Balancing Trade-offs: Adaptive Differential Privacy in Interpretable Machine Learning Models

***Abstract*—In the advancing field of machine learning, balancing accuracy, interpretability, and privacy represents a significant challenge. The problem is exacerbated by the widespread deployment of pre-trained models locally in diverse applications, which could lead to various amounts of privacy leakage. Conventional Differential Privacy strategies in which uniform noises are applied to model gradients, guarantee data privacy at the expense of accuracy and interpretability. This paper introduces a Feature-Sensitive Adaptive Differential Privacy (FADP) framework with a novel noise-adding strategy. Noises are adaptively added based on feature importance clustering where important features are considered for interpretability. By employing a unique masking technique, FADP selectively preserves crucial features with minimal noise interference, maintaining accuracy while enhancing interpretability. The FADP framework addresses the limitations of traditional DP methods by preserving critical channels and improving interpretability — a vital requirement in machine learning applications that demand transparency in model decisions. Through comprehensive testing, FADP is shown to balance the trade-offs among accuracy, privacy, and interpretability, marking a substantial advancement in the field of privacy-preserving machine learning.**

***Index Terms*—Privacy-Accuracy-Interpretability Tradeoffs, Differential Privacy, Feature Importance**

I. Introduction

Machine learning (ML) models are increasingly employed across diverse sectors, such as healthcare and finance, where the privacy of sensitive training data is a major concern [1] [2]. These models are prone to memorizing data, exposing them to risks like Membership Inference Attacks (MIA), where adversaries deduce if data points were used in training, leading to potential privacy breaches [3]. As ML deployment expands, the incidence of such attacks escalates, prompting the use of Differential Privacy (DP) techniques. DP, typically implemented by adding Gaussian noise to gradients, helps protect privacy but can diminish model performance due to the uniform noise application [4] [5] [6].

Traditional DP methods, while safeguarding data privacy, often compromise model accuracy. Current research explores various frameworks to lessen this impact, with a growing emphasis on model transparency and interpretability [7]. As ML models become integral to critical applications, ensuring they are both accurate and interpretable becomes essential. However, the typical DP strategy of applying uniform noise affects not just the accuracy but also the interpretability of models, presenting a challenging trade-off between privacy, accuracy, and interpretability [8] [9].

To the best of our knowledge, the proposed underlying technique is a unique approach, a simple yet effective way to selectively add noise to model parameters while preserving critical features that are directly involved in enhancing model interpretability along with model accuracy. By mitigating the impact of high-intensity uniform noise on important features, our method not only preserves interpretability but also demonstrates a positive impact on model accuracy and privacy.

In conventional DP methods, such as Stochastic Gradient Descent with Differential Privacy (DP-SGD) [10], noise is uniformly added to all parameters, impacting those crucial for the model's decision-making. While necessary for privacy, this can compromise the model's accuracy and interpretability, sometimes rendering the data practically irrelevant. Conversely, too little noise risks exposing sensitive data and failing privacy objectives [11]. Our FADP framework thoughtfully adjusts noise application on key features, ensuring optimal noise levels to maintain privacy and facilitate decision-making. This approach strikes a balanced trade-off, enhancing model performance and data privacy simultaneously. We leverage the feature map of the last convolutional layer along with Gradient Maps during training to guide strategic feature clustering for adaptive noise masks. The last layer in Convolutional Neural Networks (CNN) is crucial as it contains abstract, high-level information that the model uses for final decision-making. By preserving important features from excessive uniform noise, our approach enhances interpretability while maintaining strong privacy guarantees. Unlike Grad-CAM (Gradient-weighted Class Activation Mapping) [12], which uses feature maps for interpretability during inference, our FADP approach integrates these maps with gradients during the model training.

Figure 1 demonstrates the main steps of the proposed FADP framework applied to each batch of input samples. Each sample is processed through the convolutional layers of the neural network, where the network extracts features and computes gradients for each parameter during back-propagation, reflecting the contribution of samples to the model's predictions. In the first step (marked as ① in Figure 1), the feature map of the last convolutional layer is integrated with the gradient maps (retrieved before the softmax). The integrated α_1,α_2,....,α_K values represent the important weight scores for each channel that plays a role in the decision-making process of the model. In step ②, these scores are then clustered into three classes based on their importance (high, moderate, and low), which are then used to generate the adaptive noise mask. The adaptive noise masks can have values greater than zero to one. This ensures that no feature map is left unmasked, and the value of this mask helps to control the noise intensity added to the model parameters. In step ③,

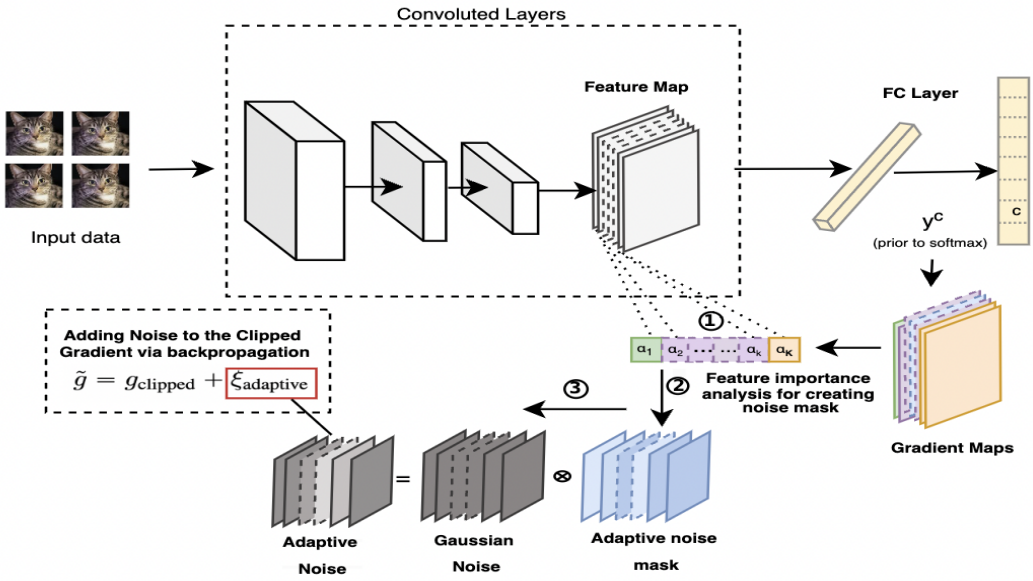


Fig. 1: FADP framework overview

standard Gaussian noise is generated, which is then multiplied with the adaptive noise mask to control the further noise addition to the parameters. The control is done by lowering the intensity of noises in some parts of the generated Gaussian noise, as when "1" is multiplied by any noise value the noise intensity remains the same for lower important features but when any value lower than "1" is multiplied with the noise value it lowers the intensity of the noise for more important features. How the value of the mask is determined is discussed in the following sections. Lastly, the generated Adaptive noise is added to the clipped gradient during backpropagation to update the parameters.

The evaluation of our proposed framework was conducted through multiple experiments to validate its effectiveness in privacy, interpretability, and accuracy. Privacy guarantees were assessed using the standard MIA attack technique, confirming the framework's capacity to protect sensitive training data. For interpretability, GradCAM was applied during inference to visually analyze the model's decision-making process. Model performance was also evaluated using traditional metrics such as accuracy, loss, and confusion matrix, providing a thorough assessment of the framework's accuracy and reliability.

The contributions of the framework can be summarized as below:

- The FADP framework introduces a novel technique by applying noise to model parameters adaptively, based on the importance of features, rather than uniformly across all parameters.
- The study addresses the gap where data privacy techniques traditionally focus on improving model accuracy, but with the growing demand for model transparency, it becomes essential to balance both trade-offs effectively.
- By leveraging feature maps from the last convolutional layer and corresponding gradient maps, the proposed work introduces a feature clustering mechanism to generate the adaptive gradient masking.
- The framework demonstrates a detailed approach to selecting values for adaptive noise masking, effectively controlling noise intensity. This strategy of reducing noise for specific gradients improves model performance without compromising privacy.
- Through extensive evaluation, the framework shows improved privacy preservation while effectively balancing model accuracy and interpretability.

II. Related Work

This section reviews recent research on improving model utility in privacy-preserving machine learning with DP. It discusses how our proposed framework offers a new approach compared to existing work in this area.

A. DP in Machine Learning

DP has emerged as one of the most prominent techniques for ensuring data privacy in machine learning models [13] [14]. DP introduces controlled noise to model computations to obscure the contributions of individual data points, thereby preventing attackers from distinguishing between models trained with or without specific data samples [15]. One of the foundational works on DP, by Dwork *et al.* [16], establishes DP as a mathematical framework for privacy guarantees, which was further expanded in machine learning applications by Abadi *et al.* [17], who introduced the widely used DP-SGD technique which adds noise to the gradients during the training process, offering a balance between privacy and utility. [18] revisited the Gaussian mechanism for DP, focusing on efficiency and perhaps reducing noise. However, traditional DP-SGD often results in reduced accuracy as noise is applied uniformly to

all parameters, regardless of their importance to the model's decision-making.

B. Adaptive DP in Machine Learning

Asi *et al.* proposed the AdaGrad algorithm, which focuses on adaptive methods in the context of convex optimization, whereas FADP applies in a broader context of neural networks and deep learning, dealing with both convex and non-convex problems. [19] proposed a method for adaptively scaling noise based on data sensitivity, minimizing the impact on model accuracy while ensuring robust privacy. [20] introduced a fine-grained control over privacy parameters that adaptively change according to the dataset's properties. [21] developed an approach where noise levels are adjusted dynamically throughout the training process to balance the trade-offs between privacy, accuracy, and convergence. Unlike FADP, these studies demonstrated advancements in adaptive privacy but did not include the direct impact on the interpretability of the model.

C. Trade-off Between Privacy and Accuracy

The privacy-accuracy trade-off remains a critical challenge in DP. Noise-based protection methods like DP introduce a fundamental dilemma: stronger privacy guarantees typically come at the cost of accuracy [22] [23]. To address this, researchers have proposed various strategies to improve accuracy while maintaining privacy. Several works have aimed to address the accuracy degradation caused by DP. For instance [24] [25] [26] [27] [28], [29] propose frameworks that focus on improving model performance. More refined noise mechanisms, such as the Rényi DP proposed by Mironov [30], offer tighter privacy bounds, enabling more flexible trade-offs between privacy and accuracy. [31] proposes MVG, which focuses on providing differential privacy by directional noise for matrix-valued queries. Chen *et al.* [32] discuss a scalable DP approach with gradient compression, which helps maintain higher accuracy by optimizing privacy-utility trade-offs.

Despite these advancements, most of the focus has been on balancing privacy and accuracy, with little attention to how these techniques impact model interpretability. The FADP framework distinguishes itself by not only adapting noise based on the privacy-accuracy trade-off but also emphasizing the preservation and enhancement of model interpretability.

D. Interpretability and DP

Model interpretability, the ability to explain a model's decision-making process, is crucial in domains where understanding predictions is as important as making accurate predictions. However, uniformly adding noise to model gradients in traditional DP often degrades the model's ability to provide interpretable outputs. Techniques like Grad-CAM [33] and SHAP [34] are widely used to understand the decisions of CNNs. Recent work has started exploring the intersection of privacy and interoperability and demonstrated by evaluation how noise is degrading the interpretability. In [35], Naidu *et al.* demonstrated the effects of DP on DNN explainability, especially on medical imaging applications. Ezzeddine *et al.* showed that the enforcement of privacy through DP has a significant impact on detection accuracy and explainability [36].

E. Balancing Privacy, Accuracy, and Interpretability

Balancing privacy, accuracy, and interpretability is an open problem in privacy-preserving machine learning. Many studies have proposed different ways of balancing the trade-offs, however, there are scopes that do not demonstrate the trade-off balance as a whole, and there is room for improvement that FADP aims to achieve by the unique adaptive noise adding technique.

Li *et al.* introduced a framework that balances privacy and interpretability in federated learning settings by using an adaptive noise [37]. However, this approach leaves important features unmasked or unperturbed, raising concerns about the completeness of privacy guarantees (vulnerable to MIA), primarily focusing on better accuracy and interpretability. Patel *et al.* [38] studied the minimum privacy budget required for feature-based model explanations, while Bozorgpanah *et al.* [39] applied SHAP to examine the impact of features on DP-protected models' predictions. However, these approaches focus only on feature-based interpretation rather than the both feature and gradient-based approach that we employ in FADP. Our framework offers a solution by introducing adaptive noise scaling that adjusts noise intensity based on feature importance, thus maintaining both privacy and interoperability. Harder *et al.* [40] proposed methods to improve the interpretability of DP-protected models by optimizing noise allocation in sensitive areas of the model. Similarly, Phan *et al.* [41] developed an adaptive DP framework that applies different levels of noise to features based on their importance, thus preserving both privacy and interoperability. However, these researches do not fully address the trade-off between privacy protection and gradient-based interpretability, particularly in tasks involving image data.

III. Background and Overview

Notations. Lower-case letters like x and i denote variables, with Upper-case letters X and Y representing datasets and labels. Greek letters θ and σ are used for model parameters and noise scale. Bold letters like $\mathbf{x}$ represent vectors. α_c^k is the importance weight for feature map k and class c, while M_k is the adaptive noise mask. Gradients are $\nabla\mathcal{L}(\theta)$, and clipped gradients are $\tilde{\nabla}_k$. Matrix elements $\mathbf{W}_{i,j}$ and transpose $\mathbf{W}^T$ are also used.

A. Baseline Model

Training. The Baseline Model trains a CNN using data without privacy-preserving mechanisms, learning to map inputs to outputs by adjusting internal parameters to minimize prediction errors. This often leads to data memorization, making models susceptible to MIA, where adversaries infer the inclusion of data points in the training set. The training dataset $X = \{x_1, x_2, \ldots, x_n\}$ and labels $Y = \{y_1, y_2, \ldots, y_n\}$ guide the optimization of the model f_θ, which aims to minimize the

loss function $\mathcal{L}(f_\theta(X), Y)$, where $f_\theta(x_i)$ is the prediction for x_i.

Memorization. Models with many parameters tend to memorize training data, particularly when overfitting, as they become finely tuned to specific data points. The optimized parameters θ^* are derived by minimizing $\mathcal{L}$, where $\theta^* = \arg\min_\theta \frac{1}{n}\sum_{i=1}^{n} L(f_\theta(x_i), y_i)$, with $L(f_\theta(x_i), y_i)$ representing the error between predictions and actual labels. Excessive tuning to the training data can lead to overfitting, affecting the model's ability to generalize.

B. Membership Inference Attack

Memorization by deep learning models can be exploited through *Membership Inference Attacks*, where an adversary seeks to determine if specific data points x_i were part of the training dataset. The adversary queries the model $f_\theta(x_i)$ and observes its output, typically the predicted probability or confidence score, $p_\theta(x_i) = \text{softmax}(f_\theta(x_i))$. High confidence scores for training data points reveal their potential inclusion in the training set.

Formally, the adversary determines membership using:

$$\hat{m}_i = \begin{cases} 1 & \text{if } p_\theta(x_i) > \tau \\ 0 & \text{otherwise} \end{cases}$$

where:

- $\hat{m}_i \in \{0, 1\}$ indicates the adversary's guess about the membership of x_i in the training set.
- $p_\theta(x_i)$ is the model's confidence score for x_i.
- τ is a threshold set by the adversary, typically based on the model's behavior on known datasets.

Algorithm 1 Membership Inference Attack (MIA)

1: **Input:** Target model f_θ, shadow model f_θ^{shadow}, train generator $\mathcal{G}_{\text{train}}$, non-train generator $\mathcal{G}_{\text{non-train}}$
2: **Output:** AUC score for the MIA attack
3: Collect data $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{non-train}}$ from $\mathcal{G}_{\text{train}}$ and $\mathcal{G}_{\text{non-train}}$ using f_θ
4: Split $\mathcal{D}_{\text{train}}$ into $\mathcal{D}_{\text{train-shadow}}$ and $\mathcal{D}_{\text{non-train-shadow}}$
5: Train shadow model f_θ^{shadow} with $\mathcal{D}_{\text{train-shadow}}$
6: Generate attack data using predictions from f_θ^{shadow} on $\mathcal{D}_{\text{train-shadow}}$ and $\mathcal{D}_{\text{non-train-shadow}}$
7: Train attack model to distinguish training from non-training data
8: Evaluate the attack on the target model by calculating the AUC score
9: **return** AUC score

Algorithm 1, initializes by collecting and splitting data into training and non-training sets for both the target and shadow models, using these sets to train a shadow model that simulates the target model's behavior. The attack model is then trained to predict membership based on data labeled by the shadow model's predictions, quantifying its effectiveness via the AUC score.

C. Benchmark DP technique

DP is a robust framework for protecting privacy in machine learning models, especially against MIA. In MIAs, adversaries attempt to determine if specific data points were used in training by analyzing model outputs, like confidence scores. DP addresses these concerns by adding randomness to the model's gradient updates during training, thus obscuring individual data contributions. Specifically, DP-SGD, a variant within DP, introduces noise to gradients during Stochastic Gradient Descent, preserving privacy and hindering MIAs by disrupting pattern predictability in prediction scores, generally reducing AUC scores for attack models.

Consider a dataset X of samples with Y labels, where $x_i \in \mathbb{R}^d$ denotes an input data point. The model's parameters $\theta \in \mathbb{R}^p$ are updated by minimizing the loss function $L(\theta, X)$ with a learning rate η, using mini-batches X_t at each iteration t. DP-SGD computes gradients $\nabla\mathcal{L}(\theta_t, X_t)$ for X_t per iteration

Gradients are clipped to a maximum norm C to limit the influence of individual data points:

$$\tilde{\nabla} \leftarrow \nabla \max\left(1, \frac{\|\nabla\|}{C}\right) \tag{1}$$

Post-clipping, Gaussian noise $N(0, \sigma^2)$ is added, updating the gradients as:

$$\theta_{t+1} = \theta_t - \eta\left(\tilde{\nabla} + N(0, \sigma^2)\right) \tag{2}$$

This method adjusts noise based on the privacy budget parameters ϵ, δ, and sensitivity Δ:

$$\sigma = \frac{\Delta}{\epsilon} \cdot \sqrt{2\log\frac{1.25}{\delta}} \tag{3}$$

Typically, higher ϵ values reduce σ, decreasing noise for better accuracy but weaker privacy protections and vice versa. Typical ϵ values are between 1 and 10, with δ around 10^{-5} to ensure strong privacy [8].

This process repeats across a set number of iterations to balance learning efficacy with privacy protection.

D. Grad-CAM

Grad-CAM is a widely utilized technique in explainable AI that improves the interpretability of deep learning models, particularly CNNs. It offers visual explanations for model predictions by producing heatmaps that emphasize the key regions of an input image affecting the model's decision-making. Utilizing the gradients of the output for a specific class against the feature maps of the last convolutional layer, Grad-CAM uncovers the importance of spatial features like edges and textures in class predictions. This tool is especially crucial in sectors such as healthcare, finance, and autonomous driving, where comprehending a model's logic is as important as its prediction accuracy.

Let f_θ represent the CNN model with parameters θ, and $A_k(x)$ denote the activations of the k-th feature map in the last convolutional layer for an input x. Let y_c be the class output score. The gradient of y_c with respect to $A_k(x)$ is given by:

$$\frac{\partial y_c}{\partial A_k(x)}$$

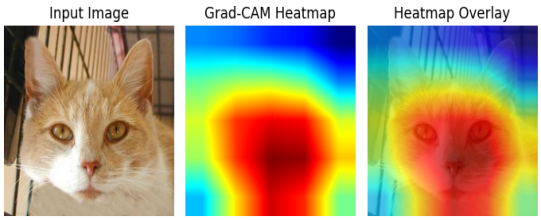


Fig. 2: Three stages of Grad-CAM from input to output, highlighting the areas the model focuses on during prediction.

The importance weights α_{ck} for each feature map are computed as:

$$\alpha_{ck} = \frac{1}{Z} \sum_i \sum_j \frac{\partial y_c}{\partial A_{ij}^k(x)} \tag{4}$$

where Z is the normalization factor, typically the spatial dimensions of the feature map, and $A_{ij}^k(x)$ refers to the activation at position (i, j).

The Grad-CAM heatmap $L^c_{\text{Grad-CAM}}(x)$ for class c is calculated as:

$$L^c_{\text{Grad-CAM}}(x) = \text{ReLU}\left(\sum_k \alpha_{ck} A_k(x)\right) \tag{5}$$

The ReLU function is used to ensure that only positive contributions to the class score are considered, focusing on features that positively influence the model's decision.

Figure 2 demonstrates the Grad-CAM process across three images. The "Input Image" undergoes forward processing to produce activations and predictions. "Grad-CAM heatmap" in figure 2 shows the heatmap, pinpointing influential image regions for the model's decision. This step involves back-propagating gradients to compute importance weights α_c^k for each feature map, which are then used to create the class-specific heatmap. "Heatmap Overlay" in figure 2 overlays the heatmap on the input image, providing a visual explanation of the model's decision-making process and enabling a clear interpretation of which image regions most influence the model's decisions.

E. Feature-Sensitive Adaptive Differential Privacy (FADP)

The proposed FADP framework aims to balance privacy, accuracy, and interpretability by introducing adaptive noise based on the importance of the features. Rather than applying uniform noise to all model parameters, FADP adjusts the noise based on the contribution of different features to the model's decision-making process. The feature importance is computed using the gradient of the model's output for a given class with respect to the feature maps, and the noise is adapted accordingly.

Let X be the dataset with corresponding labels Y. The model's output for input x is denoted as $f_\theta(x)$, where $\theta \in \mathbb{R}^p$ are the learnable parameters, and the loss function $L(\theta, X)$ measures the prediction error. The model's score for class c is represented as y_c , and $A_k(x)$ denotes the activations of the k-th feature map at layer A for input x.

Algorithm 2 FADP: Feature-Sensitive Adaptive Differential Privacy with Clustering

Input: Dataset X, Learning rate η, Noise scale σ, Clipping norm C, Importance weights α_c^k
Output: Model parameters θ
Initialize model parameters θ_0
for each iteration $t = 1$ to T **do**
  Sample mini-batch X_t from dataset X
  Compute gradients $\nabla\mathcal{L}(\theta_t, X_t)$
  Compute feature importance weights α_c^k using:

$$\alpha_c^k = \frac{1}{Z} \sum_i \sum_j \frac{\partial y_c}{\partial A_{ij}^k(x)}$$

  Cluster importance weights into high, moderate, and low groups using thresholds α_{moderate} and α_{high}
  Assign adaptive noise mask M_k to each feature map:

$$M_k = \begin{cases} m_{\text{low}}, & \text{if } \alpha_c^k \in (\alpha_{\min}, \alpha_{\text{moderate}}] \\ m_{\text{moderate}}, & \text{if } \alpha_c^k \in (\alpha_{\text{moderate}}, \alpha_{\text{high}}] \\ m_{\text{high}}, & \text{if } \alpha_c^k \in (\alpha_{\text{high}}, \alpha_{\max}] \end{cases}$$

  Clip gradients:

$$\tilde{\nabla}_k \leftarrow \frac{\nabla_k}{\max(1, \frac{\|\nabla_k\|}{C})}$$

  Add adaptive noise:

$$\tilde{\nabla}_k \leftarrow \tilde{\nabla}_k + M_k \cdot \mathcal{N}(0, \sigma^2)$$

  Update parameters:

$$\theta_{t+1} \leftarrow \theta_t - \eta \tilde{\nabla}_k$$

end for
Return final model parameters θ_T

The importance of each feature map $A_k(x)$ for class c is derived from the gradient of the class score y_c with respect to the activations $A_k(x)$ which can be explained with equation 4, as the proposed framework adopts its first step drawing inspiration from the underlying technique used in Grad-CAM. However, unlike Grad-CAM which operates during the inference phase, our framework innovatively applies this concept during the training phase to enhance privacy-preserving interpretability. In equation 4 the notation Z is the normalization factor (typically the size of the feature map), and $A_{ij}^k(x)$ refers to the activation at spatial position (i, j).

The weights α_{ck} are clustered into three categories: high importance, moderate importance, and low importance. Introducing a moderate importance class offers a nuanced approach to noise application. If the importance weights were divided solely into high and low categories, the resulting adaptive noise could exhibit more pronounced patterns, potentially compromising privacy. By incorporating a third, moderate category, the framework achieves a more hierarchical and gradual control over noise intensity, ensuring a smoother transition between high and low-importance features, thereby reducing the risk of identifiable noise patterns while enhancing the balance between interpretability and privacy [41].

The three clusters of feature maps are determined by

calculating the highest and lowest importance weights and then dividing this range into three segments. Feature maps with the largest importance weights $\alpha_{ck} \in (\alpha_{\text{high}}, \alpha_{\text{max}}]$ are considered highly important. This means that any feature map whose importance weight is strictly greater than α_{high} and up to α_{max} falls into the high importance category. Similarly, maps with intermediate weights $\alpha_{ck} \in (\alpha_{\text{moderate}}, \alpha_{\text{high}}]$ are of moderate importance, and those with the smallest weights $\alpha_{ck} \in (\alpha_{\text{min}}, \alpha_{\text{moderate}}]$ are of low importance.

Once the privacy budget parameters ϵ and δ along with Δ are set, the value of σ becomes fixed, as shown in equation 3. To further control the intensity of the noise to be added, the value of M_k is introduced, which modulates the noise for different feature maps. Specifically, the noise applied is scaled by multiplying σ with M_k, allowing us to adjust the noise based on feature importance, $M_k = M(\alpha_{ck})$.

For high-importance features, $M(\alpha_{ck}) = m_{\text{high}}$, similarly Moderate-importance features receive a scaled noise m_{moderate}, where m_{moderate} lies between the values for high and low importance. Both m_{high} and m_{moderate} have values greater than 0 but less than 1, which helps reduce the noise intensity. Conversely, for low-importance features, $M(\alpha_{ck}) = m_{\text{low}} = 1$, ensuring that the full intensity of the generated Gaussian noise is applied without alteration.

By adjusting m_{high} and m_{moderate}, the framework ensures a balance between privacy and model interpretability, with more critical features receiving less noise and less critical features maintaining stronger privacy guarantees.

Gradient clipping is applied to ensure that no individual feature has too much influence on the model update, which is represented in equation 1.

Finally, the generated adaptive noise mask M_k modulates the noise intensity applied to the gradients:

$$\tilde{\nabla}_k = \nabla_k + M_k \cdot N(0, \sigma^2) \tag{6}$$

This adaptive noise scaling mechanism ensures that more critical features are safeguarded by applying lower noise levels while less important features are assigned stronger noise to enhance privacy protection. Importantly, the noise intensity is regulated such that the value of σ remains within a range that has been rigorously validated in the literature to guarantee differential privacy.

Algorithm 2 details the FADP technique, beginning with the initialization of model parameters θ_0 (line 3). Each iteration starts by sampling a mini-batch X_t from the dataset (line 5), followed by computing gradients $\nabla\mathcal{L}(\theta_t, X_t)$ for the mini-batch (line 6). Important weights α_c^k for each feature map are calculated using the gradient of the class score y_c with respect to the feature map activations $A^k(x)$ (line 7). These weights are categorized into high, moderate, and low importance based on thresholds α_{moderate} and α_{high} (line 8). Adaptive noise masks M_k are assigned based on importance (line 9). Gradients are then clipped by norm C (line 10), and adaptive noise scaled by M_k is added (line 11). The model updates using the noisy, clipped gradients $\tilde{\nabla}_k$ (line 12), repeating for T iterations.

Theorem (Differential Privacy Guarantee of FADP)
The FADP framework ensures differential privacy by adaptively applying non-isotropic Gaussian noise based on the importance of features, thereby preserving privacy while maintaining a significant level of model accuracy and interpretability.

Proof: Consider a model f_θ parameterized by θ, trained on a dataset D. The feature importance weights, denoted by α_i, influence the scale of noise added to each feature's gradient during training.

Let $g_i(\theta)$ represent the gradient of the loss function with respect to feature i of the model parameters θ. The FADP mechanism modifies $g_i(\theta)$ by adding Gaussian noise $N(0, \sigma_i^2)$, where σ_i is adapted based on α_i, the importance weight of the feature:

$$g_i'(\theta) = g_i(\theta) + N(0, \sigma_i^2(\alpha_i))$$

where $\sigma_i(\alpha_i)$ is defined such that critical features (higher α_i) receive less noise to preserve their interpretability and contribution to the model's accuracy.

The adaptive noise $N(0, \sigma_i^2(\alpha_i))$ ensures that each feature's contribution to the output is perturbed to limit the influence of any single training example, adhering to differential privacy. The overall noise variance $\sigma_i^2(\alpha_i)$ is calibrated to ensure that for any two adjacent datasets D and D' differing by a single element, the probability distributions of their outputs are indistinguishable:

$$\mathbb{P}[f_\theta(D) \in S] \leq e^\epsilon \mathbb{P}[f_\theta(D') \in S] + \delta$$

for all $S \subseteq \text{Range}(f_\theta)$, ensuring (ϵ, δ)-differential privacy.

The calibration of $\sigma_i(\alpha_i)$ is crucial and is typically set such that $\sigma_i(\alpha_i) = \frac{\Delta_i}{\epsilon}\sqrt{2\log\frac{1.25}{\delta}}$, where Δ_i is the sensitivity of feature i and depends inversely on α_i. This adaptation ensures that more important features (lower Δ_i) receive proportionally less noise.

IV. Threat Model

In this section, we outline the threat model considered in the design of the *FADP* framework. The objective is to protect the training dataset from adversarial attempts to infer sensitive information, particularly individual data points.

A. Adversary's Knowledge and Capabilities

In our threat model, we assume a *black-box attack* where the adversary cannot access model parameters but can observe outputs (e.g., probability scores or labels).

The adversary has the following capabilities:

- **Access to the model**: The adversary can query the model with data points of their choice and observe the outputs [42].
- **Knowledge of the data distribution**: The adversary may have a general understanding of the data distribution or access to auxiliary data from the same distribution as the model's training data [43].

- **No knowledge of model parameters**: The adversary cannot view or modify the model's internal weights or architecture (typical in a black-box setting). [44]
- **Limited query access**: The adversary can send a limited number of queries to the model, ensuring that their attack is realistic in terms of time and resource constraints [45].

B. Attack Vectors

The attack vector considered here is the MIA, where an adversary follows these steps:

- **Model Querying**: The adversary queries the target model with crafted inputs resembling both training and non-training data, observing outputs like confidence scores or labels [46].
- **Pattern Exploitation**: By analyzing model responses, the adversary identifies higher confidence in training data, exploiting this to infer membership [47].
- **Training a Shadow Model**: A shadow model trained on similar data mimics the target, allowing the adversary to develop a binary attack model distinguishing training from non-training patterns [45].
- **Membership Inference**: The attack model then evaluates new data, assessing the likelihood of training set membership [45], [48].

Through black-box access and observing outputs, MIAs exploit the behavioral distinctions between training and non-training data. DP techniques, such as FADP, aim to mitigate these risks by adding noise to the training process, reducing the adversary's ability to leverage these differences effectively.

C. Adversarial Objective and Failure Probability

The adversary's primary objective is to enhance their attack model's performance by increasing its capacity to differentiate between training set members and non-members. This performance is generally evaluated using the AUC score, which effectively measures the model's ability to distinguish between classes across all classification thresholds [45]. Additionally, the adversary seeks to challenge the effectiveness of differential privacy mechanisms, either by recovering a data point x_i or determining its membership in the training set. A high AUC score suggests that the adversary's model is proficient at identifying training data usage.

The privacy guarantee of FADP ensures that for any two adjacent datasets X and X', the probability of producing a specific output remains comparable:

$$P(f_\theta(X) = o) \approx P(f_\theta(X') = o) \tag{7}$$

where this similarity is regulated by the differential privacy parameters (ϵ, δ). By adding noise, particularly to less crucial features, FADP disrupts patterns in the model's output, reducing the AUC score of the attack model and thereby diminishing the adversary's ability to infer dataset membership. This controlled noise application strategically preserves privacy while minimizing the impact on model accuracy.

D. Security Guarantees

The FADP framework provides a strong defense against the adversarial models described above. Specifically: The use of **gradient clipping** limits the impact of any single data point, thus protecting against *gradient-based attacks*. The **adaptive noise scaling** ensures robustness against *membership inference attacks* by adding more noise to features less relevant to the prediction. By balancing the trade-offs between privacy and accuracy, FADP provides a comprehensive defense mechanism against common attack vectors while maintaining model interpretability and performance.

V. Evaluation

A. Experimental Setup

In this section, we describe the setup for evaluating the FADP framework. The evaluation focuses on the trade-offs between privacy, accuracy, and interpretability, utilizing both visual and numerical metrics.

*1) **Environment and Tools**:* All experiments were conducted using Google Colab's TPU (T4) for training [49]. The models were implemented using TensorFlow and Python 3.x. Standard machine learning libraries such as NumPy and Matplotlib were used for data processing and visualization, while OpenCV and scikit-learn were used for image manipulation and evaluation metrics.

*2) **Model Architecture**:* We utilized the MobileNet architecture with pre-trained ImageNet weights as the backbone of the model [50]. To enhance the feature extraction for Grad-CAM and adaptive noise application, we added an extra convolutional layer followed by fully connected layers. This allows us to leverage the activations from the last convolutional layer to compute feature importance, which plays a critical role in the FADP framework.

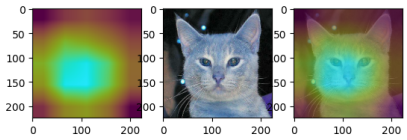
*3) **Datasets**:* The evaluation was carried out on two datasets:

- **CIFAR-10**: A dataset of 60,000 32x32 color images containing 10 classes with a distribution of 50,000 images for training and 10,000 for testing. [51]
- **Cat and Dog**: The dataset is available on Kaggle which consists of two classes: cats and dogs. The dataset includes a total of 25,000 images, evenly distributed between the two categories, with 12,500 images of cats and 12,500 images of dogs. [52]

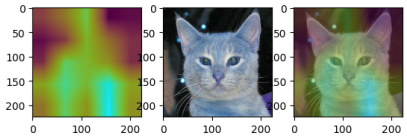
These datasets were chosen for their diversity in the number of classes, allowing us to assess the scalability and performance of the FADP framework both for binary and non-binary classification CNN models. Binary models are more prone to overfitting [53] and membership inference attacks, making them ideal for assessing the robustness and privacy trade-offs in our framework [54]. This allows us to demonstrate the FADP's effectiveness under high-risk scenarios.

*4) **Evaluation Metrics**:* We evaluated the FADP framework using key metrics to assess classification performance, privacy, and interpretability. Classification performance was measured using accuracy percentage, loss, and a confusion matrix on

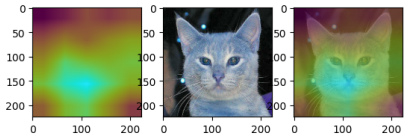
the test set. Privacy was evaluated through the MIA, where the AUC score indicated the attack model's ability to distinguish between training and non-training data. Additionally, a heatmap of data point probabilities was analyzed to assess the model's targeting accuracy. For interpretability, we used Grad-CAM heatmap overlays on input images and quantitatively measured with the Structural Similarity Index (SSIM) [55] to compare heatmaps from different models.



(a) Grad-CAM output for Baseline Model (No DP)



(b) Grad-CAM output for Benchmark DP (DP-SGD)



(c) Grad-CAM output for the proposed FADP

Fig. 3: Grad-CAM heatmaps for different models to demonstrate model interpretability

*5) **Training Setup and Hyperparameters**:* The model was trained using the Adam optimizer with a learning rate of $\eta = 0.001$, a mini-batch size of 64, and a clipping norm $C = 1.0$ to limit the gradient magnitude. In our experimental setup, we ensure that the adaptive noise scaling using m_{high}, m_{moderate}, and m_{low} maintains Gaussian noise within a range that upholds privacy guarantees. According to existing literature, ϵ values between 0.1 and 3 with a small δ (e.g., 10^{-5}) provide adequate privacy protection [8]. When $\epsilon = 0.1$, the maximum noise level σ_{base} is calculated using equation 3. Conversely, for $\epsilon = 3$, the minimum noise level σ_{min} is determined, ensuring the noise falls within $\sigma_{\text{min}} \leq \sigma \leq \sigma_{\text{base}}$. To balance privacy and utility, we apply scaling factors $m_{\text{high}} \approx 0.6$ and $m_{\text{moderate}} \approx 0.8$. For high-importance features, the noise is scaled with $M_k = m_{\text{high}} \cdot \sigma_{\text{min}}$, ensuring privacy remains intact. For moderate-importance features, $M_k = m_{\text{moderate}} \cdot \sigma_{\text{base}}$ provides a balanced reduction. Low-importance features receive the full noise intensity with $m_{\text{low}} = 1$. This ensures that noise levels correspond to the privacy guarantees associated with ϵ values within [0.1, 3].

*6) **Model Comparisons**:* For comparison, we evaluated the following models in each result section:

- **Baseline Model (No DP)**: A model trained without any privacy-preserving mechanisms.

Phase	Baseline		DP-SGD		FADP	
	Acc.	Loss	Acc.	Loss	Acc.	Loss
Train	0.9215	0.2310	0.8125	0.7063	0.882	0.3013
Val.	0.9110	0.2635	0.8200	0.4124	0.8750	0.2896
Test	0.8701	0.4981	0.8250	0.3584	0.8500	0.4286

TABLE I: Comparison of Baseline, DP-SGD, and FADP Models on Cat and Dog Dataset

Phase	Baseline		DP-SGD		FADP	
	Acc.	Loss	Acc.	Loss	Acc.	Loss
Train	0.9310	0.2100	0.8100	0.6520	0.8600	0.2031
Val.	0.9200	0.2320	0.8250	0.3990	0.8650	0.2790
Test	0.8850	0.2590	0.8350	0.2910	0.8400	0.2390

TABLE II: Comparison of Baseline, DP-SGD, and FADP Models on CIFAR-10 Dataset

- **DP-SGD (Benchmark DP)**: A model trained using Stochastic Gradient Descent Differential privacy with uniform noise addition after gradient clipping.
- **FADP**: A model trained using the proposed FADP technique with adaptive noise.

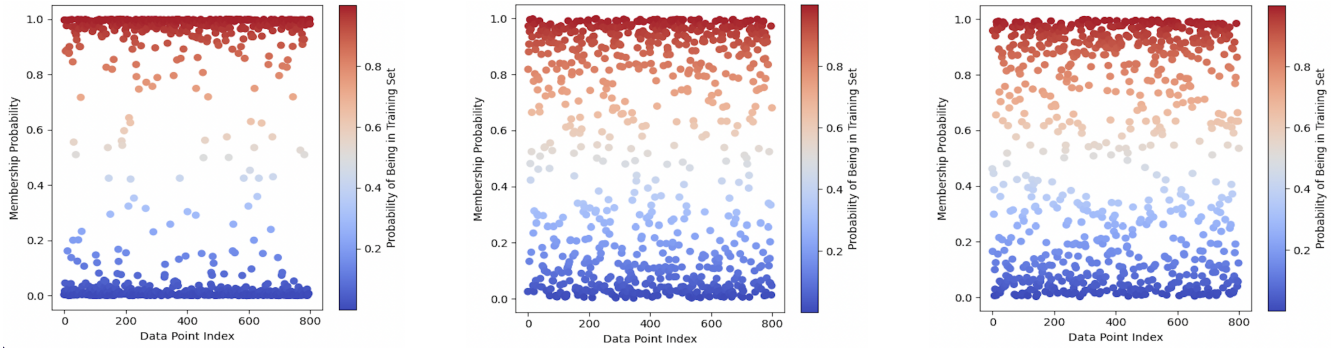
Model	Input	Baseline	Compared	SSIM
Benchmark DP-SGD				0.31
				0.45
Proposed FADP				0.86
				0.81

TABLE III: SSIM score comparison with baseline model

B. Results

*1) **Accuracy Improvement Analysis**:* The FADP framework significantly enhances accuracy over the DP-SGD method, as evidenced in Table I using the Cat and Dog Dataset. The Baseline Model achieved training, validation, and testing accuracies of 92.15%, 91.10%, and 87.01% without privacy constraints. In comparison, DP-SGD saw a decline in training accuracy to 81.25%, illustrating the privacy-performance trade-offs.

FADP improved training accuracy to 88.2%, validation to 87.50%, and testing accuracy to 85.00%, clearly surpassing DP-SGD and, in testing, nearly matching the Baseline. This underscores FADP's ability to balance accuracy with privacy in a binary CNN classifier setting. For CIFAR-10, as shown in Table II, the Baseline Model's training, validation, and testing accuracies were 93.10%, 92.00%, and 88.50% respectively, without privacy interventions. DP-SGD reduced the training accuracy significantly to 81.00%. In contrast, FADP not only



(a) Baseline Model (No DP) AUC: 0.61 (b) Benchmark DP-SGD model AUC: 0.45 (c) FADP model AUC: 0.48

Fig. 4: Comparison of AUC Scores of the MIA Attack models on the Baseline, Benchmark DP (DP-SGD), and FADP Models

outperformed DP-SGD with 86.00% training accuracy but also recorded 86.50% validation and 84.00% testing accuracy, demonstrating robust performance in a non-binary CNN context.

*2) **Interpretability Improvement Analysis**:* Figure 3 compares the interpretability of the Baseline Model (No DP), the Benchmark DP (DP-SGD), and the proposed FADP framework, using Grad-CAM heatmaps as a visual tool to highlight model focus areas during prediction.

In Figure 3a, the Baseline Model's Grad-CAM heatmap focuses sharply on the facial features of the cat, such as the eyes and nose. This precision suggests high interpretability as the model bases predictions on key, intuitive features crucial for classification. In Figure 3b, the Grad-CAM output for the DP-SGD model shows a diffused heatmap, indicating that the uniform application of noise has broadened the model's focus, including irrelevant areas. This broadened attention reflects a reduction in interpretability due to privacy-focused noise addition, diluting the model's ability to focus on essential features. Conversely, Figure 3c illustrates that the FADP framework's Grad-CAM output remains focused on important image regions like the eyes and nose. Despite noise application for privacy, the FADP's adaptive noise approach helps preserve interpretability. The heatmap is broader than the baseline but still highlights critical features, indicating minimal impact on key decision-making areas.

This comparison shows that the FADP model successfully balances privacy and interpretability by adaptively controlling noise intensity based on feature importance, maintaining focus on essential features. Unlike the DP-SGD model, which scatters attention and degrades interpretability, the FADP framework ensures critical regions are emphasized, demonstrating its effectiveness in preserving interpretability alongside privacy. Table III provides SSIM scores to quantify the similarity of heatmaps to the baseline. The DP-SGD model scores of 0.31 and 0.45 reflect a notable decline in heatmap similarity due to uniform noise diffusing the model's focus. In contrast, the FADP model's scores of 0.86 and 0.81 suggest a high similarity, underscoring the FADP's superior ability to maintain critical focus while implementing privacy-preserving measures.

*3) **Under Attack Performance Analysis**:* Figure 4 illustrates the results of the MIA on the baseline, DP-SGD, and FADP models, visualized as heatmaps indicating the membership probability for different data points of the Cat and Dog Dataset.

"In Figure 4a, the heatmap shows a clear separation between training and non-training data, with training set data points having a membership probability close to 1, and non-training data closer to 0. This clear distinction highlights the model's vulnerability to MIA, as evidenced by a higher AUC score of 0.59. The gradient pattern underscores this difference, indicating a successful attack. Conversely, in Figure 4b, the gradient is more diffused, and membership probabilities are less distinct, showing overlapping between members and non-members. This suggests that DP-SGD effectively reduces the model's susceptibility to MIA, evidenced by a lower AUC score of 0.45. The added noise blurs the distinction between training and non-training data, enhancing privacy but at some cost to model performance. The color gradient shows more blending between probabilities, indicating a stronger defense. Figure 4c presents the results for the FADP framework, where membership probabilities are more distributed than in the baseline but more structured than in the DP-SGD model. The heatmap indicates a better balance between privacy and accuracy, challenging the adversary more than the baseline. With an AUC score of 0.48, FADP shows improvement over the baseline, reducing model vulnerability to MIA while maintaining a more focused distribution of membership probabilities."

In summary, Figure 4 demonstrates the baseline model's vulnerability to MIA, with a distinct separation of members and non-members resulting in a high AUC score. DP-SGD diffuses this distinction more effectively, significantly lowering the AUC but reducing focus on key features. FADP achieves a balanced approach, reducing attack success while maintaining interpretability and an intermediate level of privacy and accuracy.

*4) **Selection of Noise Mask Values**:* In Table IV, the FADP model's performance is evaluated under varying m values, considering both accuracy and the AUC score of Membership Inference Attacks (MIA). A lower AUC reflects better

Case	Model Parameters	Accuracy		MIA Attack AUC	
		Cat vs Dog	CIFAR-10	Cat vs Dog	CIFAR-10
Case 1	$m_{\text{high}} = 0.5, m_{\text{moderate}} = 0.7, m_{\text{low}} = 1$	0.8950	0.8750	0.51	0.54
Case 2	$m_{\text{high}} = 0.55, m_{\text{moderate}} = 0.75, m_{\text{low}} = 1$	0.8900	0.8600	0.49	0.54
Case 3	$m_{\text{high}} = 0.6, m_{\text{moderate}} = 0.8, m_{\text{low}} = 1$	0.8800	0.8400	0.45	0.48
Case 4	$m_{\text{high}} = 0.65, m_{\text{moderate}} = 0.85, m_{\text{low}} = 1$	0.8650	0.8150	0.44	0.47

TABLE IV: FADP Model Accuracy and MIA Attack AUC Scores for Different m Values Across Datasets

Model	Training Time (minutes)	Number of Steps	Remarks on Computational Cost
Baseline Model (No-DP)	≈ 45.1	Standard steps	Least cost, no privacy steps
DP-SGD Model	≈ 53.7	Extra privacy steps	Higher cost for DP mechanisms
FADP Model	≈ 58.3	Feature clustering + Adaptive mask generating	Slightly higher for mask generation

TABLE V: Comparison of Computational Cost for Baseline, DP-SGD, and FADP Models

privacy, while higher accuracy indicates better model performance. Among the cases, Case 3 ($m_{\text{high}} = 0.6, m_{\text{moderate}} = 0.8, m_{\text{low}} = 1$) provides a well-balanced trade-off between accuracy and privacy. It achieves AUC scores of 0.45 for Cat vs. Dog and 0.48 for CIFAR-10, indicating stronger privacy protection compared to the other cases. Although its accuracy is slightly lower than in Case 1, the reduction in AUC makes it the most balanced configuration.

*5) **Computational Cost**:* Table V presents the computational cost comparison for the Baseline Model (No DP), Benchmark DP (DP-SGD), and FADP models, all trained on the Cat and Dog dataset using Google Colab GPU T4 for 50 epochs. The training times reported in the table represent the approximate average from multiple independent training runs for each model across the dataset. The table indicates that the Baseline Model has the lowest training time, as no privacy mechanisms are involved. The DP-SGD model requires more time due to the added differential privacy steps. The FADP model incurs a slightly higher cost than DP-SGD due to the additional steps needed to cluster the features and generate the adaptive noise mask. Despite the increase in training time which is a limitation for the proposed framework, the primary objective of enhancing privacy while maintaining performance was successfully achieved.

VI. Conclusion

The proposed framework successfully addresses the critical trade-offs between privacy, accuracy, and interpretability in machine learning models. By applying noise adaptively based on feature importance, rather than uniformly, the FADP model preserves key features essential for decision-making while maintaining adequate privacy. Extensive testing on various datasets demonstrates that FADP achieves a more balanced trade-off, improving interpretability and maintaining high model accuracy while offering strong privacy guarantees. Although the framework incurs slightly higher computational costs, particularly in comparison to standard DP methods, the significant improvements in model performance and privacy preservation underscore the effectiveness of the FADP approach.

References

[1] P. Himthani, G. P. Dubey, B. M. Sharma, and A. Taneja, "Big data privacy and challenges for machine learning," pp. 707–713, 2020.

[2] X. Wu, R. Duan, and J. Ni, "Unveiling security, privacy, and ethical concerns of chatgpt," *Journal of Information and Intelligence*, vol. 2, no. 2, pp. 102–115, 2024.

[3] H. Hu, Z. Salcic, L. Sun, G. Dobbie, P. S. Yu, and X. Zhang, "Membership inference attacks on machine learning: A survey," *ACM Computing Surveys (CSUR)*, vol. 54, no. 11s, sep 2022. [Online]. Available: https://doi.org/10.1145/3523273

[4] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 308–318.

[5] S. L. Garfinkel, J. M. Abowd, and S. Powazek, "Issues encountered deploying differential privacy," in *Proceedings of the 2018 Workshop on Privacy in the Electronic Society*, ser. WPES'18. New York, NY, USA: Association for Computing Machinery, 2018, p. 133–137.

[6] A. Machanavajjhala, X. He, and M. Hay, "Differential privacy in the wild: A tutorial on current practices & open challenges," in *Proceedings of the 2017 ACM International Conference on Management of Data*, ser. SIGMOD '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 1727–1730.

[7] P. Thunki, S. R. B. Reddy, M. Raparthi, S. Maruthi, S. Babu Dodda, and P. Ravichandran, "Explainable ai in data science - enhancing model interpretability and transparency," *African Journal of Artificial Intelligence and Sustainable Development*, vol. 1, no. 1, p. 1–8, Apr. 2021.

[8] J. Lee and C. Clifton, "How much is enough? choosing ϵ for differential privacy," in *Information Security*, X. Lai, J. Zhou, and H. Li, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 325–340.

[9] R. Naidu, A. Priyanshu, A. Kumar, S. Kotti, H. Wang, and F. Mireshghallah, "When differential privacy meets interpretability: A case study," *arXiv preprint arXiv:2106.13203*, 2021.

[10] J. M. Altschuler and K. Talwar, "Privacy of noisy stochastic gradient descent: more iterations without more privacy loss," in *Proceedings of the 36th International Conference on Neural Information Processing Systems*, ser. NIPS '22. Red Hook, NY, USA: Curran Associates Inc., 2024.

[11] C. Dwork, A. Roth *et al.*, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[12] P. Das and A. Ortega, "Gradient-weighted class activation mapping for spatio temporal graph convolutional network," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 4043–4047.

[13] A. Bittau, Ú. Erlingsson, P. Maniatis, I. Mironov, A. Raghunathan, D. Lie, U. Rudominer, A. Kode, S. Tinnes, and B. Seefeld, "Differential privacy with shuffled data," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 123–135.

[14] A. Blanco-Justicia, D. Sánchez, J. Domingo-Ferrer, and K. Muralidhar, "A critical review on the use (and misuse) of differential privacy in machine learning," *ACM Computing Surveys*, vol. 55, no. 8, pp. 1–16, 2023.

[15] J. Zhao, Y. Chen, and W. Zhang, "Differential privacy preservation in deep learning: Challenges, opportunities and solutions," *IEEE Access*, vol. 11, pp. 1–15, 2023.

[16] C. Dwork, "Differential privacy: A survey of results," in *International conference on theory and applications of models of computation*. Springer, 2008, pp. 1–19.

[17] M. Abadi *et al.*, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16)*, 2016.

[18] T. Ji and P. Li, "Less is more: Revisiting the gaussian mechanism for differential privacy," in *Proceedings of the 33rd USENIX Security Symposium*, 2024, pp. 937–954.

[19] Z. Jorgensen and T. Yu, "Conservative or liberal? personalized differential privacy," *2015 IEEE 31st International Conference on Data Engineering*, pp. 1023–1034, 2015.

[20] F. McSherry, "Privacy integrated queries: an extensible platform for privacy-preserving data analysis," in *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*. ACM, 2009, pp. 19–30.

[21] A. G. Thakurta and A. Smith, "Differentially private empirical risk minimization," in *Journal of Machine Learning Research*, vol. 14, 2013, pp. 1067–1109.

[22] K. Chaudhuri *et al.*, "Differentially private empirical risk minimization," *Journal of Machine Learning Research*, vol. 12, pp. 1069–1109, 2011.

[23] N. D. Pham and N. Chilamkurti, "Towards improved privacy-utility trade-off in differential privacy," *Journal of Information Security and Privacy*, vol. 7, pp. 49–60, 2021.

[24] N. D. Pham, K. T. Phan, and N. Chilamkurti, "Enhancing accuracy-privacy trade-off in differentially private split learning," *arXiv preprint arXiv:2104.14618*, 2021, school of Computing, Engineering, and Mathematical Sciences (SCEMS), La Trobe University, Victoria, Australia.

[25] E. Bagdasaryan, O. Poursaeed, and V. Shmatikov, "Differential privacy has disparate impact on model accuracy," in *Advances in Neural Information Processing Systems*, vol. 32. Curran Associates, Inc., 2019.

[26] Y. Zheng, W. Zhang, Y. Zhang, W. Song, K. Zhou, and B. Han, "Rethinking improved privacy-utility trade-off with pre-existing knowledge for dp training," *arXiv preprint arXiv:2409.03344*, 2024.

[27] D. L. Oberski and F. Kreuter, "Differential Privacy and Social Science: An Urgent Puzzle," *Harvard Data Science Review*, vol. 2, no. 1, jan 31 2020, https://hdsr.mitpress.mit.edu/pub/g9o4z8au.

[28] H. Yan, M. Yin, C. Yan, and W. Liang, "A survey of privacy preserving methods based on differential privacy for medical data," 04 2024, pp. 104–108.

[29] N. Papernot *et al.*, "Scalable private learning with pate," in *Proceedings of the 6th International Conference on Learning Representations (ICLR '18)*, 2018.

[30] I. Mironov, "Rényi differential privacy," in *Proceedings of the 2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, 2017.

[31] T. Chanyaswad, A. Dytso, H. V. Poor, and P. Mittal, "Mvg mechanism: Differential privacy under matrix-valued query," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2896–2910, 2018.

[32] M. Chen, Z. Hong, S. Xu *et al.*, "Datalens: Scalable privacy preserving training via gradient compression and aggregation," *arXiv preprint arXiv:2103.11109*, 2021.

[33] R. R. Selvaraju *et al.*, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[34] S. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

[35] R. Naidu, A. Priyanshu, A. Kumar, S. Kotti, H. Wang, and F. Mireshghallah, "When differential privacy meets interpretability: A case study," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security Workshops*, 2021, pp. 1–6.

[36] F. Ezzeddine, M. Saad, O. Ayoub, D. Andreoletti, M. Gjoreski, I. Sbeity, M. Langheinrich, and S. Giordano, "Differential privacy for anomaly detection: Analyzing the trade-off between privacy and explainability," in *Explainable Artificial Intelligence*. Springer, 2024, pp. 294–318.

[37] Z. N. Y. G. W. L. H. S. Zhe Li, Honglong Chen, "Towards adaptive privacy protection for interpretable federated learning," *IEEE Transactions on Mobile Computing*, 2021.

[38] A. Patel *et al.*, "Minimizing privacy budget for interpretability of machine learning models," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020.

[39] S. Bozorgpanah *et al.*, "Shap-enhanced model interpretability in differentially private learning," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1234–1247, 2020.

[40] P. Harder *et al.*, "Interpretable privacy-preserving models: Reconciling utility, privacy, and interpretability," in *Proceedings of the 2020 AAAI Conference on Artificial Intelligence*, 2020.

[41] N. H. Phan *et al.*, "Adaptive laplace mechanism: Differential privacy preservation in deep learning," in *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, 2017.

[42] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction apis," in *25th USENIX Security Symposium (USENIX Security 16)*, 2016, pp. 601–618.

[43] N. Carlini, C. Liu, J. Kos, Ú. Erlingsson, and D. Song, "The secret sharer: Evaluating and testing unintended memorization in neural networks," *28th USENIX Security Symposium (USENIX Security 19)*, pp. 267–284, 2019.

[44] Y. Lu, M. Magdon-Ismail, Y. Wei, and V. Zikas, "Eureka: a general framework for black-box differential privacy estimators," in *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2024, pp. 913–931.

[45] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 3–18.

[46] N. Carlini *et al.*, "Attack on privacy and model extraction using model queries," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (CCS '20)*, 2020, pp. 2029–2042.

[47] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Stand-alone and federated learning under passive and active white-box inference attacks," in *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019, pp. 739–753.

[48] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes, "Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models," in *Proceedings of the 2018 Network and Distributed Systems Security (NDSS)*, 2018.

[49] Google, "Google colaboratory," https://colab.google, 2020, accessed: 2024-08-01.

[50] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 248–255.

[51] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[52] P. Tong, "Cat and dog dataset," 2021, accessed: 2024-10-18. [Online]. Available: https://www.kaggle.com/datasets/tongpython/cat-and-dog/data

[53] S. Yeom, M. Singhal, S. Chien, and D. Wagner, "Privacy risk in machine learning: Analyzing the connection to overfitting," in *Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS)*, 2018, pp. 9746–9755.

[54] L. Melis, C. Song, E. D. Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP)*, 2019, pp. 691–706.

[55] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.