CFA: LIGHTWEIGHT DEFENSE AGAINST MEMBERSHIP INFERENCE ATTACKS THROUGH CLASS-WISE FEATURE AGGREGATION

Ao Dong, Zhihao Li, Yifei Tian, Xiaobai Chen, Jieming Yin

Nanjing University of Posts and Telecommunications, Nanjing 210023, China

ABSTRACT

As membership inference attacks increasingly threaten the privacy of machine learning models, existing defenses either lack formal guarantees or incur significant utility and computational costs. We propose *Class-wise Feature Aggregation* (CFA), a novel and lightweight privacy-preserving mechanism that shifts the defense perspective from gradient space to feature space. CFA performs per-sample normalization, class-wise aggregation, and Gaussian noise injection, offering strong privacy guarantees under Rényi Differential Privacy while maintaining high utility. Unlike prior certified approaches such as DP-SGD, CFA avoids expensive per-example gradient computations and integrates seamlessly into standard training pipelines without architectural modifications. CFA achieves accuracy within 1-5% of non-private models (versus DP-SGD's 15-50% drops) without increasing training time. By outperforming empirical defenses and establishing feature-space perturbation as a deployable solution, CFA bridges the gap between theoretical privacy guarantees and practical deployment requirements.

***Index Terms*—** Differential Privacy, Membership Inference Attack, Machine Learning

1. INTRODUCTION

As deep neural networks pervade sensitive domains (e.g., medical, finance), their privacy implications have attracted significant attention [1, 2]. In particular, adversaries have been shown to be able to exploit a trained model to infer whether a specific sample was used during training, a type of privacy breach known as *membership inference attack* (MIA) [2]. These threats have made MIA resistance a key task for privacy-preserving machine learning systems [3].

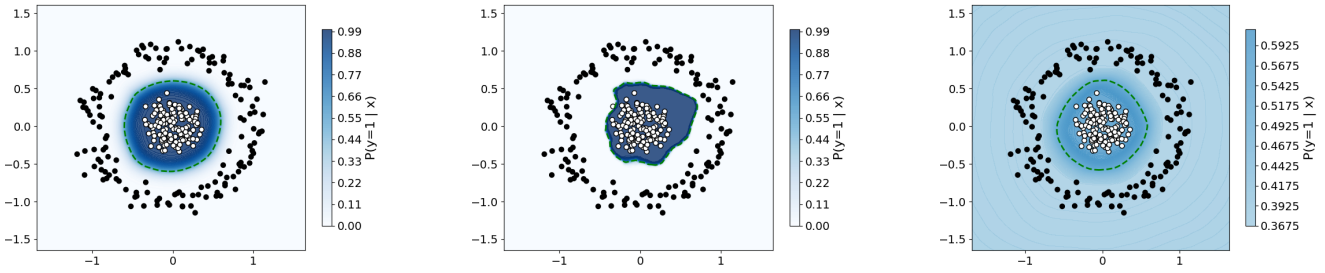
To mitigate MIAs, prior work has explored both empirical defenses and certified mechanisms. Empirical defenses try to hide the statistical cues that MIAs might exploit. Representative methods include adversarial or noisy perturbations to logits (e.g., MemGuard [4], DynaNoise [5]), ensemble training with multiple sub-models (e.g., SELENA [6], MIST [7]), and training/inference modifications such as label smoothing or adaptive mixup [8, 9]. Others leverage generative or augmentation techniques at inference time [10, 11]. Although sufficiently accurate, these methods lack formal privacy guarantees. Therefore, they are vulnerable to adaptive attacks [12], and typically require retraining, ensembles, or complex preprocessing, limiting their scalability. Certified mechanisms, in contrast, enforce provable protection by adding calibrated randomness. DP-SGD [1] perturbs clipped per-sample gradients, but incurs accuracy loss and high training overhead. Data-level methods, such as ImageDP [13] and PPGF [14], privatize inputs via generative encoders, but complicate integration and often reduce utility [15].

In this work, we design a mechanism to achieve the best of both worlds, i.e., preserving a provable privacy guarantee while offering high inference accuracy. A key limitation of existing certified mechanisms is that they perturb high-dimensional gradients that are both noise-sensitive and critical for convergence. Doing so leads to misaligned optimization [16], poor privacy-utility tradeoffs, and heavy per-sample gradient costs [15, 17]. We instead enforce privacy in the feature space, where representations are lower-dimensional, semantically richer, and better aligned with classification, allowing models to adapt more easily to noise. However, directly perturbing features introduces excessive noise variance as they lack the natural averaging effect of gradients. To address this, we propose aggregating features from samples with the same label within a mini-batch. This preserves intra-class semantics and decision boundaries while forming optimal "privacy units" for noise injection with reduced variance.

Based on these insights, we introduce a novel mechanism named Class-wise Feature Aggregation (CFA). CFA enforces privacy in the feature space through per-sample feature norm clipping, class-wise aggregation of features, and Gaussian noise injection to the aggregated groups. These operations are implemented as a lightweight, parameter-free module that can be seamlessly integrated into standard training pipelines, offering a favorable trade-off between privacy and utility. To illustrate CFA's effectiveness, Figure 1 shows decision confidence maps on the concentric circles dataset. The non-private model (Figure 1a) exhibits a tight, data-dependent boundary with overconfident predictions, indicating overfitting and

This work was supported in part by the National Nature Science Foundation of China Grant No. 92473205, National Key R&D Program of China Grant No. 2023YFB4404400, Jiangsu Province Major Scientific Project Grant No. BG2024032.

Jieming Yin (jieming.yin@njupt.edu.cn) is the corresponding author.



(a) Original model (undefended) (b) DP-SGD method (c) CFA (our method)

Fig. 1: Decision confidence maps on the concentric circles dataset. Black and white dots represent samples from the outer and inner circles, respectively. The background color indicates the model's confidence in predicting the inner class, with bluer regions indicating higher confidence. The green dashed contour denotes the decision boundary.

vulnerability to MIAs. DP-SGD (Figure 1b) mitigates memorization via noisy gradient updates, but introduces unstable boundaries and accuracy loss. In contrast, CFA (Figure 1c) flattens the confidence landscape, restricting outputs to [0.36, 0.59], which suppresses membership cues while retaining smooth, accurate decision boundaries. Our main contributions are as follows.

- We propose a provable privacy-preserving mechanism in the feature space, supported by Rényi DP analysis, achieving protection comparable to DP-SGD while outperforming state-of-the-art empirical defenses.
- We introduce a class-wise aggregation strategy that enables group-level noise injection while preserving per-class feature separation, thus minimizing utility loss.
- We design a lightweight, non-intrusive, parameter-free module that requires no architectural modifications and incurs negligible training overhead.

2. METHODOLOGY

2.1. Overall Framework

Our design is based on a key observation: typical machine learning models can be abstracted into a feature extractor f_{enc} and a classifier f_{cls}. The proposed Class-wise Feature Aggregation module is inserted between these two components. As shown in Figure 2, CFA contains three lightweight and non-parametric layers: a **Feature Normalization Layer** performing layer normalization (active in training/inference), a **Class-wise Aggregation Layer** grouping features within each class (active in training), and a **Noise Layer** injecting Gaussian noise (active in training); these layers are jointly required for the certified privacy guarantee. The CFA module is placed between the feature extractor and the classifier, where deeper features are better aligned with class semantics, ensuring aggregation preserves discriminative power. This position also allows seamless integration into standard classification models without architectural modifications. The aggregation algorithm is summarized in Algorithm 1.

2.2. Step 1: Feature Normalization

As described in Algorithm 1 (lines 1–7), each feature vector $x \in \mathbb{R}^d$ is normalized on a per-sample basis. Specifically, we subtract its mean and scale by its standard deviation to obtain a zero-centered representation $\tilde{x}$, which is further rescaled so that $\|\tilde{x}\|_2 = C$. This guarantees that every sample lies within an ℓ_2 ball of radius C, providing a fixed sensitivity bound for subsequent aggregation in the privacy analysis. In addition to satisfying the differential privacy requirement, this step empirically stabilizes the feature distribution and improves training convergence. For instance, on CIFAR10 we observe nearly a 10% accuracy drop when normalization is removed. To maintain consistency, normalization is applied during both training and inference.

2.3. Step 2: Class-wise Aggregation

In this step, CFA groups the normalized features $\tilde{x}$ by their class labels and computes the class-wise mean μ_i for each group. Unlike global batch aggregation that collapses feature space and harms classification, our class-specific approach preserves discriminative boundaries.

Moreover, this aggregation step corresponds precisely to the mechanism $\mathcal{M}$ in Rényi Differential Privacy, which assumes group-level computation to bound sensitivity.

Let $G_i = \{\tilde{x}_1, \ldots, \tilde{x}_{n_i}\}$ denote the normalized features with label i. To bound sensitivity, consider replacing one element $\tilde{x}_j \in G_i$ by another $\tilde{x}'_j$, both satisfying $\|\cdot\|_2 = C$. Then:

$$\|\mu_i - \mu'_i\|_2 = \left\| \tfrac{1}{n_i}(\tilde{x}_j - \tilde{x}'_j) \right\|_2 \leq \tfrac{2C}{n_i} \tag{1}$$

Thus, the ℓ_2 sensitivity of the aggregation mechanism is $\Delta_i = 2C/n_i$.

2.4. Step 3: Gaussian Noise Injection

To ensure differential privacy, as described in Algorithm 1 (lines 14–17), we perturb each class mean μ_i by adding Gaussian noise with standard deviation $\frac{\lambda C}{n_i}$ in every dimension. where n_i is the number of samples in the i-th class within the

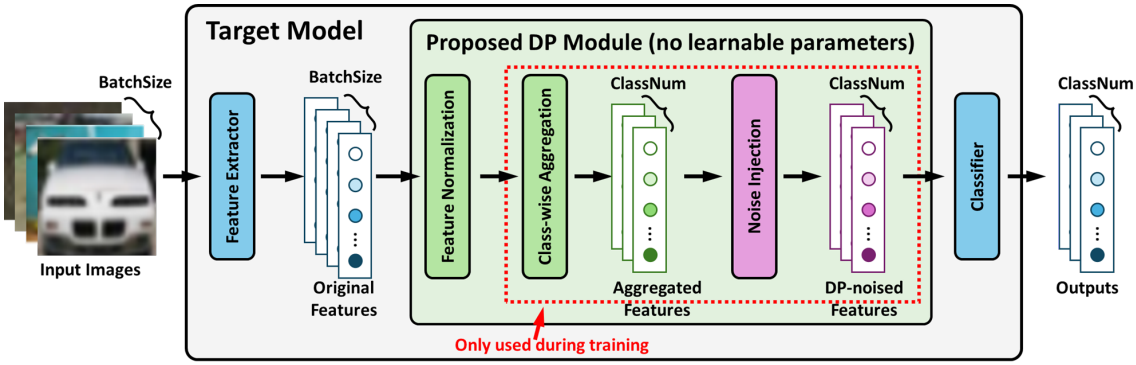


Fig. 2: Overview of the CFA framework.

batch. This dedicated setting is inspired by the corresponding noise formulation from DP-SGD. In both cases, the injected noise is calibrated to the sensitivity of the aggregated quantity (gradients in DP-SGD, and class-wise features in our method), which scales inversely with the group size.

By the Gaussian mechanism [18], a function with sensitivity Δ_i and noise std $\sigma = \lambda C/n_i$ satisfies

$$\varepsilon_{\text{set}} = \frac{\alpha\Delta_i^2}{2\sigma^2} = \frac{\alpha(2C/n_i)^2}{2(\lambda C/n_i)^2} = \frac{2\alpha}{\lambda^2} \tag{2}$$

Notably, n_i and C cancel out, so the per-class privacy cost depends only on the noise multiplier λ.

By parallel composition across classes, subsampling with per-class rates, and T iterations, the total RDP cost is

$$\varepsilon_T \leq T \cdot \frac{2\alpha q_{\max}^2}{\lambda^2} \tag{3}$$

where $q_{\max} = \max_i(b_i/n_i)$. For balanced data ($n_i = N/k$, $b_i = b/k$) this reduces to $q_{\max} = b/N$.

Finally, converting to (ε, δ)-DP [18] gives

$$\varepsilon = \varepsilon_T + \frac{\log(1/\delta)}{\alpha - 1} \tag{4}$$

This is the only stochastic component in our method and introduces a tunable hyperparameter λ. The value of λ determines the overall privacy budget (ε, δ).

Algorithm 1 Class-wise Feature Aggregation (in training)

Input: Feature-labeled batch $\mathcal{B} = \{(x_1, y_1), \ldots, (x_b, y_b)\}$, where $x_i \in \mathbb{R}^d$, b is the batch size, and d is the feature dimension; Normalization scale $C > 0$, noise scale $\lambda > 0$

Output: DP-protected class-wise means $\{\hat{\mu}_i\}_{i=1}^k$, where k is the number of classes

Step 1: Feature Normalization
for all $x \in \mathcal{B}$ **do**
  Let $x = [x^{(1)}, x^{(2)}, \ldots, x^{(d)}]$
  Compute mean: $\mu(x) = \frac{1}{d}\sum_{j=1}^{d} x^{(j)}$
  Compute variance: $\tau^2(x) = \frac{1}{d}\sum_{j=1}^{d}\left(x^{(j)} - \mu(x)\right)^2$
  Normalize: $\tilde{x} \leftarrow \frac{C}{\sqrt{d}} \cdot \frac{x - \mu(x)}{\sqrt{\tau^2(x)}}$
end for
Step 2: Class-wise Mean Aggregation
for all classes $i \in \{1, \ldots, k\}$ **do**
  $G_i \leftarrow \{\tilde{x} \mid (\tilde{x}, y) \in \mathcal{B},\ y = i\}$
  $n_i \leftarrow |G_i|$
  $\mu_i \leftarrow \frac{1}{n_i}\sum_{\tilde{x} \in G_i} \tilde{x}$
end for
Step 3: Gaussian Noise Injection
for all classes $i \in \{1, \ldots, k\}$ **do**
  $\hat{\mu}_i \leftarrow \mu_i + \mathcal{N}\left(0, \left(\frac{\lambda C}{n_i}\right)^2 I_d\right)$
end for
return $\{\hat{\mu}_i\}_{i=1}^k$

2.5. Final Classification

The perturbed class means $\{\hat{\mu}_i\}_{i=1}^k$ are passed to the classifier during training, so the effective batch size becomes k. During inference, aggregation and noise layers are disabled, and the classifier directly receives the normalized feature $\tilde{x}$, ensuring consistency.

3. EXPERIMENTS

This section presents comprehensive experiments to evaluate our proposed CFA mechanism from three key perspectives: (1) privacy protection against membership inference attacks, (2) classification performance, and (3) training overhead. We perform experiments across a range of datasets and model architectures to demonstrate both the effectiveness and generality of our approach.

3.1. Experimental Environment and Datasets

All experiments are executed on an NVIDIA RTX 4080 Super GPU. We use standard SGD with momentum 0.9 and weight decay 5×10^{-4} for optimization, and adopt cosine annealing as the learning rate schedule. The number of training epochs is 100. We treat the last layer of the model as the classifier, and all preceding layers as the feature extractor. In CFA, we

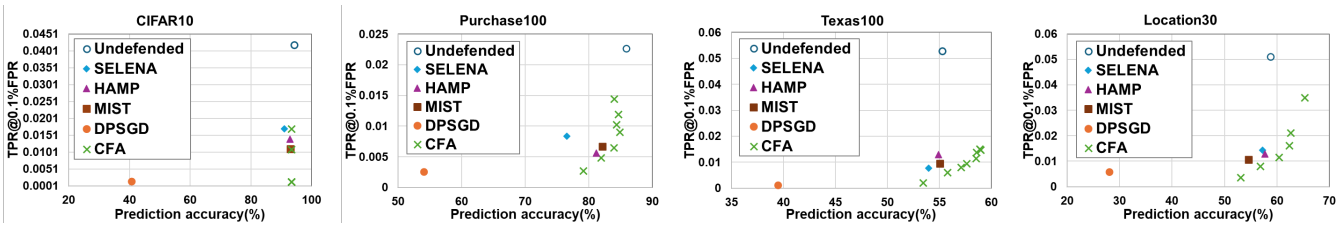


Fig. 3: Privacy leakage (TPR@0.1%FPR) vs. classification accuracy on four datasets. Lower TPR and higher accuracy are preferred (i.e., toward the bottom-right corner).

clip each feature vector to have ℓ_2 norm at most C, analogous to the gradient clipping in DP-SGD. While C does not affect the formal privacy guarantee, it influences model accuracy; following prior work [1], we adopt a small set of fixed values (e.g., 1.0 or 10.0) for reproducibility. We evaluate CFA on four widely used datasets covering both image and tabular modalities, including CIFAR10 [19], Purchase100 [2], Texas100 [2], and Location30 [2].

3.2. Attacks and Defenses

To evaluate privacy leakage, we use the LiRA attack [12], a powerful membership inference method based on shadow model calibration. Specifically, we use the online variant with fixed variance for all settings. Following standard practice [12], we report TPR@0.1%FPR as our privacy metric. This metric measures the true positive rate (i.e., fraction of training samples correctly identified as members) when the attacker is constrained to only 0.1% false positive rate. This setting reflects real-world scenarios where attackers are only willing to act on highly confident predictions. We evaluate CFA against four strong baseline defenses, including DP-SGD [1]($\varepsilon \approx 8.0$), the canonical differentially private training algorithm, as well as three SOTA empirical methods: SELENA [6], HAMP [8], and MIST [7].

3.3. Privacy Protection and Accuracy

To ensure a fair comparison, we report each method using representative hyperparameters that reflect a practical trade-off between privacy and utility. This avoids overly weak baselines or unreasonably strong noise levels.

Figure 3 presents a comprehensive comparison of different defenses, showing their privacy leakage (measured by TPR@0.1%FPR) versus classification accuracy across four benchmark datasets. In each figure, the bottom-right corner indicates both lower privacy leakage and higher accuracy. CFA is represented by multiple points under different parameter settings (batch size and noise scale). CFA consistently appears closer to this ideal than existing empirical defenses, which tend to exhibit significantly higher TPRs under attack.

While DP-SGD provides the strongest privacy protection (0.0024 TPR@0.1%FPR on average), it incurs a severe utility cost, reducing accuracy by over 33% on average. Such degradation often renders the model impractical in real-world scenarios. Empirical defenses offer better utility (1.6%–3.4% accuracy drop), but at the cost of weaker privacy, with TPRs ranging from 0.0084 to 0.0110. In contrast, CFA offers a compelling trade-off: it achieves formal (ε, δ)-differential privacy at the feature level while maintaining high utility, with only a 0.24% average accuracy drop and a TPR of 0.0061. On certain datasets, such as Texas and Location, CFA even improves classification accuracy over the baseline, likely due to the regularization effect of class-wise aggregation.

3.4. Training Overhead

To assess training efficiency, we compare the total wall-clock training time among different defense methods on CIFAR10. Our proposed CFA method incurs virtually no additional overhead compared to the undefended baseline (0.38h). In contrast, DP-SGD requires 3.91h, representing a $10.3\times$ slowdown and a $17.6\times$ increase in memory usage due to per-sample gradient computation. MIST also slows training to 2.52h ($6.6\times$ overhead) with about $4\times$ memory cost, while HAMP moderately increases the time to 1.51h ($4.0\times$). SELENA in sequential mode is the most expensive, taking 8.78h ($23.1\times$), since it trains and distills multiple submodels. Although parallel SELENA reduces runtime via multi-GPU execution, its memory consumption grows linearly with the number of models held in memory.

4. CONCLUSION

We presented CFA, a simple yet principled defense against membership inference attacks that operates in the feature space. By combining per-sample normalization, class-wise aggregation, and calibrated Gaussian perturbation, CFA achieves certified privacy under Rényi Differential Privacy with minimal impact on model utility and efficiency. Our approach requires no architectural changes and integrates easily into standard training pipelines, making it well-suited for privacy-sensitive applications. Beyond its empirical effectiveness and theoretical soundness, CFA illustrates a promising design paradigm: *structured perturbation in the latent space* as a practical and scalable alternative to traditional gradient-based defenses. We believe this perspective opens up new opportunities for rethinking privacy-preserving learning at the representation level.

5. REFERENCES

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 308–318.

[2] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 3–18.

[3] Han Liu, Yuhao Wu, Zhiyuan Yu, and Ning Zhang, "Please tell me more: Privacy impact of explainability through the lens of membership inference attack," in *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2024, pp. 4791–4809.

[4] Jinyuan Jia, Ahmed Salem, Michael Backes, Yang Zhang, and Neil Zhenqiang Gong, "Memguard: Defending against black-box membership inference attacks via adversarial examples," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 259–274.

[5] Javad Forough and Hamed Haddadi, "Dynanoise: Dynamic probabilistic noise injection for defending against membership inference attacks," *arXiv preprint arXiv:2505.13362*, 2025.

[6] Xinyu Tang, Saeed Mahloujifar, Liwei Song, Virat Shejwalkar, Milad Nasr, Amir Houmansadr, and Prateek Mittal, "Mitigating membership inference attacks by self-distillation through a novel ensemble architecture," in *31st USENIX Security Symposium (USENIX Security 22)*, 2022, pp. 1433–1450.

[7] Jiacheng Li, Ninghui Li, and Bruno Ribeiro, "MIST: Defending against membership inference attacks through Membership-Invariant subspace training," in *33rd USENIX Security Symposium (USENIX Security 24)*, 2024, pp. 2387–2404.

[8] Zitao Chen and Karthik Pattabiraman, "Overconfidence is a dangerous thing: Mitigating membership inference attacks by enforcing less confident prediction," *arXiv preprint arXiv:2307.01610*, 2023.

[9] Ying Chen, Jiajing Chen, Yijie Weng, ChiaHua Chang, Dezhi Yu, and Guanbiao Lin, "Adamixup: A dynamic defense framework for membership inference attack mitigation," *arXiv preprint arXiv:2501.02182*, 2025.

[10] Yuefeng Peng, Ali Naseh, and Amir Houmansadr, "Diffence: Fencing membership privacy with diffusion models," *arXiv preprint arXiv:2312.04692*, 2023.

[11] Heqing Ren, Chao Feng, Alberto Huertas, and Burkhard Stiller, "Augmixcloak: A defense against membership inference attacks via image transformation," *arXiv preprint arXiv:2505.07149*, 2025.

[12] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer, "Membership inference attacks from first principles," in *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2022, pp. 1897–1914.

[13] Hanyu Xue, Bo Liu, Ming Ding, Tianqing Zhu, Dayong Ye, Li Song, and Wanlei Zhou, "Dp-image: Differential privacy for image data in feature space," *arXiv preprint arXiv:2103.07073*, 2021.

[14] Ruikang Yang, Jianfeng Ma, Yinbin Miao, and Xindi Ma, "Privacy-preserving generative framework for images against membership inference attacks," *IET Communications*, vol. 17, no. 1, pp. 45–62, 2023.

[15] Zhiqi Bu, Jinshuo Dong, Qi Long, and Weijie J Su, "Deep learning with gaussian differential privacy," *Harvard Data Science Review*, vol. 2020, no. 23, pp. 10–1162, 2020.

[16] Antoine Barczewski and Jan Ramon, "DP-SGD with weight clipping," *arXiv preprint arXiv:2310.18001*, 2023.

[17] Ao Dong, Yuxiang Wang, Pengyang Li, Yifei Tian, Xiaobai Chen, and Jieming Yin, "Dpacc: An fpga-based differential privacy acceleration framework," in *2025 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2025, pp. 1–5.

[18] Ilya Mironov, "Rényi differential privacy," in *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*. IEEE, 2017, pp. 263–275.

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