Privacy-Enhancing Sub-Sampling Meets Model Inversion Attacks

Mohammed Abbas and Tomas Matty

***Abstract*—This work examines how optimizer selection and sub-sampling strategies affect model performance and adversarial vulnerability under differential privacy. We evaluate Differentially Private Stochastic Gradient Descent (DP-SGD) with and without sub-sampling, measuring attack success through classification rates, pixel similarity (PSNR), and structural similarity (SSIM). Results show that switching from Adam to DP-SGD, combined with sub-sampling, lowers adversarial classification accuracy across privacy budgets but can also increase pixel-level similarity in reconstructed inputs. These findings highlight a tension between differential privacy and perceptual privacy under standard DP training. Specifically, we simulate a black-box model inversion attack, where the adversary can only query the model's outputs without access to internal parameters. This threat model reflects realistic deployment scenarios and underscores the importance of defenses that do not rely on obfuscation but provide formal privacy guarantees.**

***Sammanfattning*—Detta arbete undersöker hur valet av optimeringsalgoritm och användningen av delmängdsurval påverkar modellens prestanda och sårbarhet för attacker under differentiell sekretess. Vi utvärderar stokastisk gradientnedstigning med differentiell sekretess (DP-SGD) med och utan delmängdsurval genom att mäta attackframgång baserat på klassificeringsgrad, pixelsimilaritet (PSNR) och strukturell similaritet (SSIM). Resultaten visar att övergången från Adam till DP-SGD, i kombination med delmängdsurval, sänker den adversariella klassificeringsnoggrannheten över olika integritetsbudgetar, men samtidigt kan öka pixelnivåns likhet i rekonstruerade indata. Dessa resultat belyser en konflikt mellan medlemskapssekretess och perceptuell sekretess vid standardiserad med differentiell sekretess. Specifikt simulerar vi en black-box model inversionsattack, där angriparen enbart kan göra förfrågningar till modellens utdata utan tillgång till interna parametrar. Denna hotmodell speglar realistiska driftsmiljöer och understryker vikten av försvar som inte förlitar sig på fördoldhet utan erbjuder formella sekretessgarantier.**

***Index Terms*—(Differentially Private Stochastic Gradient Descent, Sub-Sampling, CIFAR-10 Dataset, Rényi Differential Privacy, Convolutional Neural Networks, Model Inversion Attack, Black-Box Attack, Peak Signal-to-Noise Ratio, Structural Similarity Index, Image Classification)**

Supervisors: Leonhard Grosse

TRITA Number: TRITA-EECS-EX-2025:154

I. Introduction

The increasing integration of machine learning (ML) in privacy-sensitive applications, such as healthcare, finance, and personal analytics, raises significant concerns about data confidentiality. Even when models are deployed in seemingly secure environments, they remain vulnerable to privacy attacks. One prominent class of attacks, model inversion attacks, can extract sensitive information about individuals from the outputs of ML models.

Previous research has demonstrated that collaborative inference models deployed across edge–cloud infrastructures are particularly susceptible to these threats [1]. Adversaries can exploit intermediate representations to reconstruct raw input data, even without direct access to the models' internals. Building upon these insights, this work investigates how privacy enhancing sub-sampling techniques combined with differential privacy (DP) can mitigate the risks posed by such attacks.

Specifically, this project explores whether introducing randomness through selective data sampling during training can obscure individual data contributions, thereby reducing the effectiveness of model inversion attacks without severely degrading model performance. We reimplement an existing model inversion attack in a black-box setting to reflect real-world adversarial conditions. We then integrate Differentially Private Stochastic Gradient Descent (DP-SGD) into the training process, leveraging formal privacy guarantees. Throughout, we evaluate sub-sampling, tune key parameters, and assess the trade-offs between privacy, model utility, and computational efficiency.

II. Theory

A. Notation

Let $\mathcal{X}$ denote the domain of individual data records. A dataset $D \in \mathcal{X}^n$ consists of n such records. We write $D \sim D'$ to denote that D and D' are adjacent datasets differing in exactly one entry. A randomized mechanism is a mapping $\mathcal{M} : \mathcal{X}^n \to \mathcal{R}$, where $\mathcal{R}$ is the output range of the mechanism. For example, $\mathcal{M}$ could represent a classifier trained on a dataset, or a statistical query function. We use ε and δ to denote the standard privacy parameters in differential privacy, and α for the order in Rényi differential privacy.

B. Image classifier

The classification models in this project are image classifiers, which aim to assign input images to one of several predefined categories [2]. Image classifiers are commonly built using convolutional neural networks (CNNs), which are well-suited for extracting spatial hierarchies from pixel data. The model architecture we use, CIFAR10CNN, is a custom CNN tailored for the CIFAR-10 dataset. CIFAR-10 consists of 60,000 32×32 color images across 10 classes (e.g., airplane, automobile, bird, cat, etc.) [3].

C. Differential Privacy

Differential privacy (DP) comes from giving individuals in a dataset plausible deniability [4]. This ensures that the output of an algorithm remains statistically similar whether or not a single individual's data is included in the training set, making it difficult for attackers to determine whether specific data was used in training. A classic example illustrating the intuition behind differential privacy is the randomized response mechanism, originally proposed by Warner [5]. Suppose individuals are asked whether they have engaged in an illegal activity. Each participant flips a coin: if it comes up tails, they answer truthfully; if it comes up heads, they flip a second coin and answer "Yes" if it is heads and "No" if it is tails. This randomized process provides plausible deniability, as any individual's response is correlated but not identical with the true answer.

Differential privacy was later formalized by Dwork et al. [6] to rigorously capture such privacy guarantees. Let $\mathcal{X}$ denote the domain of individual data records. A randomized algorithm $\mathcal{M}$ with domain $\mathbb{N}^{|\mathcal{X}|}$ is (ε, δ)-differentially private if, for all $S \subseteq \mathrm{Range}(\mathcal{M})$ and for all $x, y \in \mathbb{N}^{|\mathcal{X}|}$ such that two datasets $x, x' \in \mathcal{X}^n$ are adjacent, denoted $x \sim x'$, if they differ in exactly one entry, it holds that

$$\Pr[\mathcal{M}(x) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(y) \in S] + \delta.$$

Where the probability space is over the randomness of the mechanism $\mathcal{M}$. The parameter ε is the privacy loss bound which controls how much the output of a mechanism can change when a single individual's data is added or removed. δ is a small probability of failure which has a small chance to break the privacy guarantee.

For example, if the probability that M outputs 'Disease detected' is 0.6 when a person is included in the dataset, then after removing that person, the probability might change to at most $0.6 \times e^{\varepsilon}$ (plus a small δ if using approximate DP). Thus, an attacker cannot confidently tell whether any individual's data was used, providing plausible deniability.

D. Stochastic Gradient Descent

Stochastic Gradient Descent (SGD) is a optimization algorithm for training machine learning models [7]. Its goal is to minimize a loss function $\mathcal{L}$ by iteratively updating the model's parameters in the direction that reduces prediction error. Unlike traditional Gradient Descent (GD), which computes the gradient $\nabla \mathcal{L}_i$ over the entire dataset at each step, SGD approximates this by using only a small batch or even a single sample at a time. This introduces stochasticity into the training process.

While the updates in SGD are inherently noisier than in GD, this variability can help the algorithm escape local minima and navigate more complex loss landscapes [7]. Moreover, using subsets of data per iteration reduces memory usage and significantly speeds up computation, making SGD especially suitable for large-scale datasets. A common and practical variant is mini-batch SGD, where updates are made using small, randomly selected batches of size B.

When applying differential privacy to the training process, as in DP-SGD, further modifications are introduced to protect sensitive data [7]. After computing gradients for each mini-batch, gradient clipping is applied. This operation limits the influence of individual samples by scaling each per-sample gradient to a maximum ℓ_2 norm C. If the gradient norm exceeds C, it is rescaled; otherwise, it remains unchanged:

$$\mathrm{clip}(\nabla \mathcal{L}_i, C) = \nabla \mathcal{L}_i \cdot \min\left(1, \frac{C}{\|\nabla \mathcal{L}_i\|_2}\right)$$

The clipped gradients are then averaged, and Gaussian noise is added to the sum to obscure the contribution of any single data point. The model parameters are finally updated using this privatized gradient estimate:

$$\theta_{t+1} = \theta_t - \eta \cdot \left(\frac{1}{|B|}\sum_{i \in B} \mathrm{clip}(\nabla_\theta \mathcal{L}_i, C) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I})\right)$$

where the terms are defined as follows:

- θ_t: model parameters at iteration t
- η: learning rate
- $\mathcal{N}(0, \sigma^2 C^2 \mathbf{I})$: Gaussian noise added for differential privacy, with standard deviation σC and identity covariance matrix $\mathbf{I}$

This modification allows us to train models with formal differential privacy guarantees while retaining much of the utility of standard SGD.

E. Sub-Sampling

Sub-sampling is a fundamental technique in the design of differentially private algorithms, particularly in machine learning. The core idea is rooted in the principle of privacy amplification by subsampling, which states that the application of a private mechanism to a random subset of a dataset yields stronger privacy guarantees than applying the same mechanism to the full dataset [8]. This effect arises because any individual not included in the sample is inherently protected from any data leakage, while those who are included benefit from the randomized nature of selection.

F. Black Box Model Inversion Attack

A black-box attack refers to a threat model where the attacker has no access to the internal structure or parameters of the machine learning model [1]. Instead, the attacker interacts with the model purely by making queries and observing outputs. Despite these limitations, black-box attacks have proven surprisingly effective in compromising privacy. In the context of this project, we assume a black-box setting to mimic real-world deployment scenarios, such as machine-learning-as-a-service platforms. This means the attacker can request predictions from the model but cannot inspect the underlying architecture, weights, or training data. This setup increases the practical relevance of our results while presenting a realistic challenge for privacy protection.

G. Confusion Matrix and Evaluation

Confusion matrices are a visual tool to assess mode performance and accuracy. A confusion matrix summarizes how often each true label is correctly or incorrectly predicted by the model. It provides a more granular view than accuracy alone and is particularly useful for identifying systematic misclassifications between classes.

The confusion matrix C summarizes the performance of a classification model by recording how often predicted classes match true classes.

Mathematically, each entry C_{ij} represents the probability that the model predicts class i when the true class is j:

$$C_{ij} = \Pr[Y = i \mid X = j],$$

where X denotes the true class label and Y denotes the predicted class label.

H. Rényi Differential Privacy

To accurately track the privacy loss during multiple training steps, we use Rényi Differential Privacy (RDP), introduced by Mironov [9].

Let $\mathcal{X}$ be the domain of individual data records, and let $\mathcal{X}^n$ denote the space of datasets consisting of n entries from $\mathcal{X}$. Two datasets $D, D' \in \mathcal{X}^n$ are adjacent, denoted $D \sim D'$, if they differ in exactly one entry. Let $\mathcal{M} : \mathcal{X}^n \to \mathcal{R}$ be a randomized mechanism.

A randomized mechanism $\mathcal{M}$ satisfies (α, ε)-Rényi differential privacy if for all adjacent datasets $D, D' \in \mathcal{X}^n$, the Rényi divergence of order $\alpha > 1$ between $\mathcal{M}(D)$ and $\mathcal{M}(D')$ is at most ε, i.e.,

$$D_\alpha(M(D) \| M(D')) \leq \varepsilon,$$

where D_α denotes the Rényi divergence of order α [10].

RDP provides a tighter and more flexible way to track privacy loss under composition compared to the traditional (ε, δ)-differential privacy framework. Importantly, privacy guarantees in RDP can be seamlessly converted back into (ε, δ)-differential privacy at the end of training, enabling a principled evaluation of the final privacy budget [9]. RDP accounting is used to measure cumulative privacy loss across the multiple steps of DP-SGD.

I. Peak Signal-to-Noise Ratio

The Peak Signal-to-Noise Ratio (PSNR) is a widely used metric for evaluating the quality of reconstructed images, particularly in tasks involving image compression or restoration [**?**]. It quantifies how closely a reconstructed image matches the original, undistorted version by comparing the maximum possible signal power to the power of the noise that affects image fidelity.

PSNR is derived from the mean squared error (MSE) between the original image I and its reconstruction $\hat{I}$. A lower MSE implies fewer differences between the two images. PSNR is then calculated as:

$$\text{PSNR} = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{\text{MSE}}\right),$$

where MAX_I represents the maximum possible pixel value of the image, and MSE is given by:

$$\text{MSE} = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\left(I(i,j) - \hat{I}(i,j)\right)^2,$$

where m and n denote the dimensions of the image.

A higher PSNR value indicates that the reconstructed image is closer to the original and has better visual quality. PSNR serves as a convenient and interpretable measure to assess degradation, especially when human visual inspection is impractical or subjective.

J. Structural Similarity Index

The Structural Similarity Index (SSIM) is a perceptual metric for measuring the similarity between two images [11]. It considers changes in structural information, luminance, and contrast, which align more closely with human visual perception.

SSIM evaluates three key components between a reference image I and a distorted image $\hat{I}$:

- **Luminance comparison** to measure brightness similarity,
- **Contrast comparison** to measure differences in variance,
- **Structure comparison** to capture correlations between pixels.

The SSIM index between two images is computed as:

$$\text{SSIM}(I, \hat{I}) = \frac{(2\mu_I\mu_{\hat{I}} + C_1)(2\sigma_{I\hat{I}} + C_2)}{(\mu_I^2 + \mu_{\hat{I}}^2 + C_1)(\sigma_I^2 + \sigma_{\hat{I}}^2 + C_2)}$$

where μ_I and $\mu_{\hat{I}}$ are the means of I and $\hat{I}$, σ_I^2 and $\sigma_{\hat{I}}^2$ are the variances, $\sigma_{I\hat{I}}$ is the covariance between I and $\hat{I}$, and C_1, C_2 are small constants to stabilize the division.

SSIM values range from -1 to 1, where a value of 1 indicates perfect structural similarity between the two images. SSIM is widely used in image restoration, compression, and quality assessment tasks, providing a more comprehensive evaluation of perceptual quality compared to pixel-wise metrics like MSE.

III. Methodology

A. Experimental Setup

The foundation of our implementation was based on the open-source codebase provided by He et al. [1], which supports model inversion attacks against neural networks in edge-cloud collaborative inference systems. For privacy-preserving training, we integrated the Opacus library [12] into a standard PyTorch pipeline, enabling training with formal differential privacy guarantees. Opacus is a library for training PyTorch models with differential privacy. It allows to directly train private image classifiers, combining high performance with formal privacy guarantees. The images in CIFAR-10 were normalized using mean and standard deviation values precalculated for each RGB channel, and were augmented with random horizontal flips and affine transformations during training to improve generalization. All experiments were conducted using Google Colab with access to NVIDIA Tesla T4 GPUs. The primary dataset used was CIFAR-10, and a custom convolutional neural network (CIFAR10CNN) was trained on a 7-class subset, excluding the underperforming classes (bird, cat, and deer) based on preliminary evaluation.

Each model was trained for 50 epochs. Hyperparameters, such as clipping norm, target ε, and batch size, were systematically tuned. The evaluation metrics included standard classification accuracy, confusion matrices, and attack success rates of the model inversion attack.

B. Integrating Differential Privacy into Training

Differential privacy was introduced during training through DP-SGD. During each training iteration:

- Per-sample gradients were computed.
- Each gradient was clipped to a maximum ℓ_2 norm C to bound the sensitivity of the updates.
- Gaussian noise with standard deviation proportional to C was added to the averaged clipped gradients.
- Model parameters were updated using the privatized gradients.

The privacy accountant tracked the accumulated privacy loss over training iterations using RDP [13]. At the end of training, RDP guarantees were converted into (ε, δ)-differential privacy guarantees for final reporting.

Key parameters used for privacy-preserving training were:

- Target privacy parameter $\varepsilon \in [2, 50]$,
- Clipping norm $C = 3$,
- Noise multiplier σ adjusted according to target ε,
- Failure probability $\delta = 1/$number of training samples.

C. Integrating Sub-Sampling

Sub-sampling was employed to reduce the effective dataset size and leverage privacy amplification. Before training, a subset of 10,000 randomly selected samples was drawn from the training data without replacement.

The algorithm for sub-sampling is given as:

Before training, we apply a random sub-sampling procedure to the full dataset D containing n samples. Specifically, we randomly select m unique indices from the set $\{1, 2, \dots, n\}$ without replacement. The sub-sampled dataset D' is then constructed by including the data points from D corresponding to the selected indices. This reduced dataset D' is subsequently used for model training and for the computation of differential privacy guarantees.

This sub-sampled dataset was then used in place of the full training set for both standard and private model training. Sub-sampling also modified the calculation of δ in privacy accounting, as the dataset size was reduced.

D. Reproducing the Model Inversion Attack

The model inversion attack proposed by He et al. [1] was applied in a black-box setting. The attack procedure remained unchanged, ensuring consistency with baseline evaluations.

The attack was conducted for 50 iterations, querying the trained models and reconstructing input data from model outputs. Attack success was measured using inverse classification accuracy.

E. Evaluation Metrics

We evaluated model performance across the following dimensions:

- **Classification Accuracy**: Percentage of correctly classified test samples.
- **Confusion Matrix**: Visual analysis of per-class misclassification patterns.
- **Inverse Attack Accuracy**: Success rate of the model inversion attack in reconstructing correct labels.
- **Privacy Margin**: Difference between model train accuracy and attack success rate, used as a proxy for privacy risk.

Accuracy and confusion matrices were calculated after training to assess generalization, while attack success was evaluated separately to measure vulnerability to adversarial reconstruction.

IV. Results

This section presents the empirical findings from this study, comparing models trained with and without sub-sampling across different privacy levels. The results assess the trade-offs between utility (accuracy) and privacy (attack resilience) under varying values of the privacy budget ε.

A. Impact of ε on Model Utility

To evaluate how privacy levels affect model utility, we measured train and test accuracy across a range of ε values for models trained with and without sub-sampling. Fig 1 shows that both accuracy metrics increase as ε increases, consistent with the expected behavior of differential privacy. Lower ε values introduce more noise during training, reducing the model's ability to learn useful representations, while higher ε allows for greater signal retention and improved utility.

In the configuration without sub-sampling, the train accuracy improves from 42.08% at $\varepsilon = 2$ to 65.15% at $\varepsilon = 50$, with test accuracy following a similar trend, from 44.96% to 67.30%. This represents an absolute gain of approximately 23% in train accuracy and 22% in test accuracy across the privacy spectrum. However, the improvement starts to plateau beyond $\varepsilon = 20$, indicating diminishing returns: for example, the jump from $\varepsilon = 20$ to $\varepsilon = 50$ yields only a marginal increase of 0.4% in train accuracy and 0.56% in test accuracy.

In contrast, when sub-sampling is applied, accuracy is lower at every level of ε, but follows the same upward trend. Train accuracy improves from 35.27% at $\varepsilon = 2$ to 51.21% at $\varepsilon = 50$, while test accuracy increases from 37.03% to 54.26%. Notably, the relative improvement from $\varepsilon = 2$ to $\varepsilon = 50$ is still substantial: +15.94% (train) and +17.23% (test).

The accuracy gap between the two setups (with and without sub-sampling) also varies with ε. At $\varepsilon = 2$, the gap is modest: 6.81% for the train and 7.93% for the accuracy of the test. However, this gap widens at $\varepsilon = 50$, reaching 13.94% (train) and 13.04% (test), suggesting that sub-sampling has a slightly stronger dampening effect on learning as privacy constraints loosen.

Despite the performance drop, sub-sampling maintains utility, particularly in low-to-mid ε regimes. For example, at

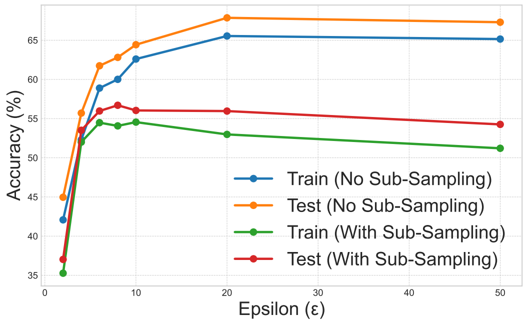

Fig. 1. Train and test accuracy vs. ε for models with and without sub-sampling.

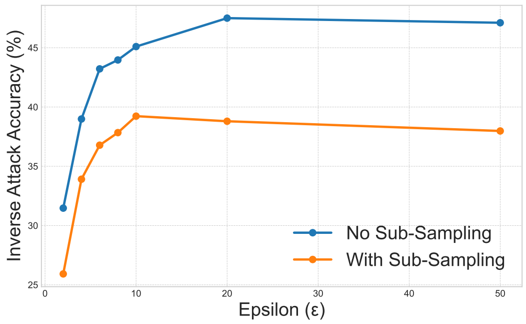

Fig. 2. Inverse attack accuracy across ε values with and without sub-sampling.

$\varepsilon = 10$, the model achieves 54.55% train and 56.04% test accuracy under sub-sampling, compared to 62.60% and 64.43% respectively without it: only 8% lower on both metrics.

B. Attack Success Across Privacy Levels

We next examine how vulnerable the trained models are to inversion attacks as privacy constraints change. Fig 2 illustrates the inverse attack accuracy across increasing ε values. As expected, models trained with higher ε values leak more information, resulting in higher attack success rates. This is because larger ε values correspond to weaker privacy guarantees, meaning less noise is added during training, and more structure from the data is preserved, including features that adversaries can exploit.

In the configuration without sub-sampling, attack accuracy increases from 31.47% at $\varepsilon = 2$ to 47.11% at $\varepsilon = 50$, showing a clear upward trend. The attack becomes significantly more successful in the range $\varepsilon = 2$ to $\varepsilon = 10$, with an increase of 13.63 percentage points. However, this trend also plateaus toward higher ε values: from $\varepsilon = 20$ to $\varepsilon = 50$, the increase in attack accuracy is only 0.39 percentage points.

When sub-sampling is applied, the attack success rate remains lower across the entire ε spectrum. For instance, at $\varepsilon = 2$, attack accuracy drops from 31.47% to 25.92%, a reduction of 5.55 percentage points. At $\varepsilon = 10$, the reduction is even more pronounced: from 45.10% (no sub-sampling) to 39.23% (with sub-sampling), representing a 13.02% relative decrease in adversarial success.

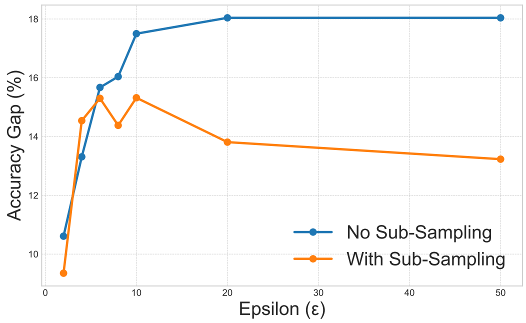

Fig. 3. Privacy margin across ε values.

Interestingly, while both configurations follow similar shapes, as ε increases. At $\varepsilon = 2$, sub-sampling reduces attack accuracy by 18%; at $\varepsilon = 50$, this advantage narrows to 9.8%. This suggests that sub-sampling is particularly effective at improving privacy for lower ε, and less effective when models are trained with relaxed privacy constraints.

C. Privacy Margin Analysis

Beyond raw accuracy and attack success, we assess the privacy margin, defined as the difference between train accuracy and attack accuracy, as a more direct indicator of data leakage. Fig 3 presents this metric across various values of ε. A larger gap implies that an adversary cannot recover more of the predictive capability of the model, indicating stronger privacy.

Without sub-sampling, the privacy margin increases steadily from 10.61% at $\varepsilon = 2$ to 18.04% at $\varepsilon = 50$, with most of the improvement occurring by $\varepsilon = 20$. Beyond this point, the margin plateaus, indicating that while utility continues to rise, the privacy gain saturates.

With sub-sampling, the privacy margin starts at 9.35% and increases more sharply at low ε, peaking at 15.30% at $\varepsilon = 6$. However, unlike the no-sub-sampling case, the margin begins to decline afterwards, dropping to 13.23% by $\varepsilon = 50$. This inverted trend suggests that while sub-sampling is especially effective at balancing utility and privacy under stricter budgets (low ε), its relative benefit diminishes as privacy constraints loosen.

Notably, at $\varepsilon = 4$, sub-sampling produces a slightly larger margin (14.54%) than the baseline (13.31%), marking the only crossover point in our data. This implies that in certain privacy regimes, sub-sampling may outperform even noisier non-sub-sampled training in obscuring sensitive information.

D. Correlation Between Accuracy and Privacy Risk

To directly examine the privacy-utility trade-off, we plot inverse attack accuracy against train accuracy. Figure 4 presents this relationship, with each point corresponding to a specific value of ε, and separate trends shown for models trained with and without sub-sampling. Linear regression lines are included to highlight the overall trend.

In both configurations, inverse attack accuracy increases with train accuracy, indicating a strong positive correlation.

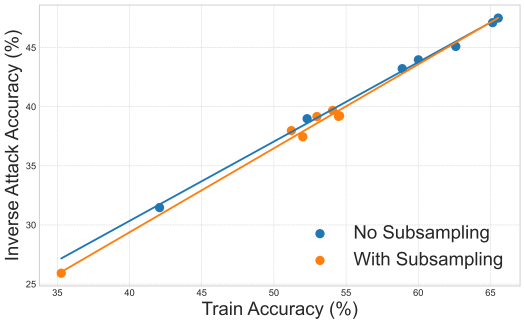

Fig. 4. Inverse attack accuracy vs. train accuracy, showing trade-off curves for both settings.

For models trained without sub-sampling, train accuracy ranges from 42.08% to 65.15%, with corresponding attack accuracy increasing from 31.47% to 47.11%. The relationship appears nearly linear, as shown by the fitted regression line.

The sub-sampling configuration follows a similar trend but with consistently lower attack accuracy at comparable utility levels. For instance, in the 52–54% train accuracy range, attack accuracy ranges from 37.46% to 39.69% with sub-sampling, compared to 38.99% to 43.97% without sub-sampling. At the upper end, a train accuracy of 51.21% (sub-sampling, $\varepsilon = 50$) corresponds to 37.98% attack accuracy, whereas 65.15% train accuracy (no sub-sampling) results in 47.11% attack accuracy.

Notably, the sub-sampling points are more tightly clustered, particularly between 51% and 55% train accuracy, where attack accuracy stays within a narrow 2-point range. This contrasts with the wider distribution seen in the baseline setting, suggesting more consistent privacy behavior under sub-sampling across mid-range utility levels.

E. Per-Class Performance at $\varepsilon = 10$

Fig 5 and 6 present confusion matrices for models trained with and without sub-sampling at $\varepsilon = 10$.

Across all evaluated classes, the model trained without sub-sampling achieves higher per-class accuracy than the sub-sampled counterpart. The most significant gap appears in the "dog" class (73.1% without sub-sampling versus 46.6% with sub-sampling), followed by noticeable differences in "truck" (52.4% vs. 43.7%) and "airplane" (60.4% vs. 54.4%).

Overall classification accuracy also favors the non-sub-sampled model (63.9%) compared to the sub-sampled version (56.1%). These results indicate that sub-sampling consistently reduces predictive performance both globally and at the class level. However, the sub-sampled model shows more dispersed error patterns, which may offer secondary privacy benefits by reducing concentrated feature memorization.

F. Comparison of Reconstruction Quality

We evaluate the impact of DP-SGD with sub-sampling on input reconstruction through both qualitative and quantitative analyses. Qualitative assessment is performed by visually comparing reconstructed images to their corresponding references, while quantitative evaluation uses Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) to measure pixel-level and structural similarities, respectively.

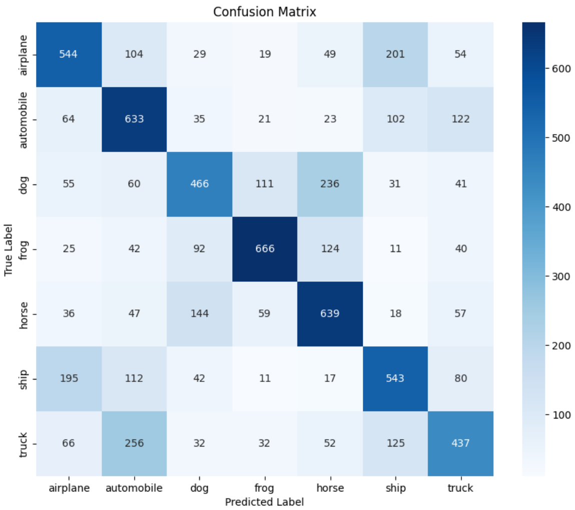

Fig. 5. Confusion matrix for model trained with sub-sampling at $\varepsilon = 10$.

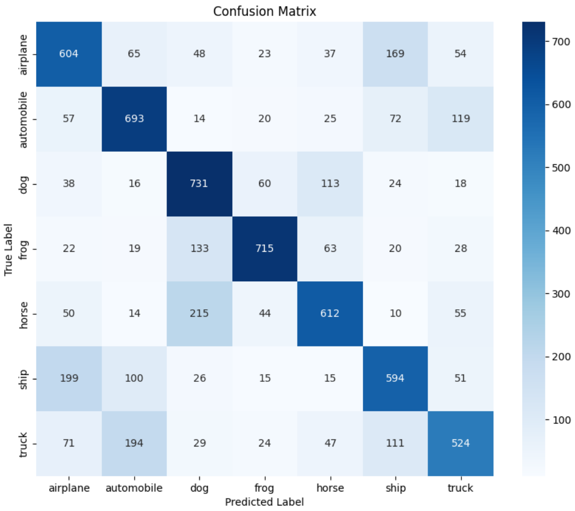

Fig. 6. Confusion matrix for model trained without sub-sampling at $\varepsilon = 10$.

Visual differences across classes are noticeable, as shown in Fig 7. In particular, for class 0 (bird), the reconstruction generated by the original attack appears more distorted and lacks structural resemblance to the reference image, whereas the reconstruction obtained under DP-SGD with sub-sampling retains more identifiable features. Similar patterns are observed across other classes, although some perceptual differences remain in all reconstructions.

These results quantitatively confirm that reconstructions obtained under the DP-SGD with sub-sampling defense achieve higher PSNR and SSIM values compared to those from the

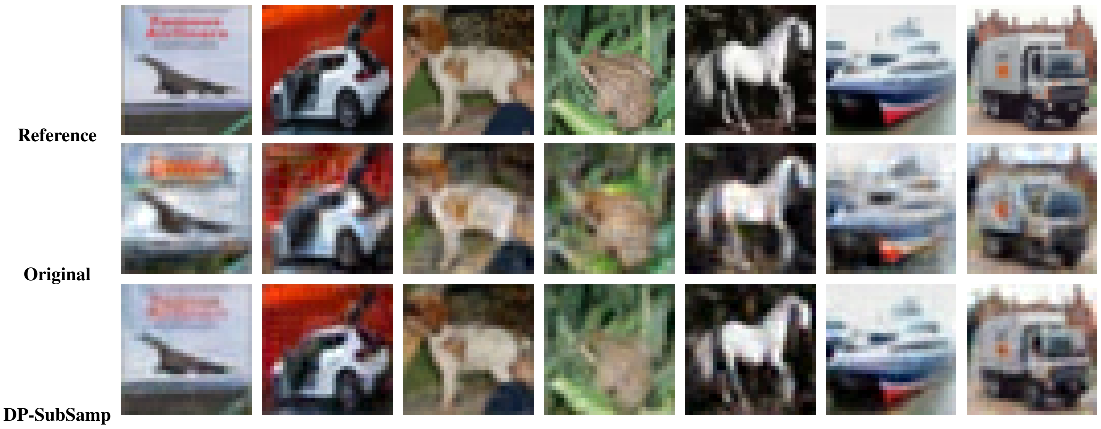

Fig. 7. Comparison of reference images, original attack reconstructions, and DP-SGD with sub-sampling reconstructions

TABLE I. Comparison of PSNR and SSIM values between original attack and DP-SGD with sub-sampling

Class	PSNR Original	PSNR DP Sub-Sampling	SSIM Original	SSIM DP Sub-Sampling
0	17.50	25.66	0.7223	0.898
1	18.14	26.02	0.7399	0.9323
5	19.17	26.91	0.7945	0.9488
6	18.15	24.44	0.6261	0.8865
7	19.12	25.86	0.8207	0.9510
8	18.82	28.42	0.7461	0.9544
9	19.12	25.67	0.7691	0.9481

original attack. Specifically, PSNR increases by approximately 6–10 dB across classes, while SSIM improves by approximately 0.15–0.21 units. Although PSNR and SSIM values are higher, visual differences between reconstructions and original images remain noticeable.

V. Discussion

This study investigated the effectiveness of sub-sampling combined with DP-SGD in mitigating model inversion attacks in a black-box setting. The results show that while sub-sampling consistently reduced the inverse attack success rate across different privacy budgets, reconstructed images remained visually similar to their references, as indicated by higher PSNR and SSIM values. In this section, we analyze these findings in detail, discuss their implications for privacy protection, and identify technical limitations and future directions.

A. Effect of Optimizer Choice

First, it is important to acknowledge that our training configuration differed from the original baseline proposed by He et al. [1]. However, recent work has shown that under differential privacy constraints, the noise added during training can distort Adam's second moment estimates, causing it to behave more like DP-SGD with momentum and sometimes diminishing its advantages [14]. Since DP-SGD inherently adds substantial noise to gradient updates, the choice of optimizer—especially one that stabilizes or adapts gradient steps—can materially affect the resulting privacy-utility trade-off. In preliminary experiments, we observed that SGD under DP-SGD settings required longer to reach comparable training accuracy levels, and often converged to slightly lower final performance. Thus, part of the observed differences in model accuracy and attack vulnerability relative to prior work may be attributed to optimizer effects, independent of privacy mechanisms themselves.

B. Sub-Sampling and Attack Success Rates

Despite the optimizer differences, the core trends remain robust. Sub-sampling, when combined with DP-SGD, consistently reduced the inverse attack success rate across the entire privacy spectrum. For example, at $\varepsilon = 20$, attack success decreased from 47.50% (no sub-sampling) to 38.80% (with sub-sampling), corresponding to an absolute reduction of 8.70%. Privacy margins, defined as the difference between train accuracy and attack accuracy, were also generally larger at low-to-mid ε values under sub-sampling. This indicates that sub-sampling improves the separation between what the model learns and what an adversary can reconstruct, at least under classification-based inversion metrics.

C. Effect of Relaxed Privacy Budgets

As ε increases beyond 20, the relative advantage of sub-sampling diminishes. Both attack success rates and privacy margins converge between sub-sampled and non-sub-sampled models, suggesting that the protective amplification offered by sub-sampling is strongest under stricter privacy regimes. At $\varepsilon = 50$, although an absolute reduction of 9.13% in attack success was still observed, the difference becomes less meaningful because the privacy guarantees are already weak at such high ε levels.

D. Perceptual Leakage in Reconstructions

While sub-sampling reduced classification-based leakage, reconstructed images under DP-SGD with sub-sampling exhibited higher PSNR (up to +10 dB) and SSIM (up to +0.21) values relative to the original attack. These higher similarity scores indicate better preservation of pixel-level and structural features, making perceptual details more identifiable.

However, despite this increase in low-level similarity, the overall classification attack success rate decreased, suggesting that semantic information useful for adversaries became harder to extract. This highlights an important tension: differential privacy mechanisms like DP-SGD primarily target membership inference resistance, not necessarily perceptual feature suppression. As such, visual similarity improvements do not directly imply greater privacy leakage. Care must be taken when interpreting PSNR or SSIM as privacy indicators, since they capture different aspects of model vulnerability.

E. Alternative Methods and Future Work

Alternative optimizers could potentially improve both model utility and privacy protection. DP-AdamBC, which combines adaptive gradient scaling with differential privacy guarantees, has demonstrated better privacy-utility trade-offs compared to vanilla DP-SGD. Incorporating such optimizers may mitigate the convergence inefficiencies observed with SGD under privacy constraints. In addition, more aggressive hyperparameter tuning, including adaptive clipping strategies and dynamic batch sizing, could improve privacy amplification effects.

Future work should systematically explore these enhancements. Specifically, evaluating on more complex datasets, such as CIFAR-100 or Tiny ImageNet [15], would help verify whether the observed trends generalize to higher-resolution and more diverse data distributions. Extending the threat model to white-box attacks or stronger reconstruction algorithms, such as Generative Model Inversion [16], could provide a more rigorous assessment of perceptual privacy risks. Finally, developing training techniques that explicitly obscure learned feature representations, rather than focusing only on preventing membership inference, could offer a promising direction for improving perceptual privacy.

VI. Conclusion

In this work, we investigated the effectiveness of combining privacy-enhancing sub-sampling techniques with DP-SGD to defend against model inversion attacks in black-box settings. Our experimental results show that sub-sampling consistently reduces inversion attack success rates, particularly under stricter privacy budgets (low ε), while maintaining acceptable model utility. Although the combination improves privacy margins, we observe that reconstructed images still retain some perceptual similarity, highlighting the limitations of current defenses against more advanced leakage.

Our findings suggest that sub-sampling, together with formal differential privacy guarantees, can meaningfully improve the privacy-utility trade-off in real-world machine learning deployments.

Acknowledgements

We are deeply grateful to our supervisor, Leonhard Grosse, whose support and encouragement were invaluable throughout the development of this thesis. His perspective on the subject sparked our curiosity and motivated us to approach our work with greater critical thinking and enthusiasm.

References

[1] Z. He, T. Zhang, and R. B. Lee, “Attacking and protecting data privacy in edge–cloud collaborative inference systems,” *IEEE Internet of Things Journal*, vol. 8, no. 12, pp. 9706–9716, 2021.

[2] Òscar Lorente, I. Riera, and A. Rana, “Image classification with classic and deep learning techniques,” 2021. [Online]. Available: https://arxiv.org/abs/2105.04895

[3] A. Krizhevsky, “Learning multiple layers of features from tiny images,” https://www.cs.toronto.edu/~kriz/cifar.html, 2009, technical report, University of Toronto.

[4] C. Dwork and A. Roth, *The Algorithmic Foundations of Differential Privacy*, ser. Foundations and Trends in Theoretical Computer Science. Now Publishers Inc., 2014, vol. 9, no. 3–4.

[5] S. L. Warner, “Randomized response: A survey technique for eliminating evasive answer bias,” *Journal of the American Statistical Association*, vol. 60, no. 309, pp. 63–69, 1965.

[6] C. Dwork, F. McSherry, K. Nissim, and A. Smith, “Calibrating noise to sensitivity in private data analysis,” in *Theory of Cryptography Conference (TCC)*. Springer, 2006, pp. 265–284.

[7] D. Yu, G. Kamath, J. Kulkarni, T.-Y. Liu, J. Yin, and H. Zhang, “Individual privacy accounting for differentially private stochastic gradient descent,” 2024. [Online]. Available: https://openreview.net/forum?id=D4JQEKlTyG

[8] B. Balle, G. Barthe, and M. Gaboardi, “Differential privacy by sampling,” *Google Scholar*, 2018, preprint. [Online]. Available: https://par.nsf.gov/servlets/purl/10106941

[9] I. Mironov, “Rényi differential privacy,” in *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*. IEEE, 2017, pp. 263–275.

[10] A. Rényi, “On measures of entropy and information,” in *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*. University of California Press, 1961.

[11] J. Nilsson and T. Akenine-Möller, “Understanding ssim,” *arXiv preprint arXiv:2006.13846*, 2020.

[12] O. Team, “Building an image classifier with differential privacy,” 2023. [Online]. Available: https://opacus.ai/tutorials/building_image_classifier

[13] Opacus Contributors, “Opacus Privacy Engine Documentation,” 2024. [Online]. Available: https://opacus.ai/api/privacy_engine.html

[14] Q. Tang, F. Shpilevskiy, and M. Lécuyer, “Dp-adambc: Your dp-adam is actually dp-sgd (unless you apply bias correction),” in *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI-24)*. Vancouver, Canada: Association for the Advancement of Artificial Intelligence (AAAI), 2024.

[15] L. Yao and J. Miller, “Tiny imagenet classification with convolutional neural networks,” *CS 231N*, vol. 2, no. 5, p. 8, 2015.

[16] Y. Zhang, R. Jia, H. Pei, W. Wang, B. Li, and D. Song, “The secret revealer: Generative model-inversion attacks against deep neural networks,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 253–261.