Fed-f-SCRUB: Federated Unlearning via SCRUB Based on f-divergence

Anonymous Author(s)
Affiliation
Address
email

Abstract

The growing adoption of Federated Learning (FL) has brought distributed machine learning (ML) to the forefront, enabling collaborative model training while preserving data locality. However, emerging legal and ethical requirements, such as the "right to be forgotten," and the need to counter data poisoning attacks, highlight the critical necessity of efficient data unlearning in FL systems. Unlike traditional ML, where centralized data access facilitates unlearning, FLs decentralized nature makes the removal of specific data significantly more challenging. In this paper, we propose a novel federated unlearning framework that adapts the strengths of SCRUB based on f-divergences to address this gap to federated setup. Extensive experiments validate the effectiveness of our method, showing that it achieves substantial unlearning speed-ups while preserving model performance and offering strong formal guarantees. This work takes an important step toward building federated learning systems that are not only scalable and efficient but also legally compliant, fair, and transparent.

1 Introduction

The rapid progress of modern machine learning (ML) systems is largely driven by the abundance of data available in todays digital landscape. However, despite the advantages this vast data offers, several critical concerns arise. First, there is the question of whether the individuals or entities contributing data have consented to its use in developing ML models. Second, the integrity of these models can be compromised by the presence of poisoned data and mislabeled examples. Moreover, legal frameworks such as the European Unions "Right to Be Forgotten", (**?**), emphasize the increasing importance of prioritizing safety and privacy in the evolving landscape of ML and AI systems.

In response to these concerns, the concept of machine unlearning has emerged (**?**). Its primary goal is to remove the influence of specific data points from a trained model. A straightforward method to achieve this, known as exact unlearning, involves retraining the model from scratch without the targeted data. However, this approach is often impractical due to its significant computational costespecially in the context of large, deep modelsmaking it unsuitable for scenarios where frequent unlearning is required, such as user deletion requests or the detection of malicious data. To overcome this limitation, approximate unlearning techniques have been developed. These aim to adjust the existing model in a way that approximates the outcome of exact unlearning, while significantly reducing the associated computational overhead. By doing so, they strike a balance between privacy compliance and computational efficiency, enabling more scalable and responsive solutions to data removal.

In parallel, privacy concernsparticularly with sensitive data such as health-related data pointshave spurred the development of collaborative learning approaches, notably federated learning (FL). FL enables model training across multiple decentralized devices while ensuring that raw data remains

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

local, thereby addressing privacy concerns by minimizing the need to centralize sensitive information. However, while FL improves data privacy during training, it also introduces new challenges in ensuring data removal after training has commenced.

This leads to the emerging field of federated unlearning, which extends the principles of machine unlearning to the federated setting **?**. Federated unlearning aims to remove the influence of a client's data on the global model without requiring full retraining from scratch. This task is fundamentally more complex than traditional machine unlearning. In centralized ML, the model owner has direct access to both the training data and the model, making unlearning more straightforward. In contrast, Federated Learning (FL) uses a decentralized approach where data remains on clients' devices and only model updates are shared. This means unlearning in FL must function with limited information while respecting the communication and privacy constraints inherent to the federated system.

Moreover, Federated Learning requires a broader approach to unlearning: rather than removing individual data points, it often necessitates eliminating entire clients or groups of updates. This fundamental distinction demands different unlearning frameworks. Furthermore, the collaborative nature of FL means client updates become integrated in the global model, making it challenging to isolate and remove a single participant's effect.

Despite these challenges, federated unlearning is critical for enabling compliance with emerging data protection regulations and for preserving trust in FL systems. Efficient and principled federated unlearning methods are essential not only for meeting legal requirements but also for maintaining the robustness and reliability of FL models in the face of adversarial behaviors or user revocation requests.

GA: I think we should ~~discuss~~ what is our difference in comparison with other works.

Our main contributions are as follows:

- We extend the SCRUB framework for federated Unlearning to address ....
- Then, we improve the SCRUB performance inspired by f-divergences.
- Extensive experiments conducted to show performance of our approach.

2 Related Works

In this section, we discuss notable works in the federated unlearning.

Federated Unlearning: Federated unlearning (FU) can generally be categorized into two distinct scenarios: *active unlearning* and *passive unlearning*. In the active setting, the client requesting data removal actively participates in the unlearning process, assisting the system in mitigating its contribution to the global model. In contrast, passive unlearning assumes that the forgetting client is no longer available or unwilling to cooperate, requiring the remaining clients and the central server to collaboratively eliminate the influence of the departed clients data. As observed in the majority of federated unlearning literature, the underlying architecture typically involves a number of clients interacting via a central server, which orchestrates communication and model aggregation. Our focus aligns with this centralized federated setup, wherein unlearning mechanisms are implemented across distributed participants under server coordination.

As expected, most methods in passive unlearning attempt to reconstruct the model in the absence of the forgetting client. Due to the challenges and limitations associated with this setting, we concentrate our discussion on federated unlearning under the active scenario. Several early works, such as (**?**) and (**?**), addressed unlearning in convex optimization settings. However, given that most modern deep learning systems involve non-convex objectives, the conclusions drawn from these convex approaches are of limited utility in real-world deployments.

In the domain of federated unlearning, (**?**) proposed Forgettable Federated Linear Learning with Certified Data Removal , enabling provable data removal in linear models by exploiting their analytical properties. While computationally efficient and privacy-preserving in theory, the method requires sharing gradients and weights during unlearning, which may leak client information.

To address unlearning in more complex models,(**?**) introduced Goldfish, a framework that removes client influence without full retraining. It incorporates a novel loss balancing retained accuracy, removal bias, and confidence, offering an efficient and scalable unlearning solution.

? proposed Federated Unlearning, where a client locally reverses its contribution by maximizing empirical loss under constraints, followed by limited retraining across remaining clients. The method avoids storing historical updates but may leak information through shared unlearned model updates.

? integrates dataset distillation into the unlearning process, allowing clients to generate compact representations used for gradient ascent unlearning. This reduces computation but risks privacy leakage through shared distilled updates.

?combines confusion-based updates and salience-aware masking to weaken model memory of specific data. By simulating memory degradation and avoiding full retraining, it enables lightweight, instance- to client-level unlearning in federated settings.

3 Preliminaries

We begin by formalizing the problem of unlearning in the context of Federated Learning (FL). Let $\mathcal{A}$ denote a federated training algorithm such that the resulting global model $\theta \sim \mathcal{A}(S)$ is trained on a distributed dataset $S = \bigcup_{c=1}^{C} S^{(c)}$ across C clients, where $S^{(c)}$ is the local dataset of client c. In federated unlearning, each client may request the removal of a subset $S_F^{(c)} \subset S^{(c)}$, resulting in a retained subset $S_R^{(c)} = S^{(c)} \setminus S_F^{(c)}$. The global retained set is then $S_R = \bigcup_{c=1}^{C} S_R^{(c)}$.

Broadly, federated unlearning can be categorized into two main approaches:

Federated Exact Unlearning. An unlearning algorithm $\mathcal{U} : \Theta \times 2^{|S|} \to \Theta$ is said to achieve *exact unlearning* if it satisfies the following distributional equivalence:

$$\mathcal{U}(\mathcal{A}(S), \{S_F^{(c)}\}_{c=1}^{C}) \stackrel{d}{=} \mathcal{A}(S_R).$$

Here, $\stackrel{d}{=}$ may be interpreted in two ways: (1) **Parameter-level equivalence**, where the resulting model parameters are identical or nearly indistinguishable; or (2) **Performance-level equivalence**, where the models functional behavior is preserved with respect to downstream tasks. In this work, we adopt the performance-based perspective, prioritizing behavioral similarity over parameter similarity.

Federated Approximate Unlearning: Due to the high computational cost of retraining from scratch, approximate unlearning methods aim to efficiently remove the influence of S_F without full re-optimization. Its important to highlight that federated approximate unlearning (FAU) can have multiple interpretations depending on the context of our discussion. For instance, removing the data points of a single user differs significantly from eliminating the effects of poisoned data of a user or multiple users. These distinct objectives suggest that we need tailored metrics to evaluate the effectiveness of FAU. Depending on the underlying motivation, FAU methods generally fall into one of the following scenarios.

- **Robustness-Oriented Unlearning:** Designed to mitigate the impact of noisy, poisoned, or otherwise detrimental data, with the aim of improving the models generalization.
- **Privacy-Oriented Unlearning:** Focuses on eliminating the influence of specific data to comply with privacy regulations such as the GDPRs "Right to be Forgotten." Here, the goal is for the model to behave *as if* the data had never been used, often evaluated via privacy metrics like membership inference risk.

3.1 Scenario I: *Effect* Unlearning (Robustness-Oriented)

This scenario addresses the removal of data influence for reasons such as label noise or data poisoning, without necessitating full retraining.

Let $\theta = \mathcal{A}(S)$ be the global model trained using a federated algorithm such as Federated Averaging (FedAvg), where:

$$w_t = \sum_{c=1}^{C} \frac{n_c}{n} w_t^{(c)}, \quad \text{with } n_c = |S^{(c)}|, \quad n = \sum_{c=1}^{C} n_c.$$

Each local model $w_t^{(c)}$ is trained on client cs dataset $S^{(c)}$. Suppose each client identifies a forget set $S_F^{(c)}$, such that $S_R^{(c)} = S^{(c)} \setminus S_F^{(c)}$.

the ideal performance of the unlearned model should remain competitive with or even lesser that of the exact unlearning baseline:

$$\mathcal{R}_R\left(\mathcal{U}\left(\mathcal{A}(S), \{S_F^{(c)}\}_{c=1}^C\right)\right) \leq \mathcal{R}_R\left(\mathcal{A}(S_R)\right)$$

3.2 Scenario II: *Data* Unlearning (Data Privacy-Oriented)

In this scenario, unlearning is motivated by privacy concerns, where clients demand the deletion of specific personal data in accordance with legal or ethical obligations (e.g., GDPR, CCPA). The objective is to ensure that the resulting global model behaves *as if* the forgotten data $S_F = \bigcup_{c=1}^C S_F^{(c)}$ had never been used during training.

Privacy Risk: Let $\theta = \mathcal{A}(S)$ denote the model trained on the full dataset, and let $\tilde{\theta} = \mathcal{U}(\theta, \{S_F^{(c)}\}_{c=1}^C)$ be the model after unlearning. The goal is to minimize the distinguishability between $\tilde{\theta}$ and $\mathcal{A}(S_R)$, where $S_R = \bigcup_{c=1}^C S_R^{(c)}$.

A common metric for evaluating privacy preservation is the risk of **membership inference attacks** (MIA), where an adversary attempts to determine whether a given data point was part of the training set. For the unlearned model $\tilde{\theta}$, we define the MIA advantage as:

$$\text{Adv}_{\text{MIA}}(\tilde{\theta}, S_F) = \sup_{x \in S_F} \left| \Pr[\mathcal{A}(S) \text{ trained on } x] - \Pr[\tilde{\theta} \text{ trained on } x] \right|.$$

Unlearning Objective: To ensure privacy-compliant unlearning, the algorithm must guarantee:

$$\text{Adv}_{\text{MIA}}(\tilde{\theta}, S_F) \approx 0, \tag{1}$$

while simultaneously maintaining utility on the retained data, ideally they would be equal however the inequality below would hold.

$$\mathcal{R}_R\left(\mathcal{A}(S_R)\right) \leq \mathcal{R}_R\left(\mathcal{U}\left(\mathcal{A}(S), \{S_F^{(c)}\}_{c=1}^C\right)\right) \tag{2}$$

We would very much prefer that

$$\mathcal{R}_R\left(\mathcal{A}(S_R)\right) \approx \mathcal{R}_R\left(\mathcal{U}\left(\mathcal{A}(S), \{S_F^{(c)}\}_{c=1}^C\right)\right) \tag{3}$$

3.3 Illustrative Example: Distinguishing Between Robustness and Privacy-Oriented Unlearning

To better understand the two application scenarios described above, consider the following example.

Assume a federated learning setup with $C = 5$ clients, each holding $10{,}000$ local training examples. Suppose that two of the clients discover that 20% of their training data is mislabeled due to a data collection error. They notify the server, which initiates an unlearning procedure to remove the influence of the corrupted samples.

In this *robustness-oriented* scenario, the unlearning process is expected to *improve* the generalization performance of the global model, as it eliminates harmful or misleading examples (e.g., noisy or poisoned data). The previously trained model, influenced by this corrupted data, likely exhibited degraded performance. An ideal unlearning algorithm would reverse this effect, resulting in a more accurate and robust model.

Now consider a different case in which the same two clients are legally required to erase 20% of their data due to privacy regulations, such as the GDPRs Right to be Forgotten. Unlike the previous scenario, the data in question is not erroneous but entirely valid and potentially useful for model training. However, continued use of this data could result in legal consequences for both the clients and the service provider.

Table 1: Divergences and their corresponding generator functions

Divergence	Generator Function $f(t)$
KL-divergence	$t \log t$
χ^2-divergence	$(1-t)^2$
JS-divergence	$t \log\left(\frac{2t}{1+t}\right) + \log\left(\frac{2}{1+t}\right)$

In this *privacy-oriented* scenario, unlearning is not expected to improve model performanceindeed, removing useful data may degrade it. The primary goal here is to ensure that the resulting model exhibits no detectable influence from the erased data. Specifically, the model must be resistant to any form of Membership Inference Attack (MIA) that could reveal whether a particular datapoint was part of the training process. In legal contexts, such as courtroom investigations, this privacy guarantee serves as proof of compliance.

4 Methodology

Let $\mathcal{A}$ denote a centralized federated learning algorithm that outputs a model $\theta \sim \mathcal{A}(S)$ trained on a distributed dataset $S = \bigcup_{c=1}^{C} S^{(c)}$. Each client c partitions its local data into retained and forget subsets, $S_R^{(c)}$ and $S_F^{(c)}$, respectively. The objective is to obtain an unlearned model θ^u such that

$$\theta^u \approx \mathcal{A}(S_R), \quad S_R := \bigcup_{c=1}^{C} S_R^{(c)}.$$

Fed-f-SCRUB performs T server-coordinated communication rounds, each consisting of a maximization (forgetting) step and a minimization (retention) step. This is followed by T_{post} additional minimization-only rounds.

Figure 1:

4.1 Local Objectives

Let $d_f(x; w, w_T)$ be an f-divergence between model outputs of w and a reference model w_T on input x, and let $\ell(h(x; w), y)$ be the prediction loss. For client c, we define:

$$\mathcal{L}_{\max}^{(c)}(w; w_T) = \frac{1}{|S_F^{(c)}|} \sum_{x_f \in S_F^{(c)}} d_f(x_f; w, w_T),$$

$$\mathcal{L}_{\min}^{(c)}(w; w_T) = \frac{\alpha}{|S_R^{(c)}|} \sum_{x_r \in S_R^{(c)}} d_f(x_r; w, w_T) + \frac{\gamma}{|S_R^{(c)}|} \sum_{(x_r, y_r)} \ell(h(x_r; w), y_r).$$

where $d_f(x; p, q)$ defined as:

$$D_f(p \parallel q) := \sum_{x \in \mathcal{X}} q(x) f\left(\frac{p(x)}{q(x)}\right)$$

where $f : (0, \infty) \to \mathbb{R}$ is a convex function with the property that $f(1) = 0$ and P, Q are two discrete distribution. This definition implies that $D_f(P \parallel Q) = 0$ if and only if $P = Q$. By choosing different forms for f, we obtain different types of divergences. For example, when $f(t) = t \log(t)$, we get the Kullback-Leibler (KL) divergence, which measures the difference between two probability distributions. In this work, we focus on JS-divergence and χ^2-diveregence. The definition of their generator functions are in the table **??**.

4.2 Federated Updates (Gradient Version)

At round t, the server broadcasts model w^t and instructs clients to sequentially perform:

$$g_{\max}^{(c)} = \nabla_w \mathcal{L}_{\max}^{(c)}(w_T; w_T), \quad w_{\max}^{(c)} = w_T + \eta_{\max} \cdot g_{\max}^{(c)}, \quad w^{t+\frac{1}{2}} = \sum_{c=1}^{C} \frac{n_f^{(c)}}{n_f} w_{\max}^{(c)},$$

Algorithm 1 Fed-f-SCRUB: Federated Unlearning via Divergence Optimization (Gradient Version)

Require: Initial model w^0, number of unlearning rounds T, post-training rounds T_{post}, loss weights α, γ, learning rates $\eta_{\max}$, $\eta_{\min}$

for $t = 0$ to $T-1$ **do**
  Server selects teacher model $w_T \in \{w^0, w^t\}$ ▷ Maximization (Forgetting) Phase
  for all clients $c \in [C]$ in parallel **do**
    $g_{\max}^{(c)} \leftarrow \nabla_w \mathcal{L}_{\max}^{(c)}(w_T; w_T)$
    $w_{\max}^{(c)} \leftarrow w_T + \eta_{\max} \cdot g_{\max}^{(c)}$
  end for
  $w^{t+\frac{1}{2}} \leftarrow \sum_{c=1}^{C} \frac{n_f^{(c)}}{n_f} w_{\max}^{(c)}$
  ▷ Minimization (Retention) Phase
  for all clients $c \in [C]$ in parallel **do**
    $g_{\min}^{(c)} \leftarrow \nabla_w \mathcal{L}_{\min}^{(c)}(w^{t+\frac{1}{2}}; w_T)$
    $w_{\min}^{(c)} \leftarrow w^{t+\frac{1}{2}} - \eta_{\min} \cdot g_{\min}^{(c)}$
  end for
  $w^{t+1} \leftarrow \sum_{c=1}^{C} \frac{n_r^{(c)}}{n_r} w_{\min}^{(c)}$
end for
for $\tau = 0$ to $T_{\text{post}} - 1$ **do**
  Server selects $w_T \in \{w^0, w^{T+\tau}\}$
  for all clients $c \in [C]$ in parallel **do**
    $g_{\min}^{(c)} \leftarrow \nabla_w \mathcal{L}_{\min}^{(c)}(w^{T+\tau}; w_T)$
    $w_{\min}^{(c)} \leftarrow w^{T+\tau} - \eta_{\min} \cdot g_{\min}^{(c)}$
  end for
  $w^{T+\tau+1} \leftarrow \sum_{c=1}^{C} \frac{n_c}{n} w_{\min}^{(c)}$
end for
return Final unlearned model $w^{T+T_{\text{post}}}$

$$g_{\min}^{(c)} = \nabla_w \mathcal{L}_{\min}^{(c)}(w^{t+\frac{1}{2}}; w_T), \quad w_{\min}^{(c)} = w^{t+\frac{1}{2}} - \eta_{\min} \cdot g_{\min}^{(c)}, \quad w^{t+1} = \sum_{c=1}^{C} \frac{n_r^{(c)}}{n_r} w_{\min}^{(c)}.$$

The reference model w_T may be either the original model w^0 or the current round model w^t. The former is natural for short local updates, while the latter may be preferable for longer local training.

4.3 Post-Training Minimization (Gradient Version)

After T unlearning rounds, the model undergoes T_{post} rounds of minimization-only updates:

$$g_{\min}^{(c)} = \nabla_w \mathcal{L}_{\min}^{(c)}(w^{T+\tau}; w_T), \quad w_{\min}^{(c)} = w^{T+\tau} - \eta_{\min} \cdot g_{\min}^{(c)}, \quad w^{T+\tau+1} = \sum_{c=1}^{C} \frac{n_c}{n} w_{\min}^{(c)}, \quad \tau = 0, \ldots, T_{\text{post}} - 1.$$

No further maximization is performed in this phase.

A Empirical Results

A.1 Baselines

Based on the two main scenarios addressed in our federated unlearning (FU) framework, we select two corresponding baselines for comparison. First, we consider the approach in **?**, which primarily focuses on privacy by enabling the removal of a clients entire dataset. In contrast, our framework offers a finer granularity by allowing partial data removal from a client. Second, for evaluating the removal of poisoned data, we adopt the method proposed in **?** as our baseline, where the focus is on mitigating the effects of backdoor attack datapoints.

A.2 Simulation details

A.3 Evaluation Metrics

For evaluating privacy, we employ Membership Inference Attacks (MIAs), including well-known variants such as Shokris attack **?** and Yeoms attack**?**. In the context of poisoned or noisy data, following the literature, we use classification accuracy and backdoor attack success rate as the primary evaluation metrics.

B More Related Works

GA: combine these two paragraphs

Machine Unlearning: As previously mentioned, a foundational mathematical framework for machine unlearning has been developed using principles inspired by differential privacy. While this framework has led to significant progress, its success has largely been restricted to convex optimization problems. Unfortunately, such formulations are not directly applicable to modern deep learning models, which typically involve non-convex objectives and are prone to model memorizationa phenomenon where models retain specific training data rather than learning generalizable patterns. This memorization poses a substantial challenge to unlearning, as sensitive or malicious data embedded in model parameters may persist even after standard removal techniques. Moreover, recent research has shown that it is possible to obtain arbitrarily similar model weights when training on two non-overlapping datasets. This observation implies that reaching a particular point in parameter space does not guarantee effective unlearning, since memorized data influences may not be fully erased. In addition to these theoretical insights, there has been considerable progress in developing unlearning algorithms for classical machine learning models, further illustrating the diverse and evolving landscape of this field.

GA: ???

A growing body of research has explored data-driven approaches to machine unlearning, aiming to efficiently remove the influence of specific data points. One recent work, [X], employs data attribution techniques to identify and eliminate the effects of targeted data points; however, this approach risks leaking the data of "forgetting" clients to others, making it less suitable for federated unlearning scenarios, which we do not emphasize here. Another notable method, SCRUB, has been developed to enhance stability during the unlearning process, offering a more robust alternative. Parallel to these efforts, a distinct line of work focuses on sparsity-regularized fine-tuning and partial fine-tuning, leveraging model sparsity to reduce computational overhead while unlearning. Additionally, several studies have adopted Bayesian and variational inference techniques to estimate the impact of forgetting data points, providing probabilistic frameworks for unlearning. These diverse approaches underscore the multifaceted nature of machine unlearning, balancing efficiency, privacy, and model integrity across different contexts.

Machine Unlearning: Two primary frameworks have emerged to address the challenge of unlearning: exact unlearning (**?**) and approximate unlearning (**?**). Exact unlearning requires retraining the model from scratch using only the remaining data, but this approach is computationally expensive and impractical for large-scale models (**?**). In contrast, approximate unlearning modifies the trained model to mimic the outcome of retraining on the remaining dataset. The key challenge in approximate unlearning is to ensure that the modified model is indistinguishable from a retrained one, often necessitating theoretical guarantees on the quality of the approximation (**?**).

Although much of the unlearning research has focused on convex models (**?**), the non-convexity of deep neural networks complicates the process. As a result, effective unlearning remains a challenge,

with heuristics often producing varying results across different benchmarks, making it difficult to ensure consistent reliability (**?**).

(**?**) highlight a significant challenge in fine-tuning-based unlearning methods, known as the *missing targets* problem. When unlearning a data point $x \in$ *forget set*, these methods typically apply gradient ascent on x and gradient descent on the retain set to preserve model performance. However, gradient ascent can cause the loss on x to grow indefinitely if unchecked. The desired outcome is to stop when the model's loss on x matches the counterfactual loss (i.e., the loss of a model trained only on the retain set). This presents two main issues: (a) the target loss is unknown, and (b) the optimal stopping point may vary for different points in the forget set. As a result, unlearning algorithms often "undershoot" or "overshoot" the target loss (**?**).

This problem is further analyzed in the work of (**?**), which uses data modeling to address these challenges. Our research seeks to extend SCRUB to overcome this issue by introducing a loss function that is naturally robust to overshooting and undershooting by employing various f-divergences.

While f-divergences have been effective loss functions in various machine learning tasks (**????**), they have been primarily used for validating machine unlearning processes. For example, Jensen-Shannon (JS) divergence has been applied in the context of unlearning to validate the removal of data from models (**?**), (**?**), (**?**). Furthermore, there has been some exploration of using f-divergences specifically for unlearning large language models (LLMs) (**?**).

C Motivations for JS divergence and χ^2-divergence

In this section, we study some motivations behind choosing JS divergence and χ^2 divergence. These information measures offer several advantages over KL divergence, particularly in applications involving generative modeling and robust regularization.

JS divergence is widely used as a loss function in Generative Adversarial Networks (GANs) due to its symmetric and bounded nature, which provides a stable measure of similarity between distributions (**?**). Unlike KL divergence, which can diverge to infinity when the two distributions have disjoint supports, JS divergence remains finite and well-behaved, making it particularly effective for comparing empirical distributions ((**?**)). This property is especially beneficial in our context, as it helps mitigate overshoot and undershoot problems, particularly in scenarios where exact loss values for removed data points are unavailable.

On the other hand, χ^2 divergence emphasizes large discrepancies due to its squared difference term, making it particularly useful in outlier detection and robust learning frameworks (**?**). Regularizing with χ^2 divergence can also help prevent models from becoming overly biased toward majority classes by strongly penalizing large probability gaps (**?**). This property makes it particularly effective in imbalanced learning scenarios, where standard loss functions may fail to capture significant disparities between class distributions.

Thus, by leveraging JS divergence for stable probability comparisons and χ^2 divergence for strong regularization and outlier sensitivity, we can achieve a more robust and balanced learning framework compared to using KL divergence alone.

Building on this, we modify our loss functions and introduce f-SCRUB, where we select different f-divergences for the retain set and the forget set. Each divergence term, $d(x_r; w^u)$ and $d(x_f; w^u)$, can be chosen from JS, KL, or χ^2 divergences ((**?**)).

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (12 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: **[TODO]**

Justification: **[TODO]**

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so No is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: **[TODO]**

    Justification: **[TODO]**

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.