_\Omega\mathcal{L}_\mathcal{E})$, then, writing $\rho_{\text{CL}} = \|\nabla\mathcal{L}_\mathcal{E}(\boldsymbol{\theta}^{\text{rt}})\|_2$,*

$$-\frac{\rho_{\text{CL}}^2}{2\mu_{\text{PL}}} \le \mathcal{L}_\mathcal{E}(\hat{\boldsymbol{\theta}}) - \mathcal{L}_\mathcal{E}(\boldsymbol{\theta}^{\text{rt}}) \le \frac{(\Gamma + \epsilon_B)^2}{2\mu_{\text{PL}}}. \tag{8}$$

Appendix C.2 gives the proof and a parameter-distance corollary under strong convexity of $\mathcal{L}_\mathcal{E}$.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Datasets and CL settings. Three widely used CL benchmarks are used: CIFAR-100 (Krizhevsky et al., 2009; Duan et al., 2025; Tong et al., 2025), CIFAR-10 (Krizhevsky et al., 2009; Duan et al., 2025; Tong et al., 2025), and ImageNet-100 (Deng et al., 2009; Duan et al., 2025; Cho et al., 2025). We follow (Duan et al., 2025) to partition CIFAR-100, CIFAR-10, and ImageNet-100 into 10, 5, and 10 disjoint tasks, where each task contains 10, 2, and 10 classes, respectively. We adopt CSReL-CL (Tong et al., 2025) as the backbone CL model, which is trained with a state-of-the-art Coreset replay strategy. We follow Tong et al. (2025) to adopt a ResNet-18 architecture and set the buffer size to 200. More experimental setup and implementation details are given in Appendix E.1.

Unlearning scenarios. Following Wu & Harandi (2025), we evaluate two complementary scenarios: (1) *Class-wise unlearning*: $\mathcal{S}$ consists of all training samples of a random target class, modelling a GDPR-style erasure of an entire data category. (2) *Random-sample unlearning*: $\mathcal{S}$ contains 1,000 samples drawn randomly across all classes, modelling fine-grained per-instance erasure requests.

Baselines. Three classes of baselines are used: (1) *Reference*: **Origin** method (Tong et al., 2025) without unlearning; **Retrain** the backbone on $\mathcal{D} \setminus \mathcal{S}$ as the gold-standard unlearning target. We then compare CLIF with following unlearning baselines using accessible data $\mathcal{R}$ and forget $\mathcal{S}$. (2) *Representative unlearning*: **GA** (Thudi et al., 2022), **FT** (Warnecke et al., 2023), **IF** (Koh & Liang, 2017; Warnecke et al., 2023), **SSD** (Foster et al., 2024), **FML** (Tarun et al., 2024), and state-of-the-art methods **DELETE** (Zhou et al., 2025), **MUNBa** (Wu & Harandi, 2025), **GFOES** (Song et al., 2026). (3) *Unlearning in CL*: **ErrorEraser** (Cao et al., 2025). Note that there is limited prior work on this class. One work (Liu et al., 2022) explores unlearning in CL, but it needs to know unlearning data at training time, which is not applicable and not fair for comparison. Several studies focus on sequential conventional unlearning requests (Thakral et al., 2025; Lee et al., 2025b; George et al., 2025; Xu et al., 2026), instead of the unlearning for CL models, are also not directly comparable. Detailed descriptions for baselines are provided in Appendix E.1.2.

Metrics. For class-wise unlearning, we follow Zhou et al. (2025) and report the accuracy $\mathrm{Acc_f}$ on forget set $\mathcal{S}$, $\mathrm{Acc_r}$ on retain set $\mathcal{R}$, $\mathrm{Acc_{ft}}$ on test data of the unlearned class, and $\mathrm{Acc_{rt}}$ on test data of retained classes. We also report two combined metrics to evaluate the forget-retain balance: $\mathrm{Acc_a} = \sqrt{(1-\mathrm{Acc_f}) \cdot \mathrm{Acc_r}}$ and $\mathrm{Acc_{at}} = \sqrt{(1-\mathrm{Acc_{ft}}) \cdot \mathrm{Acc_{rt}}}$. Following Zhou et al. (2025), we also adopt membership inference attack (MIA) success rate as a metric to evaluate the forgetting performance of the unlearn model. For random-sample unlearning, a lower $\mathrm{Acc_f}$ is not necessarily better; the closeness to Retrain $\Delta\mathrm{Acc_f}$ is the main forgetting metric. As the forget set is randomly sampled across all classes, we report the overall accuracy $\mathrm{Acc_t}$ on test data.

5.2 RESULTS ON CLASS-WISE UNLEARNING

Table 1 reports class-wise unlearning results across the three benchmarks. From these results, we make the following observations: **(1) CLIF most closely tracks the Retrain reference.** Across all three datasets, CLIF achieves effective unlearning with low forget accuracy while maintaining the highest retain accuracy. Overall, CLIF consistently achieves the best combined accuracy among all unlearning baselines and performs closest to the Retrain reference, demonstrating its superior balance between forgetting and retention. **(2) Standard influence functions collapse.** The standard IF degrades performance across all benchmarks: $\mathrm{Acc_r}$ collapses to single-digit values, while $\mathrm{Acc_f}$ exhibits both very high mean values and very high standard deviations, indicating unstable performance and the risk of catastrophic damage to the model. This aligns with the failure modes identified in Section 3.1. **(3) Other baselines either over-correct or under-preserve.** Most unlearning baselines, such as GA, FT, SSD, and FML, achieve low forget accuracy but at the cost of severe retention degradation (e.g., SSD and FML on CIFAR-100 and ImageNet-100). ErrorEraser, as a CL-aware unlearning method, shows competitive performance on test data, especially on CIFAR-100 and ImageNet-100. Overall, MUNBa, DELETE, and GFOES achieve a better unlearning balance and perform closer to CLIF. **(4) CLIF is efficient.** CLIF is significantly faster than the Retrain reference, comparable to efficient baselines like MUNBa, and faster than state-of-the-art baselines like GFOES and ErrorEraser, demonstrating its practical efficiency for unlearning in CL.

Table 1: Class-wise unlearning results (%) on three CL benchmarks. **Bold** marks the best result among unlearning baselines (Origin and Retrain excluded); underline marks the second best.

Datasets	Methods	$\mathrm{Acc_f}\downarrow$	$\mathrm{Acc_r}\uparrow$	$\mathrm{Acc_a}\uparrow$	$\mathrm{Acc_{ft}}\downarrow$	$\mathrm{Acc_{rt}}\uparrow$	$\mathrm{Acc_{at}}\uparrow$	MIA ↓	Time (s) ↓
CIFAR-100	Origin	9.33 ± 10.10	98.88 ± 1.89	94.61 ± 6.18	11.33 ± 10.60	17.27 ± 0.63	39.07 ± 2.12	5.27 ± 6.36	6071.88
	retain	0.00 ± 0.00	97.67 ± 0.85	98.83 ± 0.43	0.00 ± 0.00	16.82 ± 0.28	41.01 ± 0.34	0.00 ± 0.00	5570.99
	GA	**0.00 ± 0.00**	84.31 ± 7.28	91.77 ± 3.92	**0.00 ± 0.00**	9.50 ± 0.81	30.81 ± 1.30	**0.00 ± 0.00**	48.98
	FT	**0.00 ± 0.00**	89.79 ± 6.16	94.72 ± 3.24	**0.00 ± 0.00**	8.94 ± 2.68	29.69 ± 4.34	**0.00 ± 0.00**	56.82
	Influence	33.33 ± 57.74	3.23 ± 5.56	11.01 ± 17.40	33.33 ± 57.74	0.67 ± 0.58	6.70 ± 5.80	**0.00 ± 0.00**	977.42
	SSD	**0.00 ± 0.00**	1.07 ± 1.81	7.06 ± 9.28	**0.00 ± 0.00**	1.05 ± 0.07	10.24 ± 0.34	**0.00 ± 0.00**	**3.69**
	FML	**0.00 ± 0.00**	17.67 ± 7.14	41.44 ± 8.57	**0.00 ± 0.00**	1.85 ± 0.74	13.41 ± 2.69	**0.00 ± 0.00**	8.88
	DELETE	1.00 ± 0.80	88.08 ± 7.45	93.33 ± 3.93	**0.00 ± 0.00**	13.37 ± 2.20	36.48 ± 3.08	1.20 ± 1.11	4.08
	MUNBa	**0.00 ± 0.00**	93.33 ± 1.78	96.60 ± 0.92	0.33 ± 0.58	11.50 ± 1.10	33.83 ± 1.49	**0.00 ± 0.00**	45.17
	GFOES	**0.00 ± 0.00**	76.76 ± 1.23	87.61 ± 0.70	**0.00 ± 0.00**	7.62 ± 0.14	27.60 ± 0.26	**0.00 ± 0.00**	229.84
	ErrorEraser	1.13 ± 1.01	87.97 ± 7.59	93.20 ± 4.09	**0.00 ± 0.00**	13.85 ± 2.31	37.12 ± 3.19	1.47 ± 1.55	44.00
	Ours	0.13 ± 0.12	**97.89 ± 3.35**	**98.86 ± 1.73**	**0.00 ± 0.00**	**15.65 ± 0.52**	**39.55 ± 0.65**	0.13 ± 0.12	25.89
CIFAR-10	Origin	28.15 ± 14.00	83.63 ± 11.38	77.12 ± 10.18	28.26 ± 16.08	44.71 ± 7.42	55.75 ± 3.25	30.58 ± 15.23	2701.26
	retain	0.00 ± 0.00	86.35 ± 10.47	92.78 ± 5.78	0.00 ± 0.00	43.53 ± 4.77	65.90 ± 3.61	0.00 ± 0.00	2539.98
	GA	**0.00 ± 0.00**	61.06 ± 4.52	78.10 ± 2.90	**0.00 ± 0.00**	13.49 ± 0.88	36.71 ± 1.19	**0.00 ± 0.00**	23.28
	FT	**0.00 ± 0.00**	84.94 ± 4.77	92.13 ± 2.62	**0.00 ± 0.00**	21.49 ± 2.25	46.31 ± 2.43	**0.00 ± 0.00**	34.50
	Influence	20.00 ± 44.72	38.92 ± 21.77	55.80 ± 31.20	20.00 ± 44.72	8.89 ± 4.97	26.67 ± 14.91	**0.00 ± 0.00**	825.61
	SSD	1.95 ± 4.27	58.44 ± 22.79	74.49 ± 11.39	1.96 ± 4.27	15.60 ± 10.67	37.51 ± 10.40	0.08 ± 0.11	17.82
	FML	**0.00 ± 0.00**	50.04 ± 4.42	70.69 ± 3.08	**0.00 ± 0.00**	11.41 ± 0.71	33.77 ± 1.04	0.14 ± 0.28	**6.40**
	DELETE	**0.00 ± 0.00**	65.39 ± 8.66	80.71 ± 5.51	**0.00 ± 0.00**	24.27 ± 8.98	48.59 ± 9.07	**0.00 ± 0.00**	38.35
	MUNBa	**0.00 ± 0.00**	76.79 ± 5.41	87.59 ± 3.13	**0.00 ± 0.00**	17.55 ± 0.97	41.88 ± 1.17	**0.00 ± 0.00**	52.27
	GFOES	**0.00 ± 0.00**	89.37 ± 3.18	94.53 ± 1.69	**0.00 ± 0.00**	21.03 ± 1.73	45.83 ± 1.88	**0.00 ± 0.00**	164.72
	ErrorEraser	6.99 ± 1.24	75.26 ± 11.08	83.44 ± 6.07	6.66 ± 0.78	18.25 ± 6.11	40.87 ± 6.45	10.95 ± 1.56	109.40
	Ours	0.10 ± 0.13	**90.87 ± 8.32**	**95.20 ± 4.45**	**0.00 ± 0.00**	**25.27 ± 6.36**	**49.96 ± 6.23**	0.06 ± 0.07	25.87
ImageNet-100	Origin	9.09 ± 5.32	99.21 ± 0.80	94.95 ± 3.17	7.33 ± 3.06	10.57 ± 0.22	31.29 ± 0.44	7.09 ± 6.13	33536.30
	retain	0.00 ± 0.00	97.46 ± 0.42	98.72 ± 0.21	0.00 ± 0.00	10.05 ± 0.42	31.69 ± 0.67	0.00 ± 0.00	31607.38
	GA	**0.00 ± 0.00**	69.84 ± 6.10	83.52 ± 3.61	**0.00 ± 0.00**	5.20 ± 0.42	22.79 ± 0.90	**0.00 ± 0.00**	**1.55**
	FT	**0.00 ± 0.00**	51.74 ± 8.46	71.77 ± 5.89	**0.00 ± 0.00**	4.73 ± 0.52	21.72 ± 1.19	**0.00 ± 0.00**	6.61
	Influence	**0.00 ± 0.00**	9.82 ± 0.29	31.33 ± 0.46	**0.00 ± 0.00**	1.01 ± 0.00	10.05 ± 0.00	**0.00 ± 0.00**	1077.47
	SSD	**0.00 ± 0.00**	0.08 ± 0.04	2.76 ± 0.73	**0.00 ± 0.00**	1.02 ± 0.01	10.08 ± 0.06	**0.00 ± 0.00**	5.99
	FML	**0.00 ± 0.00**	16.43 ± 3.55	40.37 ± 4.50	**0.00 ± 0.00**	1.70 ± 0.45	12.94 ± 1.78	**0.00 ± 0.00**	7.16
	DELETE	**0.00 ± 0.00**	77.48 ± 2.78	88.01 ± 1.58	**0.00 ± 0.00**	7.79 ± 0.17	27.92 ± 0.31	**0.00 ± 0.00**	3.53
	MUNBa	**0.00 ± 0.00**	58.31 ± 1.23	76.36 ± 0.81	**0.00 ± 0.00**	5.61 ± 0.32	23.69 ± 0.68	**0.00 ± 0.00**	24.84
	GFOES	**0.00 ± 0.00**	52.71 ± 4.42	72.56 ± 3.01	**0.00 ± 0.00**	4.69 ± 0.37	21.64 ± 0.84	**0.00 ± 0.00**	148.21
	ErrorEraser	0.95 ± 0.27	90.98 ± 3.14	94.92 ± 1.77	4.00 ± 0.00	8.38 ± 0.42	28.35 ± 0.71	1.52 ± 0.00	124.33
	Ours	0.41 ± 0.70	**91.54 ± 9.01**	**95.41 ± 5.09**	**0.00 ± 0.00**	**8.78 ± 0.28**	**29.63 ± 0.48**	0.82 ± 1.41	35.55

Table 2: Average random-sample unlearning results (%). **Bold** marks the best result among unlearning baselines (Origin and Retrain excluded); underline marks the second best.

Methods	CIFAR-100					CIFAR-10					ImageNet-100				
	$\mathrm{Acc_f}$	$\Delta\mathrm{Acc_f}\downarrow$	$\mathrm{Acc_r}\uparrow$	$\mathrm{Acc_a}\uparrow$	$\mathrm{Acc_t}\uparrow$	$\mathrm{Acc_f}$	$\Delta\mathrm{Acc_f}\downarrow$	$\mathrm{Acc_r}\uparrow$	$\mathrm{Acc_a}\uparrow$	$\mathrm{Acc_t}\uparrow$	$\mathrm{Acc_f}$	$\Delta\mathrm{Acc_f}\downarrow$	$\mathrm{Acc_r}\uparrow$	$\mathrm{Acc_a}\uparrow$	$\mathrm{Acc_t}\uparrow$
Origin	19.20	3.20	98.92	89.40	17.54	50.50	8.40	82.57	63.77	47.99	15.80	4.67	99.17	91.37	10.67
Retrain	16.47	0.00	93.84	88.47	16.19	47.63	0.00	86.92	67.42	47.47	11.13	0.00	97.10	92.89	10.35
GA	6.63	9.83	69.22	80.35	6.65	12.43	35.20	60.14	72.42	12.20	3.73	7.40	35.31	58.03	2.88
FT	5.60	10.87	56.69	73.15	5.64	12.90	34.73	55.97	69.64	11.88	5.47	5.67	60.06	75.29	5.08
Influence	0.97	15.50	6.45	21.11	5.38	9.63	38.00	16.84	28.48	10.18	1.47	9.67	3.16	11.61	1.00
SSD	1.00	15.47	9.65	30.90	1.00	9.93	37.70	0.44	6.22	10.00	9.83	1.30	75.25	82.34	7.09
FML	1.37	15.10	14.11	37.01	1.46	11.93	35.70	50.97	66.85	11.57	1.50	9.63	13.48	36.37	1.37
DELETE	5.93	10.53	33.73	56.32	9.69	23.77	23.87	49.11	61.13	25.03	7.10	4.03	43.25	63.23	7.01
MUNBa	6.90	9.57	70.24	80.83	6.89	11.13	36.50	48.65	65.74	10.03	6.80	4.33	52.21	69.58	5.69
GFOES	5.23	11.23	55.48	72.51	5.48	15.70	31.93	71.21	**76.77**	14.97	5.90	5.23	56.73	73.03	5.21
ErrorEraser	12.26	4.21	15.73	36.76	10.49	20.40	27.23	0.82	7.98	19.88	11.33	0.87	27.06	48.97	**7.68**
CLIF	14.37	**3.17**	**89.94**	**87.47**	**13.06**	28.87	**18.77**	**79.59**	75.03	**28.40**	10.87	**0.27**	**82.55**	**85.77**	7.28

5.3 Results on Random-Sample Unlearning

In random-sample unlearning, the gold-standard Retrain model retains a non-zero accuracy on $\mathcal{S}$: because $\mathcal{S}$ is sampled across classes, the retrained classifier still generalises to the forget instances by virtue of training on *statistically similar* retain samples. Consequently, the target on $\mathrm{Acc_f}$ is no longer zero but *closeness* to the Retrain reference, i.e., $\Delta\mathrm{Acc_f}$. From Table 2, we find that: **(1) CLIF better preserves performance.** On train-side retain accuracy, CLIF outperforms every unlearning baseline by a wide margin, showing the strongest preservation of the original CL model's performance on the retain set. This is particularly challenging given that the forget set is randomly sampled across all classes, which means that the unlearning method must preserve performance on every task simultaneously. In contrast, most baselines struggle to preserve performance on the retain set under this scenario. **(2) CLIF is closer to Retrain on the forget side.** On $\mathrm{Acc_f}$, CLIF achieves the closest match to Retrain on CIFAR-100 and CIFAR-10. On ImageNet-100, although CLIF has the second-closest mean forget accuracy $\mathrm{Acc_f}$, narrowly behind ErrorEraser, it has a smaller average run-wise deviation from retraining, i.e., $\Delta\mathrm{Acc_f}$, showing a more consistent match with the retraining behaviour across runs. The combined metric $\mathrm{Acc_a}$ shows that CLIF achieves the best balance between forgetting and retention on CIFAR-100 and ImageNet-100, and the second-best balance on CIFAR-10.

Meanwhile, CLIF outperforms unlearning baselines on the overall test accuracy $\mathrm{Acc_t}$ on CIFAR-100 and CIFAR-10, and performs competitively on ImageNet-100, demonstrating its exceptional generalisation performance under the random-sample unlearning protocol. **(3) Standard influence and SSD remain unstable.** Consistent with the observation in class-wise unlearning, the standard Influence baseline corrupts the retention performance greatly on all three datasets. Similarly, the SSD method performs instably on both CIFAR-100 and CIFAR-10, with very low $\mathrm{Acc_r}$. Overall, DELETE and ErrorEraser achieve competitive performance. The full results are in Appendix E.2.

5.4 Ablation Study

We isolate the contribution of each phase of CLIF by ablating: (1) w/o BCT: removing bias-corrected target while using the classical IF target; (2) w/o CFI: removing Continual Fisher Information while using standard IF Hessian matrix; (3) w/o VFF: removing variance-form Fisher; (4) w/o TBE: removing task-balanced estimation; (5) w/o TP: removing target projection; (6) w/o ROR: removing retain-overlap removal.

Table 3: Ablation study of CLIF.

Variants	$\mathrm{Acc_f}\downarrow$	$\mathrm{Acc_r}\uparrow$	$\mathrm{Acc_a}\uparrow$	$\mathrm{Acc_{ft}}\downarrow$	$\mathrm{Acc_{rt}}\uparrow$	$\mathrm{Acc_{at}}\uparrow$
w/o BCT	33.33	0.03	1.48	33.33	0.67	6.70
w/o CFI	14.60	**96.20**	90.63	15.00	**16.78**	37.76
w/o VFF	1.80	88.35	92.64	3.33	14.91	37.58
w/o TBE	**0.07**	92.75	96.13	**0.00**	14.74	38.29
w/o TP	1.60	82.99	89.96	3.00	11.57	32.90
w/o ROR	**0.07**	94.55	97.18	**0.00**	14.02	37.31
CLIF	0.20	95.85	**97.80**	**0.00**	14.96	**38.67**

We report the results of variants of CLIF on CIFAR-100 in Table 3. We observe that, without a bias-corrected target, the performance collapses, with both failing to forget and failing to retain. Removing CFI leads to higher retention accuracy but at the cost of higher forget accuracy. Notably, removing target projection causes a significant drop in retention accuracy, demonstrating the necessity of confining the update to forget-relevant directions. Overall, the full CLIF achieves the best balance between forgetting and retention, confirming the complementary contributions of each phase. Full Ablation results are provided in Appendix E.3.

5.5 Parameter Analysis

We study CLIF's robustness to its two hyperparameters, the Fisher damping λ and the maximum subspace rank $r_{\max}$, and show the results on CIFAR-100 with class-wise unlearning in Figure 3. **(1) Impact of λ.** We vary λ within the range of $\{10^i\}_{i=-4}^{1}$ and observe a sharp performance transition from $\lambda = 10^{-2}$ to 10^{-1}, after which performance becomes stable. This indicates that insufficient damping can over-amplify poorly estimated curvature directions and produce an unstable influence update. **(2) Impact of $r_{\max}$.** We vary $r_{\max}$ from 1 to 50 with an interval of 1. We find that the optimal subspace rank is small, indicating that the dominant forget directions are highly low-dimensional. Increasing $r_{\max}$ beyond this point overall degrades performance, suggesting that additional directions primarily introduce noise. Additional results are provided in Appendix E.4.

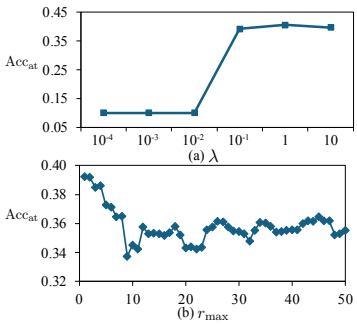


Figure 3: The performance of CLIF with different λ and $r_{\max}$.

6 Conclusion

We have presented CLIF, a novel framework for influence-based unlearning in replay-based continual learning. We identified two fundamental challenges when continual learning meets standard influence-based unlearning, i.e., non-stationarity and curvature unreliability. CLIF addresses both through three complementary phases: a bias-corrected influence target that re-derives the update without invoking stationarity (Phase I), a Continual Fisher Information that removes residual gradient inflation via a variance form and rebalances across tasks (Phase II), and a subspace projection that confines the update to forget-relevant directions (Phase III). Experiments on three benchmarks confirm that CLIF achieves effective unlearning while preserving performance on retained tasks, outperforming representative and state-of-the-art unlearning baselines.

AI USE STATEMENT

In this work, we used generative AI tools to assist with language polishing, grammar correction, and code debugging. We have not used generative AI tools for generating synthetic data sets, developing theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, supporting qualitative and thematic data analysis, assisting with translation, cleaning and reformatting datasets. All AI-assisted text and code suggestions were carefully reviewed and tested by the authors. We take responsibility for the final content of this work.

ETHICS STATEMENT

This paper proposes CLIF, a novel framework for influence-based unlearning in replay-based continual learning. The goal of our approach is to enable effective removal of the influence of designated data from continually trained models without full retraining, thereby supporting privacy, data correction, and responsible model maintenance. We do not anticipate any direct negative societal impacts from this work, and we believe that it complies with the ICLR Code of Ethics.

REPRODUCIBILITY STATEMENT

All results reported in this paper are fully reproducible. We provide the code and data for our method at `https://anonymous.4open.science/r/CLIF-DD56`. The repository includes implementation details, experiment configurations, hyperparameters, and scripts needed to reproduce the reported results.

REFERENCES

Sayanta Adhikari, Vishnuprasadh Kumaravelu, and P. K. Srijith. An unlearning framework for continual learning. *CoRR*, abs/2509.17530, 2025.

Cristian Daniel Alecsa and Szilárd Csaba László. Tikhonov regularization of a perturbed heavy ball system with vanishing damping. *SIAM Journal on Optimization*, 31(4):2921–2954, 2021.

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*, Lecture Notes in Computer Science, pp. 144–161. Springer, 2018.

Fatemeh Amrollahi, Supreeth P Shashikumar, Andre L Holder, and Shamim Nemati. Leveraging clinical data across healthcare institutions for continual learning of predictive risk models. *Scientific reports*, 12(1):8380, 2022.

Muhammad Awais, Muzammal Naseer, Salman Khan, Rao Muhammad Anwer, Hisham Cholakkal, Mubarak Shah, Ming-Hsuan Yang, and Fahad Shahbaz Khan. Foundation models defining a new era in vision: A survey and outlook. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(4):2245–2264, 2025.

Ali Ayub and Alan R. Wagner. EEC: learning to encode and regenerate images for continual learning. In *International Conference on Learning Representations (ICLR)*. OpenReview.net, 2021.

Juhan Bae, Nathan Ng, Alston Lo, Marzyeh Ghassemi, and Roger B. Grosse. If influence functions are the answer, then what is the question? In *Advances in Neural Information Processing Systems: Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

Samyadeep Basu, Phillip Pope, and Soheil Feizi. Influence functions in deep learning are fragile. In *International Conference on Learning Representations (ICLR)*. OpenReview.net, 2021.

Frederik Benzing. Unifying importance based regularisation methods for continual learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, Proceedings of Machine Learning Research, pp. 2372–2396. PMLR, 2022.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *IEEE Symposium on Security and Privacy (SP)*, pp. 141–159. IEEE, 2021.

Xuemei Cao, Hanlin Gu, Xin Yang, Bingjun Wei, Haoyang Liang, Xiangkun Wang, and Tianrui Li. Erroreraser: Unlearning data bias for improved continual learning. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 119–130. ACM, 2025.

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *IEEE Symposium on Security and Privacy (SP)*, pp. 463–480. IEEE Computer Society, 2015.

Romit Chatterjee, Vikram S. Chundawat, Ayush K. Tarun, Ankur Mali, and Murari Mandal. A unified framework for continual learning and machine unlearning. *CoRR*, abs/2408.11374, 2024.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. *CoRR*, abs/1902.10486, 2019.

Ruizhe Chen, Jianfei Yang, Huimin Xiong, Jianhong Bai, Tianxiang Hu, Jin Hao, Yang Feng, Joey Tianyi Zhou, Jian Wu, and Zuozhu Liu. Fast model debias with machine unlearning. In *Advances in Neural Information Processing Systems: Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

Dongkyu Cho, Taesup Moon, Rumi Chunara, Kyunghyun Cho, and Sungmin Cha. Forget forgetting: Continual learning in a world of abundant memory. *arXiv preprint arXiv:2502.07274*, 2025.

Yulai Cong, Miaoyun Zhao, Jianqiao Li, Sijia Wang, and Lawrence Carin. GAN memory with no forgetting. In *Advances in Neural Information Processing Systems: Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

R Dennis Cook and Sanford Weisberg. Residuals and influence in regression. 1982.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255. IEEE Computer Society, 2009.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations, ICLR*, 2021.

Yue Duan, Taicai Chen, Lei Qi, and Yinghuan Shi. Divide-and-conquer for enhancing unlabeled learning, stability, and plasticity in semi-supervised continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 583–593, October 2025.

Jack Foster, Stefan Schoepf, and Alexandra Brintrup. Fast machine unlearning without retraining through selective synaptic dampening. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pp. 12043–12051. AAAI Press, 2024.

Chongyang Gao, Lixu Wang, Kaize Ding, Chenkai Weng, Xiao Wang, and Qi Zhu. On large language model continual unlearning. In *International Conference on Learning Representations (ICLR)*. OpenReview.net, 2025.

Naveen George, Naoki Murata, Yuhta Takida, Konda Reddy Mopuri, and Yuki Mitsufuji. Distill, forget, repeat: A framework for continual unlearning in text-to-image diffusion models. *CoRR*, abs/2512.02657, 2025.

Aditya Golatkar, Alessandro Achille, Avinash Ravichandran, Marzia Polito, and Stefano Soatto. Mixed-privacy forgetting in deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 792–801. Computer Vision Foundation / IEEE, 2021.

Saisubramaniam Gopalakrishnan, Pranshu Ranjan Singh, Haytham M. Fayek, Savitha Ramasamy, and Arulmurugan Ambikapathi. Knowledge capture and replay for continual learning. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 337–345. IEEE, 2022.

Chuan Guo, Tom Goldstein, Awni Y. Hannun, and Laurens van der Maaten. Certified data removal from machine learning models. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, Proceedings of Machine Learning Research, pp. 3832–3842. PMLR, 2020.

Mustafa Burak Gurbuz and Constantine Dovrolis. NISPA: neuro-inspired stability-plasticity adaptation for continual learning in sparse networks. In *International Conference on Machine Learning (ICML)*, Proceedings of Machine Learning Research, pp. 8157–8174. PMLR, 2022.

Haochen Han, Alex Jinpeng Wang, Peijun Ye, and Fangming Liu. Unlearning the noisy correspondence makes clip more robust. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4518–4528, October 2025.

Jie Hao, Kaiyi Ji, and Mingrui Liu. Bilevel coreset selection in continual learning: A new formulation and algorithm. In *Advances in Neural Information Processing Systems: Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

Zhehao Huang, Xinwen Cheng, Jie Zhang, JingHao Zheng, Haoran Wang, Zhengbao He, Tao Li, and Xiaolin Huang. A unified gradient-based framework for task-agnostic continual learning-unlearning. *CoRR*, abs/2505.15178, 2025.

Zachary Izzo, Mary Anne Smart, Kamalika Chaudhuri, and James Zou. Approximate data deletion from machine learning models. In *The International Conference on Artificial Intelligence and Statistics, (AISTATS)*, Proceedings of Machine Learning Research, pp. 2008–2016. PMLR, 2021.

Hyundong Jin and Eunwoo Kim. Helpful or harmful: Inter-task association in continual learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, Lecture Notes in Computer Science, pp. 519–535. Springer, 2022.

Haeyong Kang, Rusty John Lloyd Mina, Sultan Rizky Hikmawan Madjid, Jaehong Yoon, Mark Hasegawa-Johnson, Sung Ju Hwang, and Chang D. Yoo. Forget-free continual learning with winning subnetworks. In *International Conference on Machine Learning (ICML)*, Proceedings of Machine Learning Research, pp. 10734–10750. PMLR, 2022.

Zixuan Ke, Bing Liu, Nianzu Ma, Hu Xu, and Lei Shu. Achieving forgetting prevention and knowledge transfer in continual learning. In *Advances in Neural Information Processing Systems: Annual Conference on Neural Information Processing Systems (NeurIPS)*, pp. 22443–22456, 2021.

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

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the International Conference on Machine Learning (ICML)*, Proceedings of Machine Learning Research, pp. 1885–1894. PMLR, 2017.

Pang Wei Koh, Kai-Siang Ang, Hubert H. K. Teo, and Percy Liang. On the accuracy of influence functions for measuring group effects. In *Advances in Neural Information Processing Systems: Annual Conference on Neural Information Processing Systems (NeurIPS)*, pp. 5255–5265, 2019.

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

N. Kishore Kumar and Jan Shneider. Literature survey on low rank approximation of matrices. *CoRR*, abs/1606.06511, 2016.

Frederik Kunstner, Philipp Hennig, and Lukas Balles. Limitations of the empirical fisher approximation for natural gradient descent. In *Advances in Neural Information Processing Systems: Annual Conference on Neural Information Processing Systems (NeurIPS)*, pp. 4158–4169, 2019.

Yongchan Kwon, Eric Wu, Kevin Wu, and James Zou. Datainf: Efficiently estimating data influence in lora-tuned llms and diffusion models. In *International Conference on Learning Representations (ICLR)*. OpenReview.net, 2024.

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Gregory G. Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3366–3385, 2022.

Olivier Ledoit and Michael Wolf. The power of (non-) linear shrinking: A review and guide to covariance matrix estimation. *Journal of Financial Econometrics*, 20(1):187–218, 2022.

Gyuseok Lee, SeongKu Kang, Wonbin Kweon, and Hwanjo Yu. Continual collaborative distillation for recommender system. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 1495–1505. ACM, 2024.

Justin Lee, Zheda Mai, Chongyu Fan, and Wei-Lun Chao. An empirical exploration of continual unlearning for image generation. In *The Impact of Memorization on Trustworthy Foundation Models: ICML 2025 Workshop*, 2025a.

Justin Lee, Zheda Mai, Jinsu Yoo, Chongyu Fan, Cheng Zhang, and Wei-Lun Chao. Continual unlearning for text-to-image diffusion models: A regularization perspective. *CoRR*, abs/2511.07970, 2025b.

Erich Leo Lehmann and George Casella. *Theory of point estimation*. Springer, 1998.

Timothée Lesort, Vincenzo Lomonaco, Andrei Stoian, Davide Maltoni, David Filliat, and Natalia Díaz Rodríguez. Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges. *Information fusion*, 58:52–68, 2020.

Jin Li, Huilin Gu, Shoujin Wang, Qi Zhang, Shui Yu, Chen Wang, Xiwei Xu, and Fang Chen. Towards fair large language model-based recommender systems without costly retraining. *CoRR*, abs/2601.17492, 2026a.

Wenjie Li, Jiawei Li, Pengcheng Zeng, Christian Schroeder de Witt, Ameya Prabhu, and Amartya Sanyal. Delta-influence: Identifying poisons via influence functions. *Transactions on Machine Learning Research*, 2026b.

Sen Lin, Peizhong Ju, Yingbin Liang, and Ness B. Shroff. Theory on forgetting and generalization of continual learning. In *International Conference on Machine Learning (ICML)*, Proceedings of Machine Learning Research, pp. 21078–21100. PMLR, 2023.

Bo Liu, Qiang Liu, and Peter Stone. Continual learning and private unlearning. In *Conference on Lifelong Learning Agents (CoLLAs)*, Proceedings of Machine Learning Research, pp. 243–254. PMLR, 2022.

Zhanwang Liu, Yuting Li, Haoyuan Gao, Yexin Li, Linghe Kong, Lichao Sun, and Weiran Huang. IDER: idempotent experience replay for reliable continual learning. *CoRR*, abs/2603.00624, 2026.

Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7765–7773. Computer Vision Foundation / IEEE Computer Society, 2018.

James Martens. Deep learning via hessian-free optimization. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 735–742. Omnipress, 2010.

Muhammad Anwar Ma'sum, Mahardhika Pratama, Savitha Ramasamy, Lin Liu, Habibullah, and Ryszard Kowalczyk. Vision and language synergy for rehearsal free continual learning. In *International Conference on Learning Representations (ICLR)*. OpenReview.net, 2025.

Seth Neel, Aaron Roth, and Saeed Sharifi-Malvajerdi. Descent-to-delete: Gradient-based methods for machine unlearning. In *Algorithmic Learning Theory (ALT)*, Proceedings of Machine Learning Research, pp. 931–962. PMLR, 2021.

Quoc Phong Nguyen, Bryan Kian Hsiang Low, and Patrick Jaillet. Variational bayesian unlearning. In *Advances in Neural Information Processing Systems: Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

Thanh Tam Nguyen, Thanh Trung Huynh, Zhao Ren, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. A survey of machine unlearning. *ACM Transactions on Intelligent Systems and Technology*, 16(5):108:1–108:46, 2025.

Data Protection. General data protection regulation. *Intersoft Consulting, Accessed in October*, 24 (1), 2018.

Xinbao Qiao, Meng Zhang, Ming Tang, and Ermin Wei. Hessian-free online certified unlearning. In *International Conference on Learning Representations (ICLR)*. OpenReview.net, 2025.

Penghao Rao, Runmin Jiang, and Min Xu. SG-OIF: A stability-guided online influence framework for reliable vision data. *CoRR*, abs/2511.19466, 2025.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5533–5542. IEEE Computer Society, 2017.

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 *International Conference on Learning Representations (ICLR)*. OpenReview.net, 2019a.

Matthew Riemer, Tim Klinger, Djallel Bouneffouf, and Michele Franceschini. Scalable recollections for continual lifelong learning. In *The AAAI Conference on Artificial Intelligence (AAAI)*, pp. 1352–1359. AAAI Press, 2019b.

Andrea Schioppa, Polina Zablotskaia, David Vilar, and Artem Sokolov. Scaling up influence functions. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pp. 8179–8186. AAAI Press, 2022.

Thanveer Shaik, Xiaohui Tao, Haoran Xie, Lin Li, Xiaofeng Zhu, and Qing Li. Exploring the landscape of machine unlearning: A comprehensive survey and taxonomy. *IEEE Transactions on Neural Networks and Learning Systems*, 36(7):11676–11696, 2025.

Shaofei Shen, Chenhao Zhang, Yawen Zhao, Alina Bialkowski, Weitong Chen, and Miao Xu. Machine unlearning for streaming forgetting. In *European Conference on Artificial Intelligence (ECAI)*, Frontiers in Artificial Intelligence and Applications, pp. 3266–3273. IOS Press, 2025.

Xinyi Sheng, Wei Bao, and Liming Ge. Robust federated unlearning. In *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM)*, pp. 2034–2044. ACM, 2024.

Haizhou Shi, Zihao Xu, Hengyi Wang, Weiyi Qin, Wenyuan Wang, Yibin Wang, Zifeng Wang, Sayna Ebrahimi, and Hao Wang. Continual learning of large language models: A comprehensive survey. *ACM Computing Surveys*, 58(5):120:1–120:42, 2026.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems: Annual Conference on Neural Information Processing Systems (NeurIPS)*, pp. 2990–2999, 2017.

Qipeng Song, Nan Yang, Ziqi Xu, Yue Li, Wei Shao, and Feng Xia. Synthetic forgetting without access: A few-shot zero-glance framework for machine unlearning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pp. 25545–25553. AAAI Press, 2026.

Zhicheng Sun, Yadong Mu, and Gang Hua. Regularizing second-order influences for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20166–20175. IEEE, 2023.

Jianheng Tang, Huiping Zhuang, Di Fang, Jiaxu Li, Feijiang Han, Yajiang Huang, Kejia Fan, Leye Wang, Zhanxing Zhu, Shanghang Zhang, Houbing Herbert Song, and Yunhuai Liu. ACU: analytic continual unlearning for efficient and exact forgetting with privacy preservation. *CoRR*, abs/2505.12239, 2025.

Ayush K. Tarun, Vikram S. Chundawat, Murari Mandal, and Mohan S. Kankanhalli. Fast yet effective machine unlearning. *IEEE Transactions on Neural Networks and Learning Systems*, 35 (9):13046–13055, 2024.

Stefano Teso, Andrea Bontempelli, Fausto Giunchiglia, and Andrea Passerini. Interactive label cleaning with example-based explanations. In *Advances in Neural Information Processing Systems: Annual Conference on Neural Information Processing Systems (NeurIPS)*, pp. 12966–12977, 2021.

Kartik Thakral, Tamar Glaser, Tal Hassner, Mayank Vatsa, and Richa Singh. Continual unlearning for foundational text-to-image models without generalization erosion. *CoRR*, abs/2503.13769, 2025.

Anvith Thudi, Gabriel Deza, Varun Chandrasekaran, and Nicolas Papernot. Unrolling SGD: understanding factors influencing machine unlearning. In *IEEE European Symposium on Security and Privacy (EuroSP)*, pp. 303–319. IEEE, 2022.

Ruilin Tong, Yuhang Liu, Javen Qinfeng Shi, and Dong Gong. Coreset selection via reducible loss in continual learning. In *International Conference on Learning Representations (ICLR)*. OpenReview.net, 2025.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8):5362–5383, 2024a.

Weiqi Wang, Zhiyi Tian, and Shui Yu. Machine unlearning: A comprehensive survey. *CoRR*, abs/2405.07406, 2024b.

Alexander Warnecke, Lukas Pirch, Christian Wressnegger, and Konrad Rieck. Machine unlearning of features and labels. In *Annual Network and Distributed System Security Symposium (NDSS)*. The Internet Society, 2023.

Mitchell Wortsman, Vivek Ramanujan, Rosanne Liu, Aniruddha Kembhavi, Mohammad Rastegari, Jason Yosinski, and Ali Farhadi. Supermasks in superposition. In *Advances in Neural Information Processing Systems: Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

Jiancan Wu, Yi Yang, Yuchun Qian, Yongduo Sui, Xiang Wang, and Xiangnan He. GIF: A general graph unlearning strategy via influence function. In *Proceedings of the ACM Web Conference (WWW)*, pp. 651–661. ACM, 2023.

Jing Wu and Mehrtash Harandi. Munba: Machine unlearning via nash bargaining. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4754–4765, October 2025.

Abudukelimu Wuerkaixi, Qizhou Wang, Sen Cui, Wutong Xu, Bo Han, Gang Niu, Masashi Sugiyama, and Changshui Zhang. Adaptive localization of knowledge negation for continual LLM unlearning. In *Proceedings of the International Conference on Machine Learning (ICML)*, Proceedings of Machine Learning Research. PMLR / OpenReview.net, 2025.

Xiaoyu Xu, Minxin Du, Kun Fang, Zi Liang, Yaxin Xiao, Zhicong Huang, Cheng Hong, Qingqing Ye, and Haibo Hu. FIT: defying catastrophic forgetting in continual LLM unlearning. *CoRR*, abs/2601.21682, 2026.

Mengqi Xue, Haofei Zhang, Jie Song, and Mingli Song. Meta-attention for vit-backed continual learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 150–159. IEEE, 2022.

Haohan Yang, Yanxin Zhou, Jingda Wu, Haochen Liu, Lie Yang, and Chen Lv. Human-guided continual learning for personalized decision-making of autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 26(4):5435–5447, 2025a.

Yutao Yang, Jie Zhou, Xuanwen Ding, Tianyu Huai, Shunyu Liu, Qin Chen, Yuan Xie, and Liang He. Recent advances of foundation language models-based continual learning: A survey. *ACM Computing Surveys*, 57(5):112:1–112:38, 2025b.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *Proceedings of the International Conference on Machine Learning (ICML)*, Proceedings of Machine Learning Research, pp. 3987–3995. PMLR, 2017.

Yang Zhang, Zhiyu Hu, Yimeng Bai, Jiancan Wu, Qifan Wang, and Fuli Feng. Recommendation unlearning via influence function. *ACM Transactions on Recommender Systems*, 3(2):22:1–22:23, 2025.

Yu Zhou, Dian Zheng, Qijie Mo, Renjie Lu, Kun-Yu Lin, and Wei-Shi Zheng. Decoupled distillation to erase: A general unlearning method for any class-centric tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20350–20359. Computer Vision Foundation / IEEE, 2025.

Appendix of CLIF

CONTENTS

A Table of Notations **18**

B Limitations and Future Opportunities **19**

C Theoretical Analysis **20**

C.1 Proof of Proposition 4.1 20

C.2 Conditional Residual and Loss Bounds 20

C.3 First-order preservation of unlearning effect 22

C.4 Complexity Analysis 23

D Algorithm **24**

E More Details about Experiments **24**

E.1 Experimental Setup 24

E.2 The Full Results on Random-Sample Unlearning 26

E.3 The Full Results on Ablation Study 26

E.4 The Full Results on Parameter Analysis 27

E.5 Results of Different Buffer Sizes 27

E.6 Results with Different Backbones and Architectures 28

E.7 Unlearning at Different Trained Stages 29

F Broader Impacts **29**

G Related Work **30**

A TABLE OF NOTATIONS

Notation	Description
$\mathcal{D}$	The full training data.
$\mathcal{D}_t$	The associated dataset at the t-th task.
$\mathcal{B}$	The full buffer data collected during CL training.
$\mathcal{B}_t$	The buffer data for the t-th task.
$\mathcal{B}'$	The retained buffer data after removing the forget set $\mathcal{B} \setminus \mathcal{S}$.
$\mathcal{S}$	The forget set with samples to be unlearned.
$\mathcal{A}$	The full accessible data, i.e., $\mathcal{D}_T \cup \mathcal{B} \cup \mathcal{S}$.
$\mathcal{R}$	The accessible retain set, i.e., $(\mathcal{D}_T \cup \mathcal{B}) \setminus \mathcal{S}$.
$\mathcal{E}$	The ideal retain set $\mathcal{D} \setminus \mathcal{S}$.
$\mathcal{L}_{\mathcal{A}}$	The empirical loss over a dataset $\mathcal{A}$.
$\mathcal{L}_{\mathcal{D}_{1:T}}$	The full-history loss over all data from task 1 to T.
$\boldsymbol{\delta}$	The unlearning update on model parameters.
$\boldsymbol{\delta}_I$	The unlearning update computed by Phase II.
$\boldsymbol{\delta}_{\text{proj}}$	The unlearning update computed by Phase III.
$\boldsymbol{\theta}$	The model parameters.
$\boldsymbol{\theta}_t$	The model parameters trained at the t-th task.
$\boldsymbol{\theta}_T^{-\mathcal{S}}$	The model parameters after unlearning the forget set $\mathcal{S}$.
$\boldsymbol{\theta}^{\text{rt}}$	The model parameters after retraining on the remaining data $\mathcal{D} \setminus \mathcal{S}$ at the end of T tasks.
$\boldsymbol{\theta}^*$	The optimal model parameters that sit at the stationary point.
$\mathbf{H}$	The Hessian matrix of the empirical loss.
$\hat{\mathbf{F}}_{\text{diag}}$	The empirical diagonal Fisher.
$\hat{\mathbf{F}}_{\text{var}}$	The variance-form Fisher.
$\hat{\mathbf{F}}_{\text{var}}^{(\mathcal{R}_t)}$	The variance-form Fisher for the t-th task.
$\hat{\mathbf{F}}_{\text{CFI}}$	The continual Fisher information.
$\hat{\mathbf{F}}_{\text{naive}}$	The naive empirical Fisher.
$\mathbf{g}_{\mathcal{S}}$	The gradient of the empirical loss over the forget set $\mathcal{S}$.
$\mathbf{g}_{\text{target}}$	The bias-corrected influence target gradient for unlearning.
$\mathbf{g}_{\mathcal{A}}$	The residual gradient of the loss over the accessible data $\mathcal{A}$.
$\mathbf{I}$	The identity matrix.
$\mathbf{Q}_{\mathcal{S}}$	The orthonormal basis for the gradient subspace of forget set.
$\mathbf{Q}_{\mathcal{B}'}$	The orthonormal basis for the gradient subspace of retained buffer data.
$\mathcal{G}_{\mathcal{S}}$	The gradient subspace spanned by $\mathbf{Q}_{\mathcal{S}}$.
$\mathcal{G}_{\mathcal{B}'}$	The gradient subspace spanned by $\mathbf{Q}_{\mathcal{B}'}$.
λ	The damping coefficient for the continual Fisher information.
r_f	The dimension of orthonormal basis $\mathbf{Q}_{\mathcal{S}}$.
r_r	The dimension of orthonormal basis $\mathbf{Q}_{\mathcal{B}'}$.
$r_{\max}$	The maximum dimension for orthonormal basis.
p	The dimension of model parameters $\boldsymbol{\theta}$.
T	The number of total tasks.

B Limitations and Future Opportunities

CLIF provides the first framework of influence-based unlearning for replay-based CL. In this section, we discuss several potential limitations of our work and future opportunities for improvement.

Scope to replay-based CL. CLIF is designed for replay-based CL methods because buffer data is important to serve as a proxy to represent historical training data under the constraints of data availability. Correspondingly, CLIF is devised to make full use of the limited available data in three phases to achieve effective unlearning in CL. Regularisation-based methods (Benzing, 2022; Aljundi et al., 2018) and architecture-based methods (Mallya & Lazebnik, 2018; Wortsman et al., 2020) maintain no exemplars and therefore fall outside the framework. Extending CLIF to these methods becomes a natural future direction.

Buffer representativeness. CLIF necessarily relies on the accessible retain set as a proxy for the unavailable ideal retain set. Theorem 4.1 captures this mismatch, which bounds the discrepancy between the gradients of $\mathcal{L}_{\mathcal{R}}$ and $\mathcal{L}_{\mathcal{E}}$. A small or task-skewed replay buffer can increase this discrepancy and consequently loosen the residual and loss-fidelity bounds. More representative replay strategies, e.g., CSReL-CL (Tong et al., 2025) and BCSR (Hao et al., 2023), may therefore improve the quality of the accessible-data surrogate. Nevertheless, our empirical analysis in Section E.5 shows that CLIF remains effective across different buffer sizes and can stay close to the retraining reference even with limited replay data.

Approximation errors. CLIF approximates the effect of retraining using only limited accessible data and therefore remains subject to several approximation errors. First, its local analysis relies on a quadratic approximation around $\boldsymbol{\theta}_T$. In Theorem 4.1, the corresponding Taylor remainder is controlled by $\frac{L_H}{2}\|\boldsymbol{\delta}_{\mathrm{proj}}\|_2^2$, so the approximation is tighter when the final update remains within a sufficiently regular local neighbourhood. Second, Phase II uses the task-balanced, centred diagonal CFI together with damping to construct an inexpensive and numerically stable preconditioner from limited and task-skewed accessible data. This preconditioner need not exactly reproduce the local Hessian action, and the resulting mismatch is captured by $r_F = \|\mathbf{g}_{\mathcal{R}} - \mathbf{H}_{\mathcal{R}}\boldsymbol{\delta}_I\|_2$. Damping can limit numerical amplification, but cannot recover curvature information that is missing or distorted in $\mathcal{R}$; developing more faithful curvature approximations while controlling estimation variance and computational cost is therefore an important direction for future work. Third, Phase III truncates the forget and retain gradient-subspace ranks at $r_{\max}$ for efficiency. Although a smaller $r_{\max}$ can reduce estimation noise empirically, rank truncation may discard informative directions and increase the projection residual $r_P = \|\mathbf{H}_{\mathcal{R}}(\boldsymbol{\delta}_I - \boldsymbol{\delta}_{\mathrm{proj}})\|_2$. Adaptive or task-dependent rank selection could potentially improve this fidelity–efficiency trade-off.

Single-shot unlearning. This work targets the canonical setting where one unlearning request is issued against a trained CL model, as a starting point to explore influence-based unlearning in CL. It is, however, not limited to this setting. CLIF can be applied to multiple unlearning requests sequentially. More generally, the unlearning requests can be issued at any time during the CL process, e.g., after training on each task, which has been discussed in Section E.7. This opens up interesting future directions to explore the influence-based unlearning in more dynamic and interactive CL scenarios.

Inherited assumptions. CLIF inherits the standard regularity conditions used to analyse IF-based unlearning, e.g., local strong convexity and Lipschitz Hessian. Deep networks can be non-convex globally, so the bound applies in a neighbourhood of $\boldsymbol{\theta}_T$, in line with prior IF analysis (Koh & Liang, 2017; Bae et al., 2022). CLIF also produces an approximation guarantee on parameter distance rather than an (ϵ, δ)-certified removal guarantee (Guo et al., 2020). Deriving certified guarantees that accommodate CL non-stationarity is a worthwhile target for follow-up theoretical work.

Future directions. Beyond the directions mentioned above, CLIF opens up several other interesting future directions. (1) Scaling CLIF to large pretrained backbones used in continual fine-tuning, e.g., vision and language foundation models (Awais et al., 2025), is an important future direction. (2) Evaluating CLIF under stronger membership-inference would analyse the gap between parameter-space approximation fidelity and empirical privacy. (3) The variance-form Fisher and task weighting

introduced in CLIF can be useful outside unlearning, e.g., in CL training to address the similar task imbalance issue.

C Theoretical Analysis

C.1 Proof of Proposition 4.1

Proposition 4.1 (Bias-corrected influence target). *Let $\boldsymbol{\theta}_T$ be the CL model, $\mathcal{S}$ the forget set, and $\mathcal{R}$ the accessible retain set. Define $\mathbf{g}_{\mathcal{A}} := \nabla_{\boldsymbol{\theta}}\mathcal{L}_{\mathcal{A}}(\boldsymbol{\theta}_T)$, $\mathbf{g}_{\mathcal{S}} := \nabla_{\boldsymbol{\theta}}\mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta}_T)$, and $\mathbf{H}_{\mathcal{R}} := \nabla^2_{\boldsymbol{\theta}}\mathcal{L}_{\mathcal{R}}(\boldsymbol{\theta}_T)$. The stationary condition of the second-order quadratic approximation of $\mathcal{L}_{\mathcal{R}}(\boldsymbol{\theta}_T - \boldsymbol{\delta})$ is*

$$\mathbf{H}_{\mathcal{R}}\boldsymbol{\delta} = \mathbf{g}_{\text{target}}, \qquad \mathbf{g}_{\text{target}} = \frac{|\mathcal{R}| + |\mathcal{S}|}{|\mathcal{R}|}\mathbf{g}_{\mathcal{A}} - \frac{|\mathcal{S}|}{|\mathcal{R}|}\mathbf{g}_{\mathcal{S}}. \tag{C.9}$$

We now provide the proof of Proposition 4.1 and establish a corollary showing that the classical IF target is recoverable as a special case.

Proof. Let

$$Q(\boldsymbol{\delta}) = \mathcal{L}_{\mathcal{R}}(\boldsymbol{\theta}_T) - \mathbf{g}_{\mathcal{R}}^{\top}\boldsymbol{\delta} + \frac{1}{2}\boldsymbol{\delta}^{\top}\mathbf{H}_{\mathcal{R}}\boldsymbol{\delta} \tag{C.10}$$

denote the second-order quadratic approximation of $\mathcal{L}_{\mathcal{R}}(\boldsymbol{\theta}_T - \boldsymbol{\delta})$ around $\boldsymbol{\theta}_T$, where $\mathbf{g}_{\mathcal{R}} = \nabla_{\boldsymbol{\theta}}\mathcal{L}_{\mathcal{R}}(\boldsymbol{\theta}_T)$ and $\mathbf{H}_{\mathcal{R}} = \nabla^2_{\boldsymbol{\theta}}\mathcal{L}_{\mathcal{R}}(\boldsymbol{\theta}_T)$. Then, $\nabla_{\boldsymbol{\delta}}Q(\boldsymbol{\delta}) = -\mathbf{g}_{\mathcal{R}} + \mathbf{H}_{\mathcal{R}}\boldsymbol{\delta}$. By setting this to zero, we have

$$\mathbf{H}_{\mathcal{R}}\boldsymbol{\delta} = \mathbf{g}_{\text{target}}, \tag{C.11}$$

where we define $\mathbf{g}_{\text{target}} := \mathbf{g}_{\mathcal{R}}$ to be the target gradient for bias correction.

Next, we decompose $\mathbf{g}_{\text{target}}$ via accessible gradients. Since $\mathcal{R}$ and $\mathcal{S}$ are disjoint, we have $|\mathcal{A}| = |\mathcal{R}| + |\mathcal{S}|$, and

$$\mathcal{L}_{\mathcal{A}}(\boldsymbol{\theta}) = \frac{|\mathcal{R}|}{|\mathcal{R}| + |\mathcal{S}|}\mathcal{L}_{\mathcal{R}}(\boldsymbol{\theta}) + \frac{|\mathcal{S}|}{|\mathcal{R}| + |\mathcal{S}|}\mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta}). \tag{C.12}$$

Taking gradients at $\boldsymbol{\theta}_T$ and rearranging, we have

$$\mathbf{g}_{\mathcal{A}} = \frac{|\mathcal{R}|}{|\mathcal{R}| + |\mathcal{S}|}\mathbf{g}_{\text{target}} + \frac{|\mathcal{S}|}{|\mathcal{R}| + |\mathcal{S}|}\mathbf{g}_{\mathcal{S}}, \tag{C.13}$$

and thus

$$\mathbf{g}_{\text{target}} = \frac{|\mathcal{R}| + |\mathcal{S}|}{|\mathcal{R}|}\mathbf{g}_{\mathcal{A}} - \frac{|\mathcal{S}|}{|\mathcal{R}|}\mathbf{g}_{\mathcal{S}}. \tag{C.14}$$

□

Corollary: Standard IF target recovery.

Corollary C.0.1 (Standard IF in the stationary ERM case)**.** In the case of ERM, where $\boldsymbol{\theta} = \boldsymbol{\theta}^*$ minimises $\mathcal{L}_{\mathcal{D}}$, so $\mathbf{g}_{\mathcal{A}} = \mathbf{0}$ and $\mathcal{R} = \mathcal{D} \setminus \mathcal{S}$ with full data accessibility. Then, the bias-corrected influence target Eq. equation C.11 reduces to:

$$\mathbf{g}_{\text{target}} = -\frac{|\mathcal{S}|}{|\mathcal{R}|}\mathbf{g}_{\mathcal{S}}, \tag{C.15}$$

and thus the unlearned model becomes $\boldsymbol{\theta}^{-\mathcal{S}} = \boldsymbol{\theta}^* + \frac{|\mathcal{S}|}{|\mathcal{R}|}\mathbf{H}_{\mathcal{R}}^{-1}\mathbf{g}_{\mathcal{S}}$, which matches the classical leave-$|\mathcal{S}|$-out IF formula (Koh & Liang, 2017; Bae et al., 2022).

C.2 Conditional Residual and Loss Bounds

Let $\mathcal{E} = \mathcal{D} \setminus \mathcal{S}$ denote the ideal retain set, and

$$\mathbf{M} = \hat{\mathbf{F}}_{\text{CFI}} + \lambda\mathbf{I}, \quad \boldsymbol{\delta}_I = \mathbf{M}^{-1}\mathbf{g}_{\mathcal{R}}, \quad \boldsymbol{\delta}_{\text{proj}} = (\mathbf{I} - \mathbf{P}_{\mathcal{B}})\mathbf{P}_{\mathcal{S}}\boldsymbol{\delta}_I, \quad \hat{\boldsymbol{\theta}} = \boldsymbol{\theta}_T - \boldsymbol{\delta}_{\text{proj}},$$

with $\mathbf{P}_{\mathcal{B}} = \mathbf{Q}_{\mathcal{B}'}\mathbf{Q}_{\mathcal{B}'}^{\top}$ and $\mathbf{P}_{\mathcal{S}} = \mathbf{Q}_{\mathcal{S}}\mathbf{Q}_{\mathcal{S}}^{\top}$. Define the directional residuals as approximation errors:

$$r_F = \|\mathbf{g}_{\mathcal{R}} - \mathbf{H}_{\mathcal{R}}\boldsymbol{\delta}_I\|_2, \qquad r_P = \|\mathbf{H}_{\mathcal{R}}(\boldsymbol{\delta}_I - \boldsymbol{\delta}_{\text{proj}})\|_2, \qquad \Gamma = r_F + r_P + \frac{L_H}{2}\|\boldsymbol{\delta}_{\text{proj}}\|_2^2. \tag{C.16}$$

Lemma C.1 (Residual transfer)**.** Suppose $\mathcal{L}_{\mathcal{R}}$ is twice continuously differentiable with Hessian L_H-Lipschitz on a convex set Ω containing $\boldsymbol{\theta}_T$ and $\hat{\boldsymbol{\theta}}$. Then,

$$\|\nabla\mathcal{L}_{\mathcal{R}}(\hat{\boldsymbol{\theta}})\|_2 \le \Gamma. \tag{C.17}$$

If $\|\nabla\mathcal{L}_{\mathcal{E}}(\hat{\boldsymbol{\theta}}) - \nabla\mathcal{L}_{\mathcal{R}}(\hat{\boldsymbol{\theta}})\|_2 \le \epsilon_B$, then,

$$\|\nabla\mathcal{L}_{\mathcal{E}}(\hat{\boldsymbol{\theta}})\|_2 \le \Gamma + \epsilon_B. \tag{C.18}$$

Proof. Apply Taylor expansion of $\nabla\mathcal{L}_{\mathcal{R}}$ around $\boldsymbol{\theta}_T$ gives

$$\nabla\mathcal{L}_{\mathcal{R}}(\hat{\boldsymbol{\theta}}) = \mathbf{g}_{\mathcal{R}} - \mathbf{H}_{\mathcal{R}}\boldsymbol{\delta}_{\mathrm{proj}} + \mathbf{R}, \tag{C.19}$$

where, by the Hessian-Lipschitz assumption, we have

$$\|\mathbf{R}\| \le \frac{L_H}{2}\|\boldsymbol{\delta}_{\mathrm{proj}}\|_2^2. \tag{C.20}$$

Therefore,

$$\|\nabla\mathcal{L}_{\mathcal{R}}(\hat{\boldsymbol{\theta}})\| \le \|\mathbf{g}_{\mathcal{R}} - \mathbf{H}_{\mathcal{R}}\boldsymbol{\delta}_I\|_2 + \|\mathbf{H}_{\mathcal{R}}(\boldsymbol{\delta}_I - \boldsymbol{\delta}_{\mathrm{proj}})\|_2 + \|\mathbf{R}\|_2 \le \Gamma. \tag{C.21}$$

Given $\|\nabla\mathcal{L}_{\mathcal{E}}(\hat{\boldsymbol{\theta}}) - \nabla\mathcal{L}_{\mathcal{R}}(\hat{\boldsymbol{\theta}})\|_2 \le \epsilon_B$ throughout the relevant neighbourhood, at $\hat{\boldsymbol{\theta}}$, we have

$$\begin{aligned}\|\nabla\mathcal{L}_{\mathcal{E}}(\hat{\boldsymbol{\theta}})\|_2 &\le \|\nabla\mathcal{L}_{\mathcal{E}}(\hat{\boldsymbol{\theta}}) - \nabla\mathcal{L}_{\mathcal{R}}(\hat{\boldsymbol{\theta}})\|_2 + \|\nabla\mathcal{L}_{\mathcal{R}}(\hat{\boldsymbol{\theta}})\|_2 \\ &\le \Gamma + \epsilon_B.\end{aligned} \tag{C.22}$$

□

Then, we provide the proof of Theorem 4.1.

Proof of Theorem 4.1. Under the assumptions of Theorem 4.1, Lemma C.1 gives

$$0 \le \mathcal{L}_{\mathcal{E}}(\hat{\boldsymbol{\theta}}) - \mathcal{L}_{\mathcal{E}}^{\inf} \le \frac{(\Gamma + \epsilon_B)^2}{2\mu_{\mathrm{PL}}}. \tag{C.23}$$

Similarly, at $\boldsymbol{\theta}^{\mathrm{rt}}$, we have

$$0 \le \mathcal{L}_{\mathcal{E}}(\boldsymbol{\theta}^{\mathrm{rt}}) - \mathcal{L}_{\mathcal{E}}^{\inf} \le \frac{\rho_{\mathrm{CL}}^2}{2\mu_{\mathrm{PL}}}. \tag{C.24}$$

Subtracting these two gaps to their common infimum gives

$$-\frac{\rho_{\mathrm{CL}}^2}{2\mu_{\mathrm{PL}}} \le \mathcal{L}_{\mathcal{E}}(\hat{\boldsymbol{\theta}}) - \mathcal{L}_{\mathcal{E}}(\boldsymbol{\theta}^{\mathrm{rt}}) \le \frac{(\Gamma + \epsilon_B)^2}{2\mu_{\mathrm{PL}}}. \tag{C.25}$$

In particular, the absolute loss difference is at most $\max\{(\Gamma + \epsilon_B)^2, \rho_{\mathrm{CL}}^2\}/(2\mu_{\mathrm{PL}})$. □

Corollary C.1.1 (Parameter distance under strong convexity)**.** Under Lemma C.1, suppose $\mathcal{L}_{\mathcal{E}}$ is μ-strongly convex on a convex set containing $\hat{\boldsymbol{\theta}}$ and $\boldsymbol{\theta}^{\mathrm{rt}}$. Then

$$\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^{\mathrm{rt}}\|_2 \le \frac{\Gamma + \epsilon_B + \rho_{\mathrm{CL}}}{\mu}. \tag{C.26}$$

Proof. By μ-strong convexity of $\mathcal{L}_{\mathcal{E}}$, we have

$$\begin{aligned}\mu\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^{\mathrm{rt}}\|_2 &\le \|\nabla\mathcal{L}_{\mathcal{E}}(\hat{\boldsymbol{\theta}}) - \nabla\mathcal{L}_{\mathcal{E}}(\boldsymbol{\theta}^{\mathrm{rt}})\|_2 \\ &\le \|\nabla\mathcal{L}_{\mathcal{E}}(\hat{\boldsymbol{\theta}})\|_2 + \|\nabla\mathcal{L}_{\mathcal{E}}(\boldsymbol{\theta}^{\mathrm{rt}})\|_2 \\ &\le \Gamma + \epsilon_B + \rho_{\mathrm{CL}}.\end{aligned} \tag{C.27}$$

Divide by μ, the result is immediate. □

Scope and interpretation. In this section, we discuss the theoretical bounds under different conditions, including under PL and convexity conditions. Without PL or another error-bound condition, Lemma C.1 supplies an approximate-stationarity bound. The terms r_F and r_P can in principle be evaluated using two Hessian–vector products on $\mathcal{R}$, and $\|\nabla\mathcal{L}_{\mathcal{R}}(\hat{\boldsymbol{\theta}})\|_2$ can be evaluated on $\mathcal{R}$. These optional diagnostics are not necessary steps of CLIF. In contrast, ϵ_B and ρ_{CL} refer to unavailable history or an oracle retraining run. The constants L_H, μ, and μ_{PL} also require independent justification. Thus, the bounds serve for interpreting approximation errors.

C.3 First-order preservation of unlearning effect

Proposition C.1 (First-order effect of truncated projection)**.** *Let $\mathbf{g}_{\mathcal{S}} = \nabla\mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta}_T)$ denote the mean forget gradient, and let*

$$\boldsymbol{\delta}_{\mathrm{proj}} = (\mathbf{I} - \mathbf{Q}_{\mathcal{B}'}\mathbf{Q}_{\mathcal{B}'}^{\top})\mathbf{Q}_{\mathcal{S}}\mathbf{Q}_{\mathcal{S}}^{\top}\boldsymbol{\delta}_I \tag{C.28}$$

be the Phase III projected update. Suppose $\mathcal{L}_{\mathcal{S}}$ has a $\beta_{\mathcal{S}}$-Lipschitz gradient along the line segment joining $\boldsymbol{\theta}_T$ and $\boldsymbol{\theta}_T - \boldsymbol{\delta}_{\mathrm{proj}}$. Then

$$\mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta}_T - \boldsymbol{\delta}_{\mathrm{proj}}) - \mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta}_T) = -\mathbf{g}_{\mathcal{S}}^{\top}\boldsymbol{\delta}_I + \varepsilon_{\mathrm{proj}} + \mathbf{R}_{\mathcal{S}}, \tag{C.29}$$

where

$$\varepsilon_{\mathrm{proj}} = \mathbf{g}_{\mathcal{S}}^{\top}(\boldsymbol{\delta}_I - \boldsymbol{\delta}_{\mathrm{proj}}) = \mathbf{g}_{\mathcal{S}}^{\top}\left[\mathbf{I} - (\mathbf{I} - \mathbf{Q}_{\mathcal{B}'}\mathbf{Q}_{\mathcal{B}'}^{\top})\mathbf{Q}_{\mathcal{S}}\mathbf{Q}_{\mathcal{S}}^{\top}\right]\boldsymbol{\delta}_I, \tag{C.30}$$

and

$$|\mathbf{R}_{\mathcal{S}}| \le \frac{\beta_{\mathcal{S}}}{2}\|\boldsymbol{\delta}_{\mathrm{proj}}\|_2^2. \tag{C.31}$$

Consequently,

$$|\varepsilon_{\mathrm{proj}}| \le \|\mathbf{g}_{\mathcal{S}}\|_2\|\boldsymbol{\delta}_I - \boldsymbol{\delta}_{\mathrm{proj}}\|_2. \tag{C.32}$$

If $\mathcal{L}_{\mathcal{R}}$ is additionally μ-strongly convex at the relevant region, then, with

$$r_P = \|\mathbf{H}_{\mathcal{R}}(\boldsymbol{\delta}_I - \boldsymbol{\delta}_{\mathrm{proj}})\|_2,$$

we have

$$|\varepsilon_{\mathrm{proj}}| \le \frac{r_P}{\mu}\|\mathbf{g}_{\mathcal{S}}\|_2. \tag{C.33}$$

Thus, r_P controls the deviation of the projected update from the unprojected first-order forget-loss effect.

Proof. By the $\beta_{\mathcal{S}}$-smoothness of $\mathcal{L}_{\mathcal{S}}$ along the segment from $\boldsymbol{\theta}_T$ to $\boldsymbol{\theta}_T - \boldsymbol{\delta}_{\mathrm{proj}}$, we have

$$\mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta}_T - \boldsymbol{\delta}_{\mathrm{proj}}) - \mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta}_T) = -\mathbf{g}_{\mathcal{S}}^{\top}\boldsymbol{\delta}_{\mathrm{proj}} + \mathbf{R}_{\mathcal{S}}, \tag{C.34}$$

where

$$|\mathbf{R}_{\mathcal{S}}| \le \frac{\beta_{\mathcal{S}}}{2}\|\boldsymbol{\delta}_{\mathrm{proj}}\|_2^2. \tag{C.35}$$

Adding and subtracting $-\mathbf{g}_{\mathcal{S}}^{\top}\boldsymbol{\delta}_I$ gives

$$-\mathbf{g}_{\mathcal{S}}^{\top}\boldsymbol{\delta}_{\mathrm{proj}} = -\mathbf{g}_{\mathcal{S}}^{\top}\boldsymbol{\delta}_I + \mathbf{g}_{\mathcal{S}}^{\top}(\boldsymbol{\delta}_I - \boldsymbol{\delta}_{\mathrm{proj}}). \tag{C.36}$$

Therefore,

$$\mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta}_T - \boldsymbol{\delta}_{\mathrm{proj}}) - \mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta}_T) = -\mathbf{g}_{\mathcal{S}}^{\top}\boldsymbol{\delta}_I + \varepsilon_{\mathrm{proj}} + \mathbf{R}_{\mathcal{S}}, \tag{C.37}$$

where

$$\varepsilon_{\mathrm{proj}} = \mathbf{g}_{\mathcal{S}}^{\top}(\boldsymbol{\delta}_I - \boldsymbol{\delta}_{\mathrm{proj}}) \tag{C.38}$$

$$= \mathbf{g}_{\mathcal{S}}^{\top}\left[\mathbf{I} - (\mathbf{I} - \mathbf{Q}_{\mathcal{B}'}\mathbf{Q}_{\mathcal{B}'}^{\top})\mathbf{Q}_{\mathcal{S}}\mathbf{Q}_{\mathcal{S}}^{\top}\right]\boldsymbol{\delta}_I. \tag{C.39}$$

By the Cauchy–Schwarz inequality,

$$|\varepsilon_{\mathrm{proj}}| \le \|\mathbf{g}_{\mathcal{S}}\|_2\|\boldsymbol{\delta}_I - \boldsymbol{\delta}_{\mathrm{proj}}\|_2. \tag{C.40}$$

Now suppose that $\mathcal{L}_{\mathcal{R}}$ is μ-strongly convex. Since

$$\mathbf{H}_{\mathcal{R}} = \nabla^2\mathcal{L}_{\mathcal{R}}(\boldsymbol{\theta}_T) \succeq \mu\mathbf{I},$$

for any vector $\mathbf{v}$,

$$\|\mathbf{H}_{\mathcal{R}}\mathbf{v}\|_2 \geq \mu\|\mathbf{v}\|_2. \tag{C.41}$$

Setting

$$\mathbf{v} = \boldsymbol{\delta}_I - \boldsymbol{\delta}_{\text{proj}},$$

we obtain

$$\|\boldsymbol{\delta}_I - \boldsymbol{\delta}_{\text{proj}}\|_2 \leq \frac{1}{\mu}\|\mathbf{H}_{\mathcal{R}}(\boldsymbol{\delta}_I - \boldsymbol{\delta}_{\text{proj}})\|_2 = \frac{r_P}{\mu}. \tag{C.42}$$

Substituting this into Eq. C.40 gives

$$|\varepsilon_{\text{proj}}| \leq \frac{r_P}{\mu}\|\mathbf{g}_{\mathcal{S}}\|_2, \tag{C.43}$$

which proves the result. □

C.4 Complexity Analysis

We analyse the computational cost of CLIF in terms of the number of model parameters p, the accessible retain set size $|\mathcal{R}|$, the forget set size $|\mathcal{S}|$, the retain buffer size $|\mathcal{B}'|$, and the subspace projection rank $r_{\max}$. Let $r_f \leq \min(|\mathcal{S}|, r_{\max})$ and $r_r \leq \min(|\mathcal{B}'|, r_{\max})$ denote the retained ranks of the forget and buffer gradient subspaces, respectively.

Phase I computes the mean gradients $\mathbf{g}_{\mathcal{A}}$ and $\mathbf{g}_{\mathcal{S}}$ at $\boldsymbol{\theta}_T$, and forms Eq. equation 4. These quantities can be computed by a single gradient pass over the accessible data $\mathcal{A}$. Thus, the time complexity is $\mathcal{O}((|\mathcal{R}| + |\mathcal{S}|)p)$, and the memory cost is $\mathcal{O}(p)$ for storing the gradients.

Phase II computes the diagonal CFI preconditioner $\hat{\mathbf{F}}_{\text{CFI}}$ by accumulating the first and second moments of the gradients over the retain set. The cost of computing the task-balanced variance-form Fisher is $\mathcal{O}(|\mathcal{R}|p)$. The damped inverse is applied element-wise, which costs only $\mathcal{O}(p)$ time and $\mathcal{O}(p)$ memory. Thus, the overall time complexity of Phase II is $\mathcal{O}(|\mathcal{R}|p)$ and the memory cost is $\mathcal{O}(p)$.

Phase III constructs orthonormal bases from gradients of the forget set and retain buffer. Computing these gradients costs $\mathcal{O}((|\mathcal{S}| + |\mathcal{B}'|)p)$. Using Gram-Schmidt with maximum rank $r_{\max}$, the construction costs $\mathcal{O}(p(|\mathcal{S}|r_f + |\mathcal{B}'|r_r))$. The projection step requires low-rank projections and costs $\mathcal{O}(p(r_f + r_r))$. The memory required to store the bases is $\mathcal{O}(p(r_f + r_r))$.

Overall, the time complexity of CLIF is

$$\mathcal{O}\left((|\mathcal{R}| + |\mathcal{S}|)p + |\mathcal{R}|p + (|\mathcal{S}| + |\mathcal{B}'|)p + p(|\mathcal{S}|r_f + |\mathcal{B}'|r_r) + p(r_f + r_r)\right). \tag{C.44}$$

Since $r_f, r_r \leq r_{\max}$ and usually $r_{\max} \ll p$, this simplifies to

$$\mathcal{O}\left((|\mathcal{A}|)p + p\, r_{\max}(|\mathcal{S}| + |\mathcal{B}'|)\right). \tag{C.45}$$

The dominant memory cost is

$$\mathcal{O}(p(r_f + r_r) + p). \tag{C.46}$$

Comparison with standard IFs. A direct Hessian-based IF method requires constructing or storing a dense Hessian $\mathbf{H}_{\mathcal{R}} \in \mathbb{R}^{p\times p}$, which costs $\mathcal{O}(p^2)$ memory. Exact inversion costs $\mathcal{O}(p^3)$ time. While Hessian-vector-product (HVP) solvers avoid the $\mathcal{O}(p^2)$ memory cost of explicit Hessian storage, they require a potentially large iterative cost $\mathcal{O}((|\mathcal{S}| + K|\mathcal{R}|)p)$, where K denote the number of iterations and is dependent on the condition number of $\mathbf{H}_{\mathcal{R}}$ and the desired precision. When the curvature is ill-conditioned, which is especially common in the CL setting, K can be large and lead to a high computational cost. In contrast, CLIF avoids explicit Hessian storage and costly iterative solvers by using a diagonal CFI preconditioner, which is efficient to compute and invert. The projection step further reduces the maximum rank of the update, leading to additional efficiency gains.

D ALGORITHM

Algorithm 1 CLIF: Continual-Learning Influence-based Forgetting

Input: CL model $\boldsymbol{\theta}_T$; forget set $\mathcal{S}$; buffer $\mathcal{B}$; current-task data $\mathcal{D}_T$; loss $\mathcal{L}$; hyperparameters $\lambda, r_{\max}$
Output: Unlearned model $\boldsymbol{\theta}_T^{-\mathcal{S}}$

$\mathcal{R} \leftarrow (\mathcal{D}_T \cup \mathcal{B}) \setminus \mathcal{S}$; $\quad \mathcal{B}' \leftarrow \mathcal{B} \setminus \mathcal{S}$
Compute per-sample gradients $\{\nabla \mathcal{L}(z, \boldsymbol{\theta}_T)\}_{z \in \mathcal{A}}$
// Phase I: Bias-corrected target
$\mathbf{g}_{\mathcal{R}} \leftarrow \frac{1}{|\mathcal{R}|} \sum_{z \in \mathcal{R}} \nabla \mathcal{L}(z, \boldsymbol{\theta}_T)$
$\mathbf{g}_{\text{target}} \leftarrow \mathbf{g}_{\mathcal{R}}$
// Phase II: Continual fisher preconditioner
$\hat{\mathbf{F}}_{\text{var}}^{(\mathcal{R}_t)} \leftarrow \text{diag}\big(\text{Var}_{z \in \mathcal{R}_t}[\nabla \mathcal{L}(z, \boldsymbol{\theta}_T)]\big)$
$\hat{\mathbf{F}}_{\text{CFI}} \leftarrow \frac{1}{T} \sum_{t=1}^{T} \hat{\mathbf{F}}_{\text{var}}^{(\mathcal{R}_t)}$
$\boldsymbol{\delta}_I \leftarrow (\hat{\mathbf{F}}_{\text{CFI}} + \lambda \mathbf{I})^{-1} \mathbf{g}_{\text{target}}$
// Phase III: Forget-subspace projection
$\mathbf{Q}_{\mathcal{S}} \leftarrow \text{GramSchmidt}(\{\nabla \mathcal{L}(z, \boldsymbol{\theta}_T)\}_{z \in \mathcal{S}}, r_{\max})$
$\mathbf{Q}_{\mathcal{B}'} \leftarrow \text{GramSchmidt}(\{\nabla \mathcal{L}(z, \boldsymbol{\theta}_T)\}_{z \in \mathcal{B}'}, r_{\max})$
$\boldsymbol{\delta}_{\text{proj}} \leftarrow (\mathbf{I} - \mathbf{Q}_{\mathcal{B}'} \mathbf{Q}_{\mathcal{B}'}^{\top})\, \mathbf{Q}_{\mathcal{S}} \mathbf{Q}_{\mathcal{S}}^{\top}\, \boldsymbol{\delta}_I$
return $\hat{\boldsymbol{\theta}} \leftarrow \boldsymbol{\theta}_T - \boldsymbol{\delta}_{\text{proj}}$

E MORE DETAILS ABOUT EXPERIMENTS

E.1 EXPERIMENTAL SETUP

E.1.1 DATASETS

We follow the common practice in CL to evaluate on three widely used benchmarks: CIFAR-100, CIFAR-10, and ImageNet-100 with the following settings.

- **CIFAR-100** dataset (Krizhevsky et al., 2009; Duan et al., 2025; Tong et al., 2025) contains 100 classes with 500 training images and 100 testing images per class. Each image is in size of 32×32. We follow Tong et al. (2025) to split this dataset into 10 disjoint tasks with 10 classes per task.
- **CIFAR-10** dataset (Krizhevsky et al., 2009; Duan et al., 2025; Tong et al., 2025) is similar to CIFAR-100 but consists of 10 classes with 50,000 training images and 10,000 test images in total. We follow Tong et al. (2025) to split this dataset into 5 disjoint tasks with 2 classes per task.
- **ImageNet-100** dataset (Deng et al., 2009; Duan et al., 2025; Cho et al., 2025) is a subset of the ImageNet-1K dataset, which contains 100 classes with 1,300 training images and 50 test images per class. We follow Duan et al. (2025) to split this dataset into 10 disjoint tasks with 10 classes per task.

E.1.2 BASELINES

We compare CLIF against three categories of baselines: (1) *Reference methods*: **Origin** method trains the backbone method (Tong et al., 2025) on the full dataset $\mathcal{D}$ without unlearning; **Retrain** method retrains the backbone method from scratch on the remaining data $\mathcal{D} \setminus \mathcal{S}$, serving as the gold-standard unlearning target. (2) *Representative unlearning methods*: **GA** (Thudi et al., 2022) performs gradient ascent on the forget set; **FT** (Warnecke et al., 2023) fine-tunes the model on the remaining data; **IF** (Koh & Liang, 2017; Warnecke et al., 2023) applies standard influence functions for machine unlearning; **SSD** (Foster et al., 2024) identifies key parameters and unlearns by dampening them according to their importance to the forget set; **FML** (Tarun et al., 2024) constructs the noise matrix for unlearning in the impair step and retains the knowledge in the repair step; **DELETE** (Zhou et al., 2025) applies a mask distillation approach to balance forgetting and retention; **MUNBa** (Wu & Harandi, 2025) frames unlearning as a two-player cooperative game and balances the contributions of

Table 5: Random-sample unlearning results (%) on three CL benchmarks. **Bold** marks the best result among unlearning baselines (Origin and Retrain excluded); underline marks the second best.

Datasets	Methods	$\mathrm{Acc_f}$	$\Delta\mathrm{Acc_f}\downarrow$	$\mathrm{Acc_r}\uparrow$	$\mathrm{Acc_a}\uparrow$	$\mathrm{Acc_t}\uparrow$
CIFAR-100	Origin	19.20 ± 0.53	3.20 ± 2.45	98.92 ± 1.86	89.40 ± 0.69	17.54 ± 0.38
	Retrain	16.47 ± 2.73	0.00 ± 0.00	93.84 ± 6.46	88.47 ± 2.24	16.19 ± 2.29
	GA	6.63 ± 0.35	9.83 ± 2.97	69.22 ± 5.38	80.35 ± 3.06	6.65 ± 0.36
	FT	5.60 ± 0.30	10.87 ± 2.53	56.69 ± 1.60	73.15 ± 0.95	5.64 ± 0.18
	Influence	0.97 ± 0.23	15.50 ± 2.57	6.45 ± 5.57	21.11 ± 17.06	5.38 ± 7.59
	SSD	1.00 ± 0.17	15.47 ± 2.86	9.65 ± 0.03	30.90 ± 0.08	1.00 ± 0.00
	FML	1.37 ± 0.31	15.10 ± 2.49	14.11 ± 4.24	37.01 ± 5.68	1.46 ± 0.44
	DELETE	5.93 ± 0.06	10.53 ± 2.77	33.73 ± 1.32	56.32 ± 1.12	9.69 ± 0.33
	MUNBa	6.90 ± 0.62	9.57 ± 2.46	70.24 ± 4.92	80.83 ± 2.71	6.89 ± 0.37
	GFOES	5.23 ± 0.59	11.23 ± 2.15	55.48 ± 1.46	72.51 ± 0.78	5.48 ± 0.13
	ErrorEraser	12.26 ± 1.52	4.21 ± 1.91	15.73 ± 5.33	36.76 ± 6.85	10.49 ± 1.45
	CLIF	14.37 ± 1.63	**3.17 ± 3.25**	**89.94 ± 15.72**	**87.47 ± 7.16**	**13.06 ± 1.57**
CIFAR-10	Origin	50.50 ± 4.35	8.40 ± 7.54	82.57 ± 8.36	63.77 ± 2.48	47.99 ± 4.33
	Retrain	47.63 ± 7.94	0.00 ± 0.00	86.92 ± 6.92	67.42 ± 7.48	47.47 ± 7.26
	GA	12.43 ± 1.54	35.20 ± 9.48	60.14 ± 8.52	72.42 ± 4.39	12.20 ± 1.49
	FT	12.90 ± 1.64	34.73 ± 9.30	55.97 ± 9.22	69.64 ± 5.14	11.88 ± 2.77
	Influence	9.63 ± 0.99	38.00 ± 8.32	16.84 ± 27.09	28.48 ± 33.06	10.18 ± 0.31
	SSD	9.93 ± 0.21	37.70 ± 8.10	0.44 ± 0.14	6.22 ± 0.98	10.00 ± 0.00
	FML	11.93 ± 3.10	35.70 ± 5.78	50.97 ± 8.72	66.85 ± 6.01	11.57 ± 1.37
	DELETE	23.77 ± 1.80	23.87 ± 9.73	49.11 ± 4.54	61.13 ± 2.71	25.03 ± 0.51
	MUNBa	11.13 ± 1.86	36.50 ± 7.86	48.65 ± 0.70	65.74 ± 0.46	10.03 ± 0.04
	GFOES	15.70 ± 4.17	31.93 ± 11.06	71.21 ± 19.14	**76.77 ± 9.20**	14.97 ± 4.37
	ErrorEraser	20.40 ± 6.50	27.23 ± 12.17	0.82 ± 0.25	7.98 ± 0.97	19.88 ± 5.05
	CLIF	28.87 ± 10.60	**18.77 ± 9.20**	**79.59 ± 25.10**	75.03 ± 17.64	**28.40 ± 10.42**
ImageNet-100	Origin	15.80 ± 1.11	4.67 ± 0.61	99.17 ± 0.76	91.37 ± 0.40	10.67 ± 0.46
	Retrain	11.13 ± 0.50	0.00 ± 0.00	97.10 ± 0.46	92.89 ± 0.33	10.35 ± 0.58
	GA	3.73 ± 0.85	7.40 ± 0.85	35.31 ± 7.96	58.03 ± 6.72	2.88 ± 0.82
	FT	5.47 ± 0.81	5.67 ± 0.57	60.06 ± 5.24	75.29 ± 2.98	5.08 ± 0.69
	Influence	1.47 ± 0.12	9.67 ± 0.50	3.16 ± 5.39	11.61 ± 16.29	1.00 ± 0.00
	SSD	9.83 ± 1.27	1.30 ± 0.90	75.25 ± 4.23	82.34 ± 1.88	7.09 ± 0.90
	FML	1.50 ± 0.46	9.63 ± 0.96	13.48 ± 2.07	36.37 ± 2.75	1.37 ± 0.19
	DELETE	7.10 ± 1.21	4.03 ± 0.71	43.25 ± 7.65	63.23 ± 6.11	7.01 ± 0.44
	MUNBa	6.80 ± 0.87	4.33 ± 0.61	52.21 ± 8.63	69.58 ± 5.78	5.69 ± 1.07
	GFOES	5.90 ± 0.72	5.23 ± 0.31	56.73 ± 3.51	73.03 ± 2.03	5.21 ± 0.08
	ErrorEraser	11.33 ± 0.85	0.87 ± 0.67	27.06 ± 1.68	48.97 ± 1.45	**7.68 ± 0.80**
	CLIF	10.87 ± 0.25	**0.27 ± 0.38**	**82.55 ± 1.89**	**85.77 ± 1.02**	7.28 ± 0.33

forgetting and preservation; **GFOES** (Song et al., 2026) incorporates a generative feedback network and a two-phase fine-tuning procedure to enable unlearning with limited forget data. (3) *Unlearning in CL*: **ErrorEraser** (Cao et al., 2025) shifts the decision space of unlearned data for effective unlearning. Note that there is limited prior work on unlearning in CL. Liu et al. (2022) proposes an initial framework for unlearning in CL, but it needs to know the data to be unlearned at training time, which is not applicable and fair for comparison. Several studies on continual unlearning (Thakral et al., 2025; Lee et al., 2025b; George et al., 2025; Xu et al., 2026), which focus on multiple conventional unlearning requests, instead of the unlearning for CL models, are also not directly comparable.

E.1.3 Implementation Details

We implement our CLIF with the state-of-the-art replay-based CL method, i.e., CSReL-CL (Tong et al., 2025), as the backbone. We follow the training settings in Tong et al. (2025) to train the CL model first and then apply CLIF for class-wise unlearning and random-sample unlearning. In specific,

we use the same ResNet-18 as in Tong et al. (2025) for the architecture of the CL model. The model is optimised by SGD with learning rate $2e^{-2}$ and a batch size of 32. The replay buffer size is set to 200 for all three datasets, and the loss factor for the replay term is set to 4. Following Tong et al. (2025), we use the same selection strategy to randomly select 5,000 samples from each task for training. During CLIF unlearning, we set the damping parameter λ to 0.1 for CIFAR-100 and CIFAR-10, and 1.5 for ImageNet-100. The maximum rank $r_{\max}$ is set to 2 for CIFAR-100, 24 for CIFAR-10, and 3 for ImageNet-100.

For the unlearning baselines, we follow the implementation details in their original papers. For the GA and FT baselines, we use the same SGD optimiser for a fair comparison. For influence function-based unlearning, we follow Koh & Liang (2017) to compute using the conjugate gradient method (Martens, 2010) with a maximum iteration of 100 and a damping factor of 0.01. For SSD, we follow Foster et al. (2024) to set hyperparameters α and λ to 10 and 1. For FML, we follow Tarun et al. (2024) to set the learning rate for the impair and repair steps to 0.02 and 0.01. For DELETE, we follow Zhou et al. (2025) to search the learning rate within the range of $[10^{-7}, 10^{-1}]$. Similarly, we set the learning rate of MUNBa (Wu & Harandi, 2025) within the range $[10^{-3}, 10^{-2}]$ and apply the L_1 regularisation with a factor of 0.02. For GFOES, we follow Song et al. (2026) to set the learning rate to $4e^{-3}$ for GFN and the erasure phase, and set it to $4e^{-4}$ for the recovery phase. For ErrorEraser, following Cho et al. (2025), we leverage the Adam optimiser, and set the learning rate to 0.0001 and the weight decay to $5e^{-4}$.

Our CLIF is implemented in PyTorch. All experiments are conducted on the cluster with two NVIDIA A40 GPUs, two Intel Xeon Gold 6346 CPUs, and 256GB RAM.

E.2 The Full Results on Random-Sample Unlearning

We provide the full results of random-sample unlearning on three datasets in Table 5, including standard deviations across different runs. Beyond the observations in the main paper, we can further find that: (1) CLIF achieves the smallest $\Delta\mathrm{Acc_f}$ on all three benchmarks. In particular, on ImageNet-100, ErrorEraser obtains a marginally closer mean $\mathrm{Acc_f}$, but its run-wise deviation $\Delta\mathrm{Acc_f}$ is more than three times larger than CLIF, showing that CLIF tracks Retrain consistently per run. (2) CLIF reports relatively higher run-wise variance than Retrain. We attribute this to the random-sample protocol itself. With forget set drawn across the entire space, it varies substantially across runs, and this variation further affects the influence target. Despite this, CLIF achieves the highest mean $\mathrm{Acc_r}$ among unlearning baselines on all three benchmarks, and the corresponding combined accuracy $\mathrm{Acc_a}$ remains best (CIFAR-100 and ImageNet-100) or second-best (CIFAR-10). (3) The standard IF collapses on the random-sample unlearning scenario in the same manner as in Table 1. These patterns are consistent across both unlearning scenarios, reinforcing that C1 and C2 are intrinsic to applying standard IFs in CL.

E.3 The Full Results on Ablation Study

We provide the ablation results of CLIF on three datasets in the class-wise unlearning scenario in Table 6 using the following variants. **(1) w/o BCT:** removing Phase I's bias-corrected target while using the classical IF target. **(2) w/o CFI:** removing Phase II's Continual Fisher Information while using the standard IF Hessian matrix. **(3) w/o VFF:** removing Phase II's variance-form Fisher. **(4) w/o TBE:** removing II's task-balanced estimation. **(5) w/o TP:** removing Phase III's target projection. **(6) ROR:** removing Phase III's retain-overlap removal.

From Table 6, we can observe that: (1) The bias-corrected target can significantly affect the unlearning performance and retain accuracy. Removing the BCT module leads to a similar behaviour as the standard IF, where the unlearning performance is significantly compromised, and the retention accuracy is close to zero. (2) The CFI preconditioner is crucial for the unlearning efficacy. Removing CFI while using the standard Hessian matrix leads to a higher forget set accuracy $\mathrm{Acc_f}$ and $\mathrm{Acc_{ft}}$. Although its combined accuracy remains high, e.g., $\mathrm{Acc_{at}}$ on ImageNet-100, the performance of the main task, unlearning, is severely compromised. (3) The variance-form Fisher and task-balanced estimation offer more accurate approximation for the designed CFI preconditioner in the CL setting. Removing either of them leads to a noticeable drop in combined accuracy, especially on CIFAR-10 and ImageNet-100. Among them, the variance-form Fisher has a relatively larger impact on the overall performance by correcting the residual gradient-induced inflation. (4) The target projection and retain-

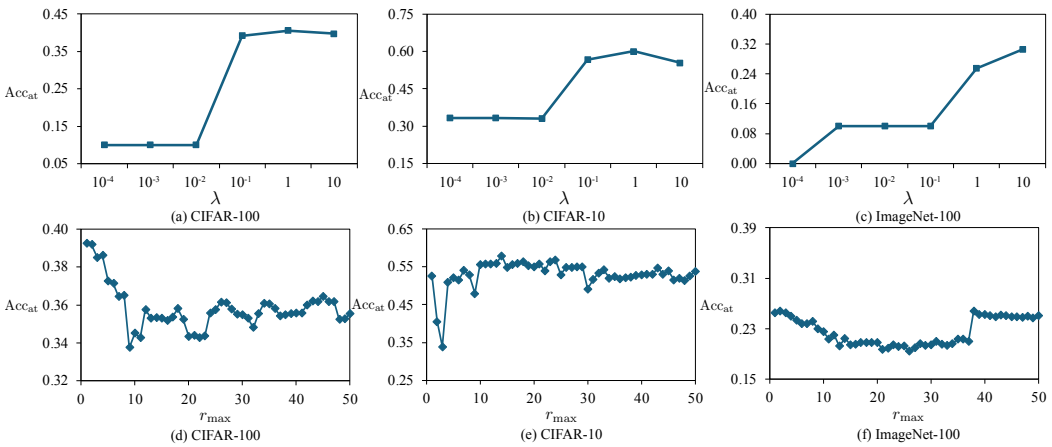


Figure 4: The performance of CLIF with different λ and $r_{\max}$ values on three datasets.

overlap removal are effective in improving the retain accuracy to prevent performance degradation. Removing either of them leads to a significant drop in $\mathrm{Acc_a}$ and $\mathrm{Acc_{at}}$. Notably, removing the target projection leads to a more severe decrease in retention accuracy, which indicates this projection can effectively constrain the unlearning update within the forget gradient subspace to prevent interference with the retain set. (5) Overall, the full CLIF method achieves the best performance in terms of combined accuracy $\mathrm{Acc_a}$ and $\mathrm{Acc_{at}}$ in most cases, except for $\mathrm{Acc_{at}}$ on ImageNet-100, where the w/o CFI variant achieves a slightly higher value but fails to unlearn thoroughly with a much higher $\mathrm{Acc_f}$ and $\mathrm{Acc_{ft}}$.

Table 6: Ablation study of CLIF on three datasets.

Variants	CIFAR-100				CIFAR-10				ImageNet-100			
	$\mathrm{Acc_f}\downarrow$	$\mathrm{Acc_a}\uparrow$	$\mathrm{Acc_{ft}}\downarrow$	$\mathrm{Acc_{at}}\uparrow$	$\mathrm{Acc_f}\downarrow$	$\mathrm{Acc_a}\uparrow$	$\mathrm{Acc_{ft}}\downarrow$	$\mathrm{Acc_{at}}\uparrow$	$\mathrm{Acc_f}\downarrow$	$\mathrm{Acc_a}\uparrow$	$\mathrm{Acc_{ft}}\downarrow$	$\mathrm{Acc_{at}}\uparrow$
w/o BCT	33.33	1.48	33.33	6.70	100.00	0.00	100.00	0.00	100.00	0.00	100.00	0.00
w/o CFI	14.60	90.63	15.00	37.76	15.64	58.53	14.47	49.27	15.71	91.03	14.00	**29.82**
w/o VFF	1.80	92.64	3.33	37.58	0.09	77.68	0.03	42.86	**0.00**	70.55	**0.00**	20.31
w/o TBE	**0.07**	96.13	**0.00**	38.29	0.09	85.06	0.03	46.87	**0.00**	71.50	**0.00**	20.62
w/o TP	1.60	89.96	3.00	32.90	**0.00**	78.08	**0.00**	38.24	**0.00**	56.45	**0.00**	17.48
w/o ROR	**0.07**	97.18	**0.00**	37.31	**0.00**	78.30	**0.00**	39.71	**0.00**	70.31	**0.00**	20.25
CLIF	0.20	**97.80**	**0.00**	**38.67**	0.10	**95.20**	**0.00**	**49.96**	0.41	**95.41**	**0.00**	29.63

E.4 The Full Results on Parameter Analysis

In this section, we provide the results of CLIF with different λ and $r_{\max}$ values on three datasets in the class-wise unlearning scenario. As shown in Figure 4, CLIF requires a relatively stronger Fisher damping parameter for the CFI preconditioner, as it achieves better performance when $\lambda \geq 10^{-1}$ on CIFAR-100 and CIFAR-10, and $\lambda \geq 1$ on ImageNet-100. The impact of $r_{\max}$ has different trends on different datasets to fit different data distributions. On CIFAR-100, CLIF achieves the best performance when $r_{\max}$ is set to 2. On CIFAR-10, the performance becomes stable when $r_{\max} \geq 10$. On ImageNet-100, the combined accuracy starts to drop when $r_{\max} > 3$ and increases when $r_{\max} \geq 38$.

E.5 Results of Different Buffer Sizes

The replay buffer plays an important role in CLIF by supplying historical-task samples to the accessible retain set. These samples contribute both to the CFI preconditioner and to how well $\mathcal{R}$ approximates the unavailable ideal retain set $\mathcal{E} = \mathcal{D} \setminus \mathcal{S}$. In Theorem 4.1, the latter discrepancy is captured by ϵ_B. To probe how CLIF responds to varying buffer sizes, we retrain the CSReL-CL backbone with different buffer sizes from 100 to 600 in steps of 100, and apply CLIF to unlearn class 0 on CIFAR-100. The results of our CLIF, Origin, and the gold-standard Retrain are reported in

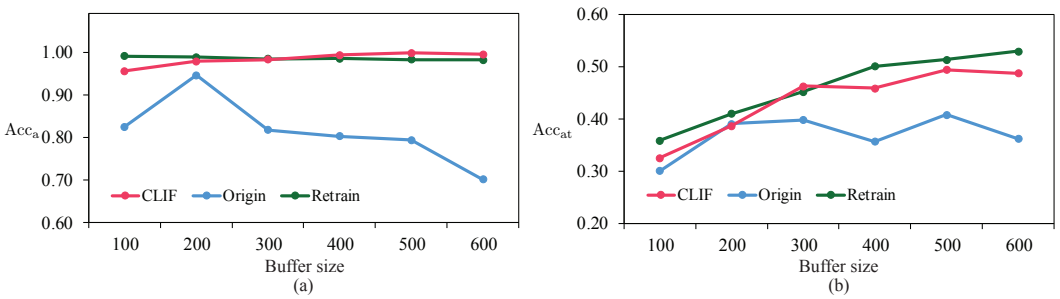


Figure 5: The performance of CLIF with different buffer sizes on CIFAR-100.

Figure 5. Two observations can be made. (1) CLIF closely tracks Retrain across all tested buffer sizes. Even with the smallest buffer size of 100, the combined accuracy of CLIF remains close to Retrain, and the gap to Retrain narrows as the buffer size grows. (2) CLIF achieves the closest performance to Retrain when the buffer size is moderate, i.e., around 300. This suggests that the buffer size ($|\mathcal{B}| = 200$) used in the main experiments is a reasonable choice that balances the trade-off between performance and efficiency. From a practical perspective, CLIF does not require unusually large buffer sizes to remain effective, which is desirable for memory-constrained CL scenarios. Overall, these results indicate that CLIF is robust to different buffer sizes and can effectively leverage the replay buffer to achieve unlearning performance close to Retrain, even with smaller buffers.

E.6 Results with Different Backbones and Architectures

To verify the generalisation of the proposed CLIF, we further implement CLIF with another representative replay-based CL method, i.e., BCSR (Hao et al., 2023). For BCSR implementation details, we follow the training settings in (Hao et al., 2023) to first train the CL model and then apply CLIF for class-wise unlearning. Specifically, BCSR uses two models: a training model M_{tr} for continual learning and a proxy model M_{cs} for coreset selection. At each stream mini-batch, the parameters of M_{tr} are copied to M_{cs}, and a bilevel optimisation procedure is applied to learn the sample-weight distribution for coreset selection. The selected coreset is then stored in the replay buffer and used together with the current stream data to update M_{tr}. For the model architecture, BCSR uses ResNet-18 for all three datasets. The proxy model M_{cs} adopts the same architecture as M_{tr}. The stream batch size is set to 50, the replay buffer size to 200, and the learning rate for M_{tr} to 0.15 on these datasets. For bilevel optimisation, the learning rates and the sample weights in M_{cs} are both set to 5, with regularisation coefficient $\lambda = 0.1$, Gaussian noise factor $\delta = 1e^{-3}$, outer loops $J \in \{5, 10\}$, inner loop number $N = 1$, and Hessian-inverse-vector estimation steps $Q = 3$. Based on the trained model, we perform each unlearning method and report the class-wise unlearning results in Table 7.

We make the following observations. (1) CLIF achieves the best $\mathrm{Acc}_{\mathrm{at}}$ on all three datasets. Combined with the results of CSReL-CL backbone in Table 1, this indicates that CLIF's design accommodates different CL backbones to achieve effective unlearning while maintaining high retention accuracy. (2) Standard IF remains ineffective with the BCSR backbone. The fact that the same failure mode occurs with different backbones further supports our analysis in Section 3.1: C1 (non-stationarity) and C2 (curvature unreliability) arise from the intrinsic mismatch between standard IFs and replay-based CL. (3) The simple GA baseline achieves a higher performance and shows better stability than several advanced unlearning methods, e.g., SSD and GFOES. This further indicates that the unlearning performance of these methods is not stable across different CL backbones, and their design may not be well-suited for the CL setting.

Beyond the ResNet-18 architecture, we further evaluate all methods using ViT-Tiny (Dosovitskiy et al., 2021) to examine whether their effectiveness generalises to Transformer-based architectures. The results are reported in Table 8. We make three observations. (1) CLIF remains effective under ViT-Tiny in most cases, achieving the highest $\mathrm{Acc}_{\mathrm{a}}$ and $\mathrm{Acc}_{\mathrm{at}}$ among the compared unlearning methods, while maintaining zero or near-zero $\mathrm{Acc}_{\mathrm{f}}$ and $\mathrm{Acc}_{\mathrm{ft}}$. (2) Standard influence function exhibits substantial instability under ViT-Tiny. Although it performs reasonably on CIFAR-10, its $\mathrm{Acc}_{\mathrm{ft}}$ increases to 40.67% and 60.00% on CIFAR-100 and ImageNet-100, respectively. Together

Table 7: The results (%) of CLIF with the BCSR backbone on three datasets.

Methods	CIFAR-100				CIFAR-10				ImageNet-100			
	$\mathrm{Acc_f}\downarrow$	$\mathrm{Acc_a}\uparrow$	$\mathrm{Acc_{ft}}\downarrow$	$\mathrm{Acc_{at}}\uparrow$	$\mathrm{Acc_f}\downarrow$	$\mathrm{Acc_a}\uparrow$	$\mathrm{Acc_{ft}}\downarrow$	$\mathrm{Acc_{at}}\uparrow$	$\mathrm{Acc_f}\downarrow$	$\mathrm{Acc_a}\uparrow$	$\mathrm{Acc_{ft}}\downarrow$	$\mathrm{Acc_{at}}\uparrow$
Origin	4.00	89.37	2.00	37.49	2.76	79.80	2.80	59.12	1.15	76.17	4.00	30.41
Retrain	0.00	92.83	0.00	57.58	0.00	81.38	0.00	70.34	0.00	76.85	0.00	45.76
GA	**0.00**	61.60	**0.00**	19.34	**0.00**	**80.09**	**0.00**	37.42	**0.00**	69.82	**0.00**	23.49
FT	1.00	56.54	**0.00**	32.33	**0.00**	47.43	**0.00**	37.58	0.54	14.62	2.00	15.81
Influence	94.00	0.00	0.01	9.85	9.72	63.64	10.20	30.31	**0.00**	1.41	**0.00**	11.79
SSD	1.40	86.33	1.00	35.40	**0.00**	4.12	**0.00**	31.78	1.38	76.21	4.00	30.42
FML	89.40	3.41	93.00	1.09	**0.00**	20.05	**0.00**	33.96	**0.00**	4.12	**0.00**	10.34
DELETE	**0.00**	41.99	**0.00**	31.96	**0.00**	64.40	**0.00**	50.00	0.77	70.05	**0.00**	30.02
MUNBa	4.20	89.31	2.00	34.92	1.38	72.06	1.60	43.74	1.38	76.22	4.00	30.45
GFOES	3.80	58.68	3.00	31.35	19.42	62.56	21.50	43.68	1.92	51.16	**0.00**	25.63
ErrorEraser	0.60	13.63	**0.00**	18.52	5.96	5.40	7.00	43.03	3.15	9.94	6.00	19.96
CLIF	0.80	**89.97**	**0.00**	**36.24**	**0.00**	79.06	**0.00**	**52.75**	0.38	**76.43**	2.00	**30.56**

Table 8: The results (%) of CLIF with the ViT-Tiny architecture on three datasets.

Methods	CIFAR-100				CIFAR-10				ImageNet-100			
	$\mathrm{Acc_f}\downarrow$	$\mathrm{Acc_a}\uparrow$	$\mathrm{Acc_{ft}}\downarrow$	$\mathrm{Acc_{at}}\uparrow$	$\mathrm{Acc_f}\downarrow$	$\mathrm{Acc_a}\uparrow$	$\mathrm{Acc_{ft}}\downarrow$	$\mathrm{Acc_{at}}\uparrow$	$\mathrm{Acc_f}\downarrow$	$\mathrm{Acc_a}\uparrow$	$\mathrm{Acc_{ft}}\downarrow$	$\mathrm{Acc_{at}}\uparrow$
Origin	36.70	37.22	26.33	31.09	66.55	35.68	62.12	35.15	40.23	33.28	21.20	28.04
Retrain	0.00	43.88	0.00	36.00	0.00	68.02	0.00	61.17	0.00	42.94	0.00	32.24
GA	**0.00**	25.41	**0.00**	21.38	**0.00**	30.81	**0.00**	33.33	**0.00**	17.22	**0.00**	17.60
FT	**0.00**	26.13	**0.00**	17.24	**0.00**	25.33	**0.00**	34.43	**0.00**	19.27	**0.00**	13.60
Influence	38.38	14.98	40.67	9.65	0.15	61.99	0.20	53.21	60.00	12.95	60.00	10.05
SSD	**0.00**	38.44	**0.00**	29.51	66.25	35.74	62.28	35.05	**0.00**	22.88	**0.00**	20.01
FML	**0.00**	27.41	**0.00**	11.24	**0.00**	55.40	**0.00**	33.95	**0.00**	27.87	**0.00**	10.68
DELETE	**0.00**	35.68	**0.00**	29.90	1.84	36.67	2.06	30.05	**0.00**	17.70	**0.00**	17.39
MUNBa	**0.00**	48.66	**0.00**	21.98	19.50	67.05	17.76	55.14	37.55	37.77	22.00	28.17
GFOES	**0.00**	42.45	**0.00**	17.32	0.77	53.50	0.84	33.82	**0.00**	17.93	**0.00**	10.27
ErrorEraser	**0.00**	32.24	**0.00**	26.34	81.99	10.61	78.66	15.77	5.75	26.85	4.80	24.36
Ours	**0.00**	**48.98**	1.33	**33.29**	0.38	**70.62**	0.13	**60.05**	2.30	**41.39**	2.00	**28.73**

with its unstable behaviour under ResNet-18, these results suggest that the limitations of standard IF in replay-based CL are not specific to CNN architectures, further supporting the motivation behind C1 and C2 in Section 1. (3) Several existing unlearning methods are also sensitive to the architecture and dataset. For example, SSD largely fails to forget on CIFAR-10, while MUNBa retains relatively high accuracy but leaves substantial residual forget accuracy on ImageNet-100. In contrast, CLIF consistently maintains low forget accuracy without severely compromising retention, supporting its applicability across both CNN- and Transformer-based architectures.

E.7 Unlearning at Different Trained Stages

To further verify the generalisation of CLIF, we apply CLIF to unlearn class 0 at different trained stages. We train the CSReL-CL backbone on CIFAR-100 up to $t \in \{2, 4, 6, 8, 10\}$ tasks, respectively, and at each intermediate stage perform class-wise unlearning of class 0 (which is contained in the first task and is therefore present in every stage) using CLIF. From Figure 6, we observe that CLIF aligns closely with the Retrain reference throughout the entire range of t. This indicates that the bias-corrected target effectively reduces the accumulating residual gradient, while Phase II and Phase III maintain stable scaling and direction control across stages. The result also suggests that, for long-running CL systems, the behaviour of CLIF remains stable and consistent with Retrain.

F Broader Impacts

CLIF addresses a problem with direct societal stakes: the removal of specific training data from continually learned models. As machine learning systems are increasingly deployed in dynamic

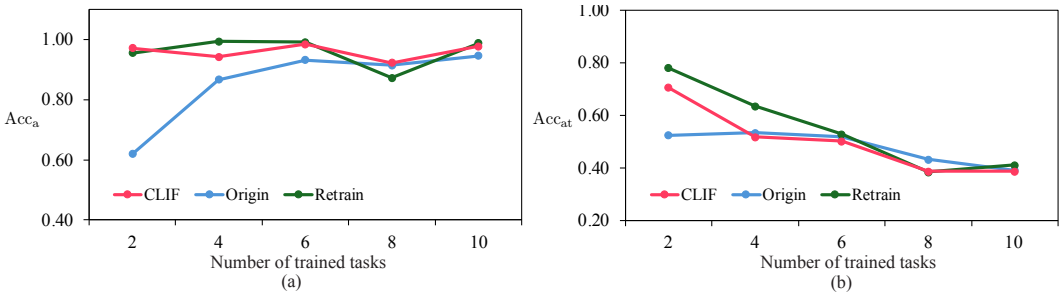


Figure 6: The performance of CLIF when unlearning at different trained stages of the CL model on CIFAR-100. The trained stage is represented by the number of tasks that the CL model has been trained on before unlearning.

environments, models are often updated sequentially from new data while retaining only limited historical records. In such settings, full retraining after every erasure request can be computationally prohibitive, making practical unlearning mechanisms important for responsible deployment.

CLIF can help make data-erasure compliance more tractable for replay-based continual learning systems. By producing a parameter update that approximates the gold-standard retrained model without re-running the full CL pipeline, CLIF lowers the operational cost of responding to deletion requests under privacy regulations and institutional data-governance policies (Protection, 2018). This is particularly relevant in domains such as healthcare (Amrollahi et al., 2022), autonomous driving (Yang et al., 2025a), robotics (Lesort et al., 2020), and recommender systems (Lee et al., 2024), where data arrives continuously, storage of full training histories may be restricted, and individual records may later need to be removed.

Beyond privacy compliance, CLIF may also support safer and more reliable model maintenance. In healthcare and clinical decision-support systems, it could help remove patient records that are later found to be erroneous, outdated, or no longer authorised for use. In autonomous driving and robotics, it may assist in excising corrupted sensor logs, unsafe demonstrations, or biased interaction data discovered after deployment. In recommender systems and personalised services, it can support the removal of user-specific behavioural traces while preserving the utility of the remaining model. More broadly, the same mechanism can be used to reduce the influence of adversarial, poisoned, biased, or unrepresentative exemplars that are identified post-deployment.

G Related Work

Continual learning aims to update models incrementally as new tasks arrive without catastrophically forgetting previously learned knowledge (Ke et al., 2021; Lange et al., 2022; Lin et al., 2023). Existing approaches broadly fall into three families. **(1) Regularisation-based methods** add penalty terms that discourage changes to parameters deemed important for previous tasks using different measures, e.g., Kirkpatrick et al. (2017); Benzing (2022); Zenke et al. (2017); Aljundi et al. (2018). They avoid storing raw data but need to store a frozen copy of the previous model (Wang et al., 2024a). **(2) Replay-based methods** maintain a small buffer of exemplars from previous tasks and interleave them with current-task data during training. Typical strategies to construct the buffer include random sampling (Riemer et al., 2019a), Mean-of-Exemplars (Rebuffi et al., 2017), feature clustering (Chaudhry et al., 2019), and influence selection (Tong et al., 2025; Sun et al., 2023). Another line of work generates pseudo-samples (Shin et al., 2017) for replay using generative models (Cong et al., 2020; Riemer et al., 2019b; Ayub & Wagner, 2021; Gopalakrishnan et al., 2022). **(3) Architecture-based methods** (Xue et al., 2022; Kang et al., 2022; Gurbuz & Dovrolis, 2022; Jin & Kim, 2022) allocate separate parameters or sub-networks per task (Mallya & Lazebnik, 2018; Wortsman et al., 2020). While these offer stronger isolation guarantees, they scale less gracefully with task count. In this work, we focus on replay-based methods, as they are widely adopted in the literature, and their finite and constrained data availability for historical tasks provides a practical yet challenging testbed for continual learning and unlearning.

Machine unlearning tackles the problem of efficiently removing the influence of specific training data from a trained model (Wang et al., 2024b; Shaik et al., 2025; Nguyen et al., 2025). We organise existing work by the unlearning guarantee. **(1) Exact unlearning** ensures that the unlearned model is statistically indistinguishable from the retrained one. A common approach is to design partitioned training pipelines for efficient retraining on affected partitions (Cao & Yang, 2015; Bourtoule et al., 2021; Golatkar et al., 2021). While principled, they need a splittable training pipeline that is incompatible with the sequential, non-partitioned nature of CL. **(2) Approximate unlearning** seeks efficient parameter updates that approximate the retrained model within controlled error bounds. Guo et al. (2020) propose certified data removal via a one-step Newton update, establishing (ϵ, δ)-certified removal. Within this framework (Guo et al., 2020; Neel et al., 2021), many works leverage IFs (Koh & Liang, 2017; Koh et al., 2019; Li et al., 2026b) to estimate unlearning influence and derive closed-form parameter updates to approximate retraining (Izzo et al., 2021; Warnecke et al., 2023; Chen et al., 2023; Li et al., 2026a). While elegant, influence functions have been shown to be fragile in deep learning (Basu et al., 2021), and several variants (Bae et al., 2022; Schioppa et al., 2022; Kwon et al., 2024) have been proposed to improve their robustness and scalability. Yet, these methods rely on access to the original training data for gradient and Hessian computation, an assumption that fundamentally breaks down in the CL setting and leads to the failure modes we identify in this work.

Unlearning in CL hasn't yet received enough attention, with only limited related works (Cao et al., 2025; Liu et al., 2022). Liu et al. (2022) propose an initial framework for exact unlearning in CL by deleting the model branches trained on the unlearned data. However, it has limited applicability for its large storage requirements of multiple model branches, and more importantly, the need to know data to unlearn during training, which is often unrealistic in practice. Cao et al. (2025) perform unlearning by shifting the decision space of unlearned data away from retained data, but due to the high entanglement of representations, it is challenging to accurately and thoroughly unlearn data. While there are some works on continual unlearning (Wuerkaixi et al., 2025; Gao et al., 2025; Lee et al., 2025a; Shen et al., 2025; Huang et al., 2025; Chatterjee et al., 2024; Tang et al., 2025; Adhikari et al., 2025), they aim to address the problem of sequential unlearning (Qiao et al., 2025; Thakral et al., 2025; Lee et al., 2025b; George et al., 2025; Xu et al., 2026), i.e., multiple conventional unlearning requests, instead of the unlearning for CL models. Thus, they are out of the scope of this work. To our knowledge, there is a lack of work studying the unique challenges of influence-based unlearning in the CL scenarios and providing an effective and practical solution.