DynaNoise: Dynamic Probabilistic Noise Injection for Defending Against Membership Inference Attacks

Javad Forough
Imperial College London
London, UK
j.forough@imperial.ac.uk

Hamed Haddadi
Imperial College London
London, UK
h.haddadi@imperial.ac.uk

Abstract

Membership Inference Attacks (MIAs) pose a significant risk to the privacy of training datasets by exploiting subtle differences in model outputs to determine whether a particular data sample was used during training. These attacks can compromise sensitive information, especially in domains such as healthcare and finance, where data privacy is paramount. Traditional mitigation techniques, such as static differential privacy, rely on injecting a fixed amount of noise during training or inference. However, this approach often leads to a detrimental trade-off: the noise may be insufficient to counter sophisticated attacks or, when increased, may substantially degrade model performance. In this paper, we present DynaNoise, an adaptive approach that dynamically modulates noise injection based on query sensitivity. Our approach performs sensitivity analysis using measures such as Shannon entropy to evaluate the risk associated with each query and adjusts the noise variance accordingly. A probabilistic smoothing step is then applied to re-normalize the perturbed outputs, ensuring that the model maintains high accuracy while effectively obfuscating membership signals. We further propose an empirical metric, the *Membership Inference Defense Privacy–Utility Trade-off (MIDPUT)*, which quantifies the balance between reducing attack success rates and preserving the target model's accuracy. Our extensive evaluation on several benchmark datasets demonstrates that DynaNoise not only significantly reduces MIA success rates but also achieves up to a four-fold improvement in the MIDPUT metric compared to the state-of-the-art. Moreover, DynaNoise maintains competitive model accuracy while imposing only marginal inference overhead, highlighting its potential as an effective and efficient privacy defense against MIAs.

1 Introduction

Machine learning has revolutionized many domains by leveraging vast amounts of data to achieve impressive performance [3–5, 13, 14, 27]. However, this success comes at a cost, where sensitive information from training datasets may be inadvertently memorized, posing serious privacy risks [2, 17, 24, 25]. For example, in a scenario where a hospital deploys a predictive model to diagnose diseases; if an attacker can determine whether a patient's record was part of the training set, it may reveal that the patient has visited the hospital, thereby compromising their privacy. Similarly, in finance, membership leakage could expose clients' investment histories. Such risks underscore the need for robust defenses against privacy attacks.

Membership Inference Attack (MIA) [19], as shown in Figure 1, exploits subtle differences in a model's output behavior to determine whether a specific data record was used during training, thereby threatening the confidentiality of individual data points [19]. Conventional defenses, such as differential privacy, introduce a fixed level of noise during training or inference to obscure membership information [1]. While these approaches offer formal privacy guarantees, they force an inherent trade-off between privacy and utility. In many practical settings, uniformly adding noise can significantly degrade model performance, yet reducing the noise level may leave the model susceptible to advanced membership inference attacks.

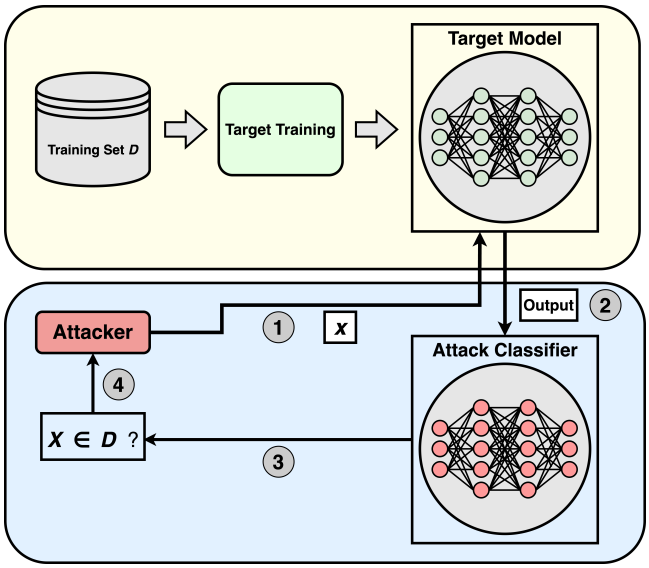


Figure 1: An illustration of how MIA works.

To address these challenges, we introduce DynaNoise, an adaptive noise injection approach that modulates the privacy noise dynamically based on query sensitivity. Our method first assesses the risk of each query through sensitivity analysis, by utilizing metrics such as Shannon entropy [10], and then adjusts the noise variance accordingly. A subsequent probabilistic smoothing step is applied to re-normalize the perturbed outputs, ensuring that the model retains high predictive accuracy while effectively obfuscating membership signals. Another key innovation of our approach is the introduction of a novel empirical metric, the *Membership Inference Defense Privacy–Utility Trade-off (MIDPUT)*, which quantitatively captures the balance between the reduction in attack success rates and the preservation of model accuracy. Our experimental results on CIFAR-10, ImageNet-10, and SST-2 datasets demonstrate that DynaNoise not only significantly reduces membership inference

attack success rates but also achieves up to a four-fold improvement in the MIDPUT metric compared to existing defenses such as SELENA.

The main contributions of this paper are threefold:

(1) Proposal of a novel defense mechanism against membership inference attacks called DynaNoise, that dynamically adjusts the noise level based on query sensitivity, thereby providing stronger privacy protection with minimal impact on target model accuracy.
(2) Proposal of a new empirical metric called *Membership Inference Defense Privacy–Utility Trade-off (MIDPUT)*, that quantifies the trade-off between the reduction in attack success rates and the loss in model performance. This metric enables a more detailed and precise evaluation of privacy defenses.
(3) Comprehensive evaluation through extensive experiments on multiple benchmark datasets and models using different system parameters.

The remainder of this paper is organized as follows. In Section 2, we review the state-of-the-art in membership inference attacks and defenses. Section 3 and 4 explain our problem statement and threat model, respectively. Section 5 reviews the preliminary concepts related to this work. Section 6 details the design and implementation of DynaNoise, including our adaptive noise injection and the MIDPUT metric. Section 7 presents our experimental setup, results, and a discussion of the advantages and limitations of the proposed approach. Finally, Section 8 concludes the paper and discusses future research directions.

2 Related Work

2.1 Membership Inference Attacks (MIAs)

Membership inference attacks can be broadly categorized into two types: shadow training-based attacks [9, 15, 19] and metric-based attacks [20, 26].

Shadow training-based attacks. Shokri et al. [19] introduced one of the earliest frameworks for membership inference attacks. In their approach, the adversary constructs multiple shadow models that mimic the behavior of the target model by training them on data drawn from a distribution similar to that of the target. The outputs of these shadow models on inputs with known membership statuses are then used to train an attack classifier that distinguishes between members and non-members. Although this method can achieve high attack accuracy, its effectiveness relies on access to a dataset closely resembling the target model's training data.

Salem et al. [15] build upon this framework by demonstrating that a single shadow model can often suffice to perform effective membership inference. By reducing the number of shadow models, their method decreases the overall computational cost and query burden, making the attack more practical. However, the approach still assumes that the adversary can obtain or generate data from a similar distribution as the target model, which may not always be realistic.

Long et al. [9] further refine shadow training-based attacks by optimizing both the construction of the shadow models and the design of the attack classifier. Their improvements reduce the number of queries needed and enhance the robustness of the attack, even when the adversary's resources are limited. This work underscores that shadow training-based methods remain potent, though they continue to depend on the availability of representative auxiliary data.

Metric-based attacks. In contrast to shadow training-based methods, metric-based attacks infer membership directly from the target model's output by evaluating specific statistical metrics. Yeom et al. [26] propose a loss-based attack that exploits the tendency of models to incur lower loss on training samples compared to non-training samples. By setting an appropriate loss threshold, the adversary can effectively distinguish between members and non-members. While this approach is straightforward and does not require training additional models, its effectiveness is highly sensitive to the chosen threshold and may vary across different models.

Similarly, Song et al. [20] explore confidence-based attacks, which leverage the observation that models often yield higher prediction confidence for training samples. In this method, membership is inferred by comparing the maximum confidence score against a threshold, which can be either fixed or adjusted on a per-class basis. Although this technique is simple and requires little auxiliary information, it is also sensitive to the threshold setting, thereby affecting its overall robustness.

2.2 Membership Inference Defenses

There are several recent works aimed at addressing MIAs, which are presented and compared in Table 1. Subsequently, we provide a detailed overview of each method along with a discussion of their respective limitations.

Abadi et al. [1] propose DP-SGD, a defense that integrates differential privacy directly into the training process by clipping per-example gradients and adding calibrated Gaussian noise. Their method employs a moments accountant to tightly track the cumulative privacy loss, thereby offering formal, provable privacy guarantees. This approach ensures that the entire model adheres to a specified privacy budget. However, because DP-SGD uses a static noise injection scheme that does not adapt to the sensitivity of individual queries, the large noise required to meet a strict privacy budget often leads to a significant drop in accuracy. Consequently, the overall utility of the trained model may suffer, especially in complex or high-dimensional tasks.

In another work, Nasr et al. [11] introduce adversarial regularization, an approach that augments the training process with a min-max game. In their approach, an auxiliary attack model is trained concurrently with the main classifier to predict membership information, while the classifier is regularized to minimize the success of this inference attack. This empirical strategy has been shown to reduce the effectiveness of membership inference attacks with only a modest loss in utility. Nevertheless, because it does not offer formal differential privacy guarantees and its performance is sensitive to the tuning of adversarial hyperparameters, the method may not be robust across different datasets and deployment scenarios.

Moving from modifications in the training procedure, Jia et al. [7] propose MemGuard, a post-processing technique that directly perturbs the output confidence scores using adversarial noise. By

Table 1: Comparison of defense mechanisms against membership inference attacks.

Defense Method	Year	Approach/Mechanism	Privacy Guarantee	Utility Impact	Comp. Overhead
DP-SGD [1]	2016	Gradient clipping + additive Gaussian noise (static noise injection)	Formal (provable DP)	High accuracy drop	Moderate
Adversarial Regularization [11]	2018	Min-max adversarial training integrating an attack model into training	Empirical	Low to moderate accuracy drop	Moderate
MemGuard [7]	2019	Post-processing: adversarial noise added to confidence score vectors	Empirical	Minimal accuracy drop	Low
SELENA [21]	2022	Adaptive ensemble (Split-AI + Self-Distillation) with dynamic noise injection	Empirical	Minimal accuracy drop	Moderate
DynaNoise (This work)	2025	Adaptive noise injection based on query sensitivity (sensitivity analysis, dynamic noise variance modulation, probabilistic smoothing)	Empirical	Minimal accuracy drop	Low

transforming these outputs into adversarial examples, MemGuard aims to confuse any attack model attempting to distinguish between training and non-training samples. This method maintains high utility since the perturbations are designed to minimally affect the predicted labels. However, its reliance on fixed perturbation patterns means that it may be less effective if an adversary adapts to the specific noise pattern used.

Finally, Tang et al. [21] take an ensemble-based approach with SELENA, which trains multiple sub-models on overlapping subsets of the training data and then distills their outputs into a single prediction through a self-distillation process. This adaptive inference strategy selectively aggregates predictions from sub-models that have not seen the queried sample, thereby enhancing the privacy-utility trade-off. Despite its advantages, the ensemble inference process introduces moderate computational overhead due to the requirement of running multiple sub-models concurrently in training phase, which can be a drawback in resource-constrained settings.

Despite these promising approaches, there are still important challenges that limit their practical effectiveness. One key limitation is that the fixed nature of noise injection does not account for the varying sensitivity of different queries. This rigidity can result in excessive noise for low-risk queries, unnecessarily degrading utility, or insufficient noise for high-risk queries, failing to adequately obfuscate membership information. Additionally, ensemble-based defenses, such as SELENA, incur significant computational and time overhead due to the need to train and evaluate multiple sub-models. These requirements may not be feasible in resource-constrained environments. To address these limitations, our work proposes a dynamic noise injection mechanism that adjusts the noise level based on query sensitivity, thereby achieving a more balanced trade-off between privacy protection and model performance, while incurring only negligible computational overhead.

3 Problem Statement

Membership inference attacks exploit subtle discrepancies between a model's outputs on training data and those on unseen data, thereby threatening the privacy of individuals whose information is used during training. Traditional privacy-preserving techniques, such as static differential privacy, address this risk by adding a fixed amount of noise to the model outputs. However, the uniform application of noise fails to account for the heterogeneous sensitivity of different queries. For low-risk queries, the excessive noise degrades model accuracy and adversely affects user experience. Conversely, for high-risk queries, an insufficient noise level may not sufficiently mask membership information, leaving the model vulnerable to membership inference attacks.

The central challenge, therefore, is to balance the trade-off between privacy and utility. A static noise injection strategy may either compromise model performance or inadequately protect against adversaries, thereby necessitating a more refined approach that dynamically adapts to the varying sensitivity across different queries.

4 Threat Model

In our threat model, the adversary has black-box access to the target model. That is, the attacker can submit any input query q and obtain its corresponding prediction vector $f(q)$. The adversary is assumed to know the input/output format (for example, the number of classes and the range of confidence values) and may either have knowledge of the model's architecture and training algorithm or only interact with the model via a machine-learning-as-a-service (MLaaS) platform where such internal details are hidden.

The adversary's goal is to infer whether a given data record q was used in training the target model. Formally, the attacker aims to distinguish between the hypotheses:

$$H_0 : \quad q \notin D_{\text{train}},$$

$$H_1 : \quad q \in D_{\text{train}}.$$

Where D_{train} stands for the training dataset. We assume that the adversary may also have some background knowledge about the population from which D_{train} is drawn, such as general statistics on feature distributions. An attack is considered successful if the adversary can determine the membership status of q with high accuracy.

Our proposed defense mechanism, DynaNoise, is designed under these assumptions. By dynamically adjusting the noise added to the model's outputs based on the sensitivity of each query, DynaNoise aims to blur the distinction between training and non-training data, thereby reducing the adversary's ability to reliably infer membership.

5 Preliminaries

In this section, we explain the background concepts related to this work. In addition, Table 2 presents a summary of the notations used throughout this paper.

5.1 Static Differential Privacy and Its Limitations

Static differential privacy (DP) is a formal framework for protecting individual data records by adding random noise during model training or inference [1, 6]. The goal is to ensure that the inclusion or exclusion of any single data point has only a limited impact on the model's output, thus protecting individual privacy.

A randomized mechanism M (such as a learning algorithm or model output function) is said to satisfy (ϵ, δ)-differential privacy if, for all possible outputs S, and for any pair of neighboring datasets D and D' (which differ in only one data record), the following inequality holds:

$$\Pr[M(D) \in S] \leq e^{\epsilon} \cdot \Pr[M(D') \in S] + \delta,$$

where:

- $\Pr[M(D) \in S]$: the probability that the mechanism produces an output within the set S, where the randomness arises from the noise intentionally added by the mechanism.
- $M(D)$: the randomized output (e.g., model prediction or learned parameters) when the mechanism operates on dataset D,
- S: any subset of possible outputs,
- D, D': neighboring datasets differing in exactly one entry,
- e^{ϵ}: a multiplicative bound controlling the degree of output similarity between neighboring datasets; e is Euler's number (approximately 2.718)
- ϵ: the privacy budget (smaller values indicate stronger privacy),
- δ: the probability of violating the ϵ-bound (typically a small value like 10^{-5}).

While this approach provides mathematically rigorous privacy guarantees, it usually applies the same level of noise to all data points or queries, regardless of their actual risk level. This static and uniform noise injection leads to a key limitation:

- If the noise is too large, it can significantly degrade the model's accuracy and utility.
- If the noise is too small, it may fail to protect against advanced membership inference attacks.

This inherent rigidity in static DP motivates the need for more flexible strategies, such as adaptive or dynamic noise injection mechanisms that tailor the amount of noise based on the sensitivity of each query or prediction.

5.2 Information Leakage in Deep Learning

Deep neural networks are known to exhibit complex behavior that may inadvertently reveal information about their training data. Several works [12, 22, 23] have shown that even models with strong generalization capabilities can overfit on certain examples, thereby creating a gap between the output distributions for training and non-training data. This leakage is often measured using differences in loss or divergence in prediction distributions, which can be formally expressed by metrics such as Kullback-Leibler divergence:

$$D_{KL}(P \parallel Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)},$$

where P and Q represent the output distributions for training and non-training examples, respectively.

Understanding and quantifying this leakage is critical for designing effective privacy-preserving mechanisms. The degree of leakage can inform the design of adaptive noise injection strategies that modulate the amount of noise based on the risk associated with each query, thereby reducing the adversary's ability to distinguish between members and non-members while preserving the utility of the model.

Table 2: Summary of notations used in the paper.

Symbol	Description
$f(q)$	Model logits for input q.
$\tilde{f}(q)$	Noisy logits: $f(q) + \eta$.
$\hat{f}(q)$	Final probabilities (after smoothing).
k	Number of classes.
$\mathbf{p}$	Softmax of $f(q)$.
$H(p)$	Entropy of $\mathbf{p}$.
$R(q)$	Sensitivity score of q.
σ_0^2	Base noise variance.
λ	Noise scaling parameter.
$\sigma(q)^2$	Adjusted noise variance.
η	Gaussian noise vector.
T	Temperature for smoothing.
ASR	Attack Success Rate.
MIDPUT	Overall Privacy–Utility Trade-off metric.
MIDPUT_C, MIDPUT_L, MIDPUT_S	Per-attack MIDPUT metrics (Confidence, Loss, Shadow).

6 Proposed Approach

Our proposed approach dynamically mitigates membership inference attacks by adapting the noise injection process to the sensitivity of each individual query. The step-by-step processes of the proposed approach is provided in Algorithm 1. The system is composed of three primary components, as described below:

- **Sensitivity Analysis:**
  In this module, we evaluate the risk associated with each query based on the model's output probability distribution. Let $p = (p_1, p_2, \ldots, p_k)$ denote the output probability vector

for a given input, where k is the number of classes. We compute the Shannon entropy [18]:

$$H(p) = -\sum_{i=1}^{k} p_i \log(p_i),$$

which quantifies the uncertainty of the prediction. We then define the sensitivity score $R(q)$ for query q as:

$$R(q) = 1 - \frac{H(p)}{\log k},$$

ensuring that $R(q) \in [0, 1]$. A higher $R(q)$ indicates that the model is highly confident (i.e., lower entropy) in its prediction, thus representing a greater risk for membership inference attacks.

- **Dynamic Noise Injection:**
  Using the computed sensitivity score, we dynamically adjust the noise injected into the model's output. We define the noise variance as a function of the sensitivity score:

  $$\sigma(q)^2 = \sigma_0^2 \left(1 + \lambda R(q)\right),$$

  where σ_0^2 is the base noise variance and λ is a scaling parameter that amplifies the noise for higher-risk queries. The noise η is then sampled from a Gaussian distribution:

  $$\eta \sim \mathcal{N}(0, \sigma(q)^2),$$

  and added to the raw output $f(q)$ of the model:

  $$\tilde{f}(q) = f(q) + \eta.$$

  This formulation ensures that queries with a higher sensitivity score receive proportionally more noise, thus effectively obfuscating any distinguishing membership signals.
- **Probabilistic Smoothing:**
  To mitigate potential distortions introduced by noise injection while maintaining model accuracy, we apply a probabilistic smoothing operation by re-normalizing the perturbed outputs using a softmax function with a temperature parameter $T > 1$:

  $$\hat{f}(q) = \text{softmax}\left(\frac{\tilde{f}(q)}{T}\right).$$

  The temperature T controls the sharpness of the resulting probability distribution, providing a trade-off between smoothing and maintaining the discriminative power of the original output.

By integrating sensitivity analysis, dynamic noise injection, and probabilistic smoothing, our approach adapts to the context of each inference request. Hence, this specific defense mechanism effectively obfuscates membership signals in high-risk queries while maintaining overall data utility and predictive accuracy.

7 Evaluation

We evaluated our approach on several benchmark datasets and model architectures that are explained in Section 7.3. Our analysis focuses on three primary aspects: (i) the effectiveness of membership inference attacks; (ii) the impact on model accuracy (i.e., utility); and (iii) the computational cost of the defense mechanism.

Algorithm 1 DynaNoise: Adaptive Noise Injection Based on Query Sensitivity

Input: Trained model $f(\cdot)$ with k output classes, base noise variance σ_0^2, scaling parameter λ, temperature parameter $T > 1$, and input query q.
Output: $\hat{f}(q)$, the final probability vector after noise injection and smoothing.
Step1 - Compute Raw Output:
  Compute logits: $\mathbf{z} \leftarrow f(q) \in \mathbb{R}^k$.
  Compute probability vector: $\mathbf{p} \leftarrow \text{softmax}(\mathbf{z})$.
Step2 - Sensitivity Analysis:
  Compute Shannon entropy: $H(\mathbf{p}) \leftarrow -\sum_{i=1}^{k} p_i \log(p_i)$.
  Set sensitivity score: $R(q) \leftarrow 1 - \frac{H(\mathbf{p})}{\log k}$.
Step3 - Dynamic Noise Injection:
  Compute noise variance: $\sigma(q)^2 \leftarrow \sigma_0^2 \left(1 + \lambda R(q)\right)$.
  Sample noise: $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0},\ \sigma(q)^2 I)$.
  Perturb logits: $\tilde{\mathbf{f}}(\mathbf{q}) \leftarrow \mathbf{f}(\mathbf{q}) + \boldsymbol{\eta}$.
Step4 - Probabilistic Smoothing:
  Compute final output: $\hat{f}(q) \leftarrow \text{softmax}\left(\frac{\tilde{\mathbf{f}}(\mathbf{q})}{T}\right)$.
Return: $\hat{f}(q)$.

7.1 Metrics

The following metrics were used to evaluate the performance of our approach:

- **Attack Success Rate (ASR):** The fraction of correctly inferred membership, defined as

  $$\text{ASR} = \frac{N_{\text{correct}}}{N_{\text{total}}},$$

  where N_{correct} is the number of correct membership predictions and N_{total} is the total number of predictions.
- **Model Accuracy:** The overall classification accuracy of the target model, measured both before and after applying a defense.
- **Membership Inference Defense Privacy–Utility Trade-off (MIDPUT):** Our proposed metric for evaluating the trade-off between privacy and utility. Let

  $$\Delta_{\text{acc}} = \text{test\_acc}_{\text{no\_def}} - \text{test\_acc}_{\text{def}},$$

  and for each attack type $A \in \{\text{conf}, \text{loss}, \text{shadow}\}$,

  $$\Delta_A = \text{attack\_acc}^{(A)}_{\text{no\_def}} - \text{attack\_acc}^{(A)}_{\text{def}}.$$

  Then, the overall MIDPUT is defined as

  $$\text{MIDPUT} = \left(\frac{\Delta_{\text{conf}} + \Delta_{\text{loss}} + \Delta_{\text{shadow}}}{3}\right) - \Delta_{\text{acc}},$$

  and the per-attack MIDPUT metrics are given by

  $$\text{MIDPUT}_A = \Delta_A - \Delta_{\text{acc}}, \quad A \in \{\text{conf}, \text{loss}, \text{shadow}\}.$$

  The MIDPUT metric is bounded within the range $[-1, 1]$, where values closer to 1 represent strong privacy gains with minimal utility loss, and values approaching -1 indicate poor trade-offs with greater accuracy degradation than privacy improvement.

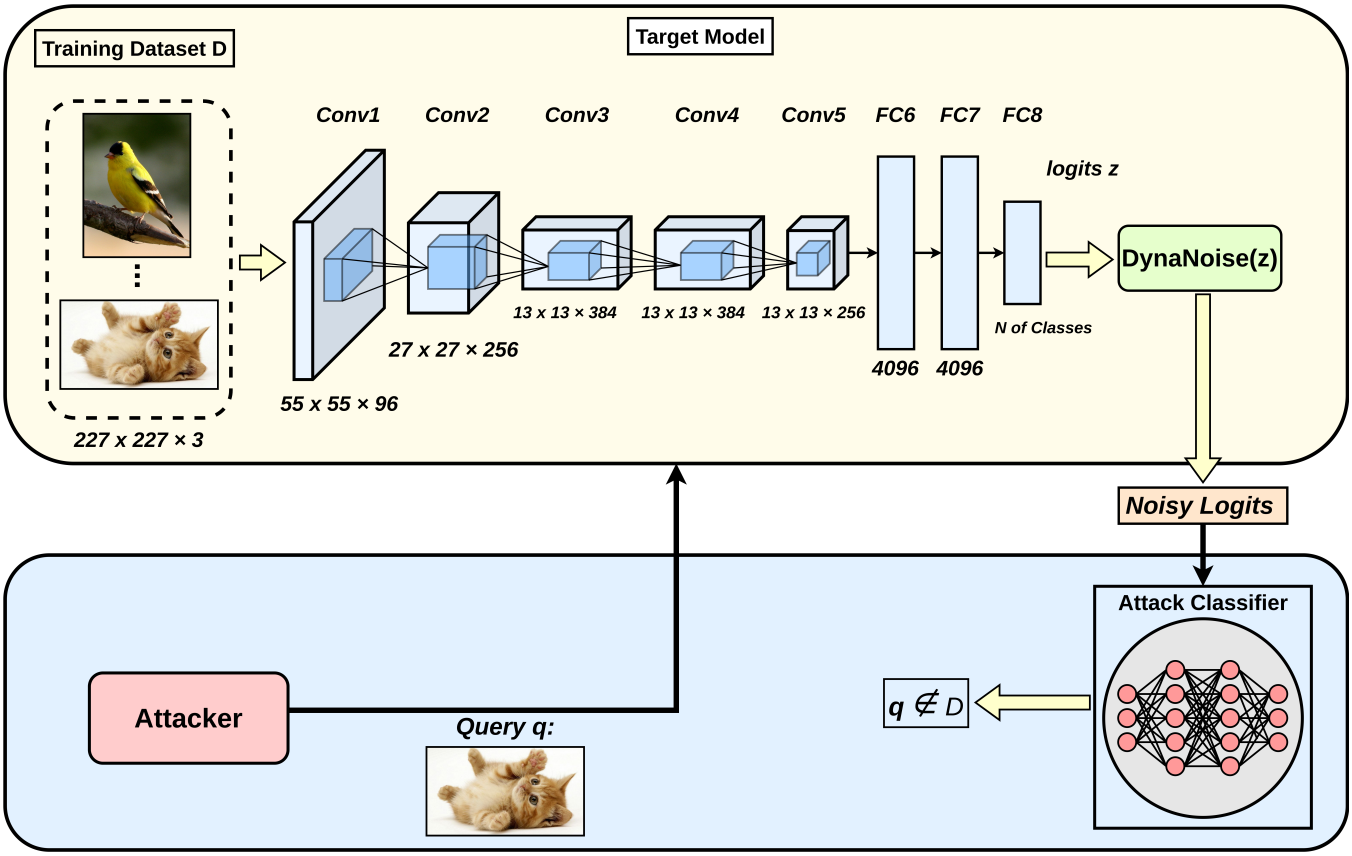


Figure 2: Overview of the proposed DynaNoise approach integrated into the AlexNet deep learning model for illustration. The noise injection process adapts to the model's uncertainty, providing enhanced obfuscation against membership inference attacks.

7.2 Membership Inference Attacks

We implemented three distinct membership inference attacks:

(1) **Confidence Threshold Attack:** Predicts a sample as a member if the maximum predicted probability exceeds a fixed threshold τ:

$$\text{Predict "in" if } \max_i p_i > \tau.$$

In our evaluation, we set $\tau = 0.9$

(2) **Loss Threshold Attack:** Computes the cross-entropy loss

$$\ell = -\log p(y)$$

for the true label y, and predicts membership if $\ell < \gamma$, where γ is a fixed threshold. In our evaluation, we set $\gamma = 0.5$.

(3) **Shadow-Model Attack:** Trains a shadow model on an auxiliary dataset drawn from the same distribution as the target model's training dataset. From the shadow model's outputs, features such as maximum confidence, cross-entropy loss, and confidence margin are extracted to train a binary classifier $A(\mathbf{x})$, where $\mathbf{x}$ is the feature vector. The membership decision is then based on the classifier's output. In our setup, we allocate 70% of the available data to training and evaluation the target model, while the remaining 30% is reserved for constructing the shadow model and training the attack classifier. We use the same architecture for the shadow model as the target model to closely mimic its behavior, and we employ a logistic regression classifier as the final attack model.

Each membership inference attack is systematically evaluated under the following three defense conditions:

(1) **No Defense:** The target model's raw outputs are directly exposed to the attacker without any privacy preserving approach applied.

(2) **SELENA Defense:** The SELENA defense approach is employed, which leverages ensemble learning and self-distillation to mitigate membership leakage prior to the execution of the attack.

(3) **DynaNoise Defense:** The proposed DynaNoise approach is applied to the model outputs in a post-processing step, where it dynamically injects calibrated probabilistic noise based on query sensitivity to conceal membership signals while preserving overall model utility.

7.3 Datasets and Models

Experiments were conducted on three widely used benchmark datasets that span both image and text domains. The datasets are described in detail below:

- **CIFAR-10**[1]**:** This dataset consists of 60,000 color images of size 32×32 distributed across 10 balanced classes. CIFAR-10 is a standard benchmark in computer vision, widely used to evaluate image classification models under moderate complexity conditions. Its relatively low resolution and balanced class distribution make it ideal for testing both model performance and the effectiveness of privacy-preserving techniques.
- **ImageNet-10**[2]**:** A curated subset of the larger ImageNet dataset, ImageNet-10 contains images from 10 diverse classes. This subset is more challenging than CIFAR-10 due to its greater variability in image content, higher resolution, and increased intra-class diversity. It provides a rigorous testbed for evaluating the scalability and robustness of defense mechanisms on complex, real-world data.
- **SST-2**[3]**:** The Stanford Sentiment Treebank (SST-2) is a sentiment classification dataset extracted from the GLUE benchmark. It comprises text samples labeled with binary sentiment (positive or negative). SST-2 is representative of natural language processing tasks and allows us to assess the performance of privacy-preserving methods on models that process unstructured text data.

The target models employed in our experiments are selected to reflect the typical architectures used in their respective domains:

- **AlexNet [8]:** Originally developed for the ImageNet Large Scale Visual Recognition Challenge, AlexNet is a deep convolutional neural network consisting of five convolutional layers followed by three fully connected layers. In our experiments on CIFAR-10 and ImageNet-10, AlexNet serves as the target model. Its layered structure, ReLU activations, and use of dropout make it an effective and widely adopted baseline for evaluating both classification performance and membership inference defenses.
- **DistilBERT [16]:** It is a compact version of the BERT transformer model, designed to retain much of BERT's language understanding capabilities while reducing its size and computational cost. In our experiments on SST-2, DistilBERT serves as the target model, offering robust performance on text classification tasks with lower inference latency. Its efficiency makes it well-suited for integrating and testing privacy-preserving mechanisms in the context of Natural Language Processing (NLP).

7.4 Experimental Setup and Results

We conduct experiments on CIFAR-10, ImageNet-10, and SST-2 datasets. For each dataset, 70% of the data is used to train and test the target model, while the remaining 30% is allocated for training shadow and attack models. All models are trained for 15 epochs using stochastic gradient descent (SGD) with a learning rate of 0.01 and a batch size of 64. Experiments are conducted on a machine equipped with NVIDIA RTX 5000 GPU to ensure efficient training and evaluation.

As our experimental baseline defense, we implement SELENA [21] as described in Section 2.2, using $K = 25$ sub-models and $L = 10$ partitions per sample. Each target model is evaluated against three types of membership inference attacks: Confidence Threshold Attack, Loss Threshold Attack, and Shadow Model Attack. For each attack, we report the attack success rate (ASR) under three conditions: (i) without any defense (None), (ii) after applying the SELENA defense mechanism, and (iii) after applying our proposed adaptive noise injection method (DynaNoise).

To quantify the privacy–utility trade-off, we utilize our proposed metric called *MIDPUT*, as described in section 7.1. This metric captures the extent to which a defense reduces attack success rate while preserving model accuracy. We report both the overall MIDPUT and per-attack values (MIDPUT_C, MIDPUT_L, and MIDPUT_S) for each defense mechanism.

Table 3 summarizes the membership inference metrics and the corresponding MIDPUT values for each dataset and defense mechanism. For CIFAR-10, the baseline (no defense) model achieves a test accuracy of 0.8211, with ASR values of 0.6956 (Confidence), 0.7639 (Loss), and 0.7841 (Shadow). SELENA reduces these to 0.7668, 0.5446, 0.6307, and 0.6755, respectively, resulting in an overall MIDPUT of 0.0766. In contrast, DynaNoise maintains a high test accuracy of 0.8156 while significantly lowering the attack success rates to 0.2785 (Confidence), 0.4221 (Loss), and 0.5334 (Shadow). Consequently, the overall MIDPUT for DynaNoise is 0.331, with per-attack MIDPUT values of 0.4116, 0.3363, and 0.2452, respectively. This shows that DynaNoise clearly outperforms SELENA in reducing MIA success rate while preserving model accuracy on CIFAR-10.

A similar trend is observed on ImageNet-10: while the baseline test accuracy is 0.9165, SELENA degrades the accuracy significantly to 0.7085 and produces very low (even negative) MIDPUT values (overall MIDPUT of -0.0342). Conversely, DynaNoise preserves the test accuracy at 0.9115, reduces the attack metrics to 0.2248 (Confidence), 0.308 (Loss), and 0.5725 (Shadow), and achieves an overall MIDPUT of 0.3358. These results highlight that DynaNoise not only provides stronger privacy protection than SELENA but also avoids the substantial utility drop seen with SELENA on ImageNet-10.

On SST-2, The baseline DistilBERT model reaches a test accuracy of 0.8865. SELENA slightly increases the accuracy to 0.8945 but yields only marginal improvements in the attack metrics, resulting in an overall MIDPUT of 0.0392. In contrast, DynaNoise achieves a test accuracy of 0.8911 while significantly reducing the confidence attack ASR to 0.2354 and achieving a slightly better reduction in the shadow attack ASR, leading to an overall MIDPUT of 0.2091. Despite the already high baseline accuracy, DynaNoise achieves a more favorable trade-off compared to SELENA, offering meaningful privacy gains in the SST-2 NLP setting.

Figures 3a, 3b, and 3c illustrate how DynaNoise responds to variations in base variance, lambda scale, and temperature on CIFAR-10. Increasing the base variance or lambda scale introduces more noise into high-risk predictions, which leads to a steady decline in membership inference attack success rates. Meanwhile, test accuracy remains relatively stable, indicating that low-sensitivity predictions

[1] https://www.tensorflow.org/datasets/catalog/cifar10
[2] https://www.kaggle.com/datasets/liusha249/imagenet10/code
[3] https://huggingface.co/datasets/gimmaru/glue-sst2

Table 3: Model accuracy and MIA attack success rates along with the proposed MIDPUT metrics on different datasets under various defense mechanisms.

Dataset	Defense	Model (↑)	Confidence (↓)	Loss (↓)	Shadow (↓)	MIDPUT$_C$ (↑)	MIDPUT$_L$ (↑)	MIDPUT$_S$ (↑)	MIDPUT$_{Overall}$ (↑)
CIFAR10	None	0.8211	0.6956	0.7639	0.7841	–	–	–	–
	SELENA [21]	0.7668	0.5446	0.6307	0.6755	0.0967	0.0789	0.0543	0.0766
	DynaNoise	**0.8156**	**0.2785**	**0.4221**	**0.5334**	**0.4116**	**0.3363**	**0.2452**	**0.331**
ImageNet-10	None	0.9165	0.6489	0.7177	0.7612	–	–	–	–
	SELENA [21]	0.7085	0.3893	0.5355	0.6817	0.0516	-0.0258	-0.1285	-0.0342
	DynaNoise	**0.9115**	**0.2248**	**0.308**	**0.5725**	**0.4191**	**0.4047**	**0.1837**	**0.3358**
SST-2	None	0.8865	0.7913	0.7961	0.7854	–	–	–	–
	SELENA [21]	**0.8945**	0.7774	**0.7603**	0.7415	0.0219	**0.0438**	0.0519	0.0392
	DynaNoise	0.8911	**0.2354**	0.7863	**0.735**	**0.5605**	0.0144	**0.0523**	**0.2091**

are minimally affected. Temperature tuning further smooths the output distributions, effectively masking membership signals while preserving confidence in correct predictions. These observations validate the adaptive nature of DynaNoise in balancing noise and utility.

Figures 4a, 4b, and 4c show similar dynamics for ImageNet-10. As the base variance and lambda scale increase, noise adapts to prediction certainty, which selectively reduces overconfident outputs and lowers attack success. Adjusting the temperature parameter introduces a softening effect on logits, which initially helps reduce the distinguishability between member and non-member outputs. However, beyond a certain point, further increases in temperature yield diminishing returns, as the shadow attack success rate begins to rise again. This suggests that moderate temperature values are optimal for balancing privacy protection and output utility.

Figures 5a, 5b, and 5c display results for SST-2, where the target model already achieves high accuracy. Even with minimal impact on accuracy, tuning the defense parameters consistently lowers attack success rates, particularly with increased temperature. These results highlight that DynaNoise can still offer meaningful privacy gains in settings where the model is already well-optimized and has limited room for further improvement.

Moreover, across all datasets, as shown in Figure 3c, 4c, and 5c, we observe that varying the temperature parameter has little to no effect on the test accuracy of the target model after DynaNoise is applied. This behavior stems from the role of temperature in post-processing the logits. To be more precise, it scales the output distribution without altering the predicted class. Since temperature is applied within the softmax function and does not change the relative ordering of logits in most cases, the top-1 prediction remains unaffected. Consequently, while higher temperatures effectively smooth the output probabilities and making it harder for an attacker to distinguish between member and non-member samples, they do so without degrading model performance. This characteristic further demonstrates the utility-preserving advantage of DynaNoise.

Overall, the figures confirm that DynaNoise's parameterization enables fine-grained control over the privacy-utility trade-off, leveraging prediction sensitivity to selectively apply membership obfuscation where it matters the most.

7.5 Time and Computational Overhead Analysis

The evaluated defense approaches differ not only in their per-sample time complexity but also in the overall computational resources required. DynaNoise perturbs the model's logits by adding Gaussian noise whose variance is scaled based on query sensitivity, and then re-computes the ℓ-dimensional probability vector using a softmax operation. This results in a per-sample computational overhead of

$$O(\ell),$$

which corresponds to a single additional noise sampling and softmax computation. In practice, this represents a negligible runtime cost, making DynaNoise a lightweight and scalable defense. In contrast, SELENA employs an ensemble-based defense by training K sub-models using the Split-AI architecture and subsequently applying self-distillation to consolidate their knowledge into a single model. During training, SELENA incurs a computational cost of

$$O(K \cdot C_{\text{model}}),$$

where C_{model} denotes the cost of training a single target model. While the self-distilled model reduces the inference cost to that of a standard target model, the training phase remains substantially more resource-intensive and time-consuming.

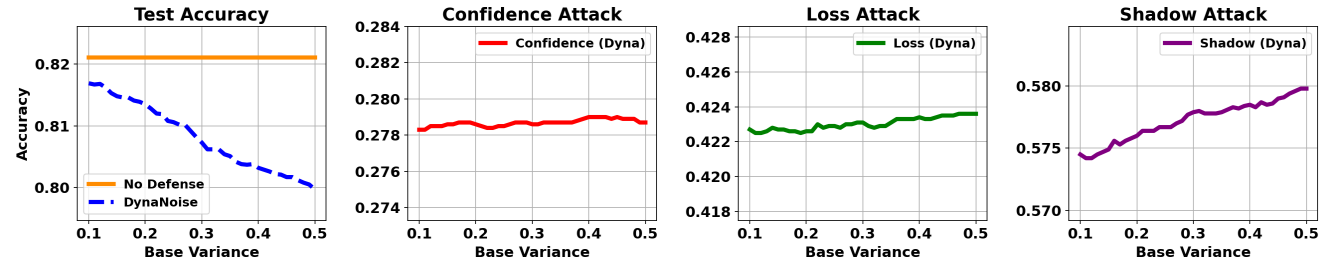
Beyond per-sample time complexity, the overall resource requirements are also critical. DynaNoise relies on a lightweight post-hoc noise injection approach, resulting in a memory footprint that remains nearly identical to that of the base model. In contrast, SELENA requires training and maintaining an ensemble of K sub-models, which, even when parallelized, demands significantly more GPU memory and computational resources. As a result, DynaNoise provides a more efficient and scalable defense with substantially lower overall overhead.

7.6 Discussions and Limitations

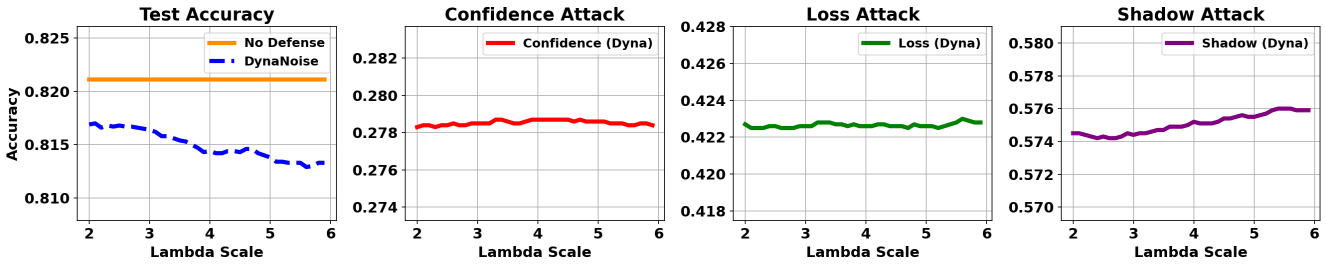
Below we list the advantages and limitations of our proposed DynaNoise approach compared to SELENA and also in general.

Advantages:

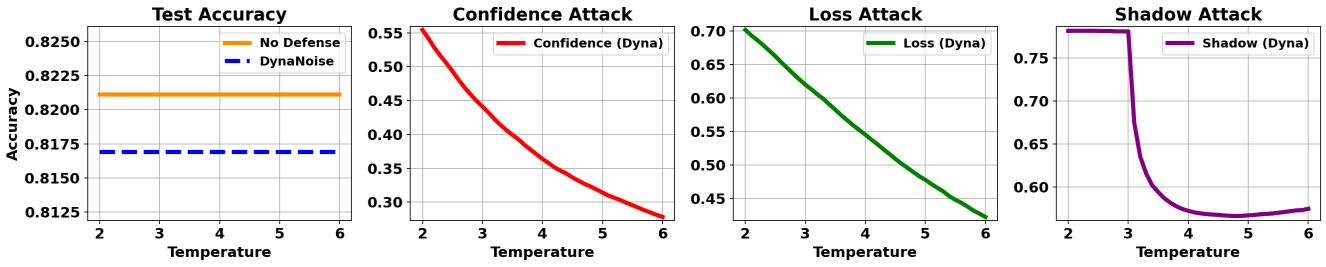
- **Adaptive Noise Injection:** DynaNoise dynamically adjusts the noise level based on query sensitivity. This adaptive mechanism enables DynaNoise to effectively mask membership signals while preserving high model accuracy.
- **Low Computational Overhead:** Unlike SELENA, which requires training and maintaining an ensemble of sub-models,



(a) Base Variance Comparison



(b) Lambda Scale Comparison



(c) Temperature Comparison

Figure 3: CIFAR10: Accuracy and ASR comparison over varying (a) Base Variance, (b) Lambda Scale, and (c) Temperature.

DynaNoise incurs only a minimal additional cost. Its lightweight post-hoc noise injection results in a resource footprint very similar to that of the target model.

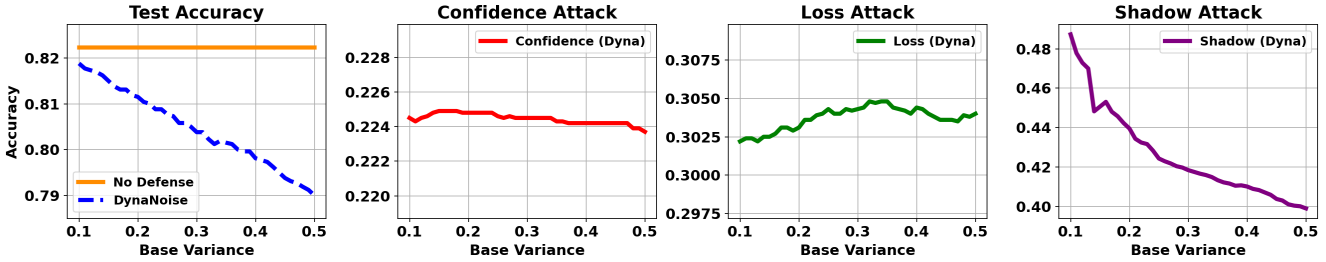
- **Robust Membership Inference Defense Privacy–Utility Trade-off (MIDPUT):** Our experiments demonstrate that DynaNoise consistently achieves higher MIDPUT values compared to SELENA. This indicates that the defense substantially reduces attack success rates with only a minor impact on model accuracy.
- **Simplicity and Scalability:** DynaNoise's implementation is straightforward and easily scalable to various datasets and model architectures without the complexities associated with ensemble training in SELENA.

Limitations:

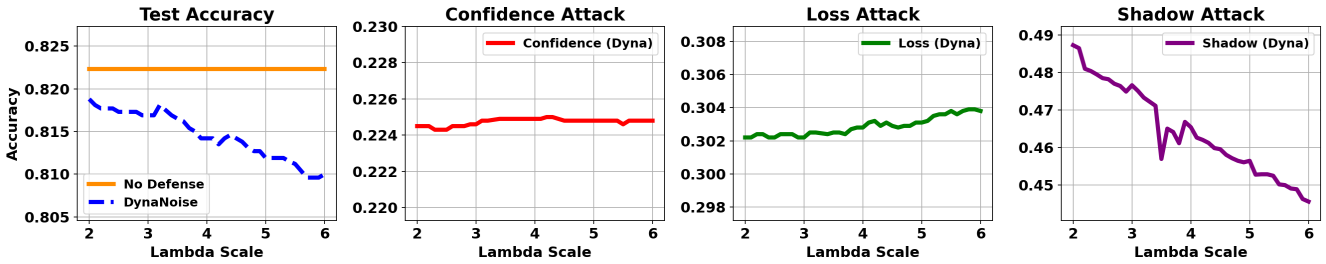
- **Hyperparameter Sensitivity:** Similar to SELENA, the performance of DynaNoise depends on the tuning of key parameters (base variance, lambda scale, and temperature). Although extensive experiments have allowed us to identify robust parameter ranges, some degree of manual tuning remains necessary to optimize performance across different settings.
- **No Formal Privacy Guarantees:** DynaNoise, like many practical defenses including SELENA, does not provide formal mathematical privacy guarantees. However, it is designed to offer strong empirical protection against membership inference attacks, as validated across diverse datasets, model architectures, and attack types.

8 Conclusion

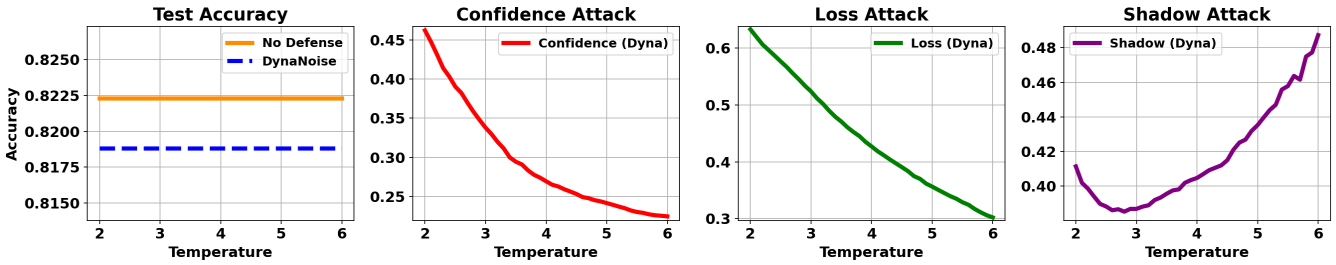
In this paper, we proposed DynaNoise, an adaptive noise injection approach that dynamically adjusts the level of noise added to a model's output based on the sensitivity of each query. In addition, we introduced a practical evaluation metric, called *Membership Inference Defense Privacy–Utility Trade-off (MIDPUT)*, which measures how effectively a defense balances membership privacy gains with



(a) Base Variance Comparison



(b) Lambda Scale Comparison



(c) Temperature Comparison

Figure 4: ImageNet-10: Accuracy and ASR comparison over varying (a) Base Variance, (b) Lambda Scale, and (c) Temperature.

the associated cost in target model accuracy. Our extensive experimental analysis shows that DynaNoise consistently outperforms SELENA as the state-of-the-art baseline MIA defense approach. Unlike SELENA, which relies on an ensemble approach and incurs high computational overhead, DynaNoise operates with a lightweight post-hoc noise injection mechanism that demands minimal additional resources. This efficiency makes our approach scalable and well-suited for resource-constrained environments, while the adaptive noise modulation leads to a significant reduction in membership inference success rates. Future work will focus on applying such adaptive noise injection during the training phase. Additionally, we plan to explore the integration of DynaNoise with formal differential privacy techniques to provide stronger, provable privacy guarantees.

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. 308–318.

[2] Soumia Zohra El Mestari, Gabriele Lenzini, and Huseyin Demirci. 2024. Preserving data privacy in machine learning systems. *Computers & Security* 137 (2024), 103605.

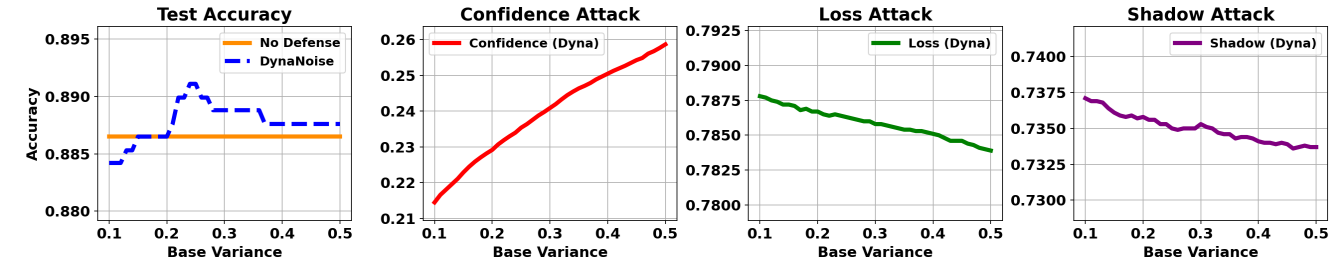
[3] Javad Forough, Monowar Bhuyan, and Erik Elmroth. 2022. DELA: A Deep Ensemble Learning Approach for Cross-layer VSI-DDoS Detection on the Edge. In *2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 1155–1165.

[4] Javad Forough, Monowar Bhuyan, and Erik Elmroth. 2023. Anomaly detection and resolution on the edge: Solutions and future directions. In *2023 IEEE International Conference on Service-Oriented System Engineering (SOSE)*. IEEE, 227–238.

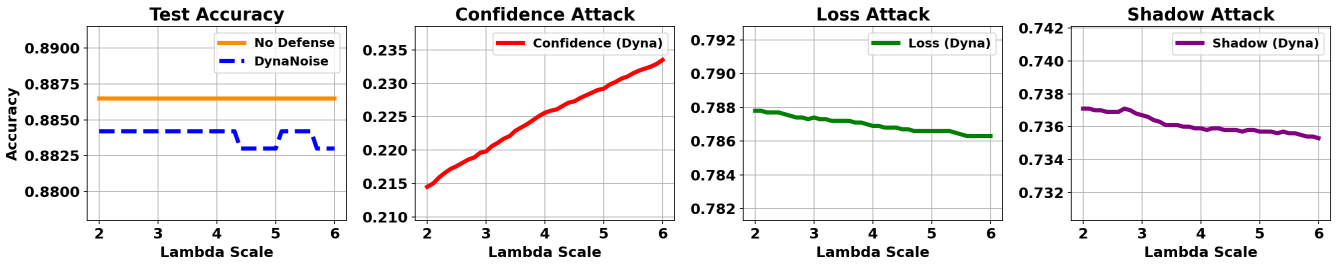
[5] Saidul Islam, Hanae Elmekki, Ahmed Elsebai, Jamal Bentahar, Nagat Drawel, Gaith Rjoub, and Witold Pedrycz. 2024. A comprehensive survey on applications of transformers for deep learning tasks. *Expert Systems with Applications* 241 (2024), 122666.

[6] Bargav Jayaraman and David Evans. 2019. Evaluating differentially private machine learning in practice. In *28th USENIX Security Symposium (USENIX Security 19)*. 1895–1912.

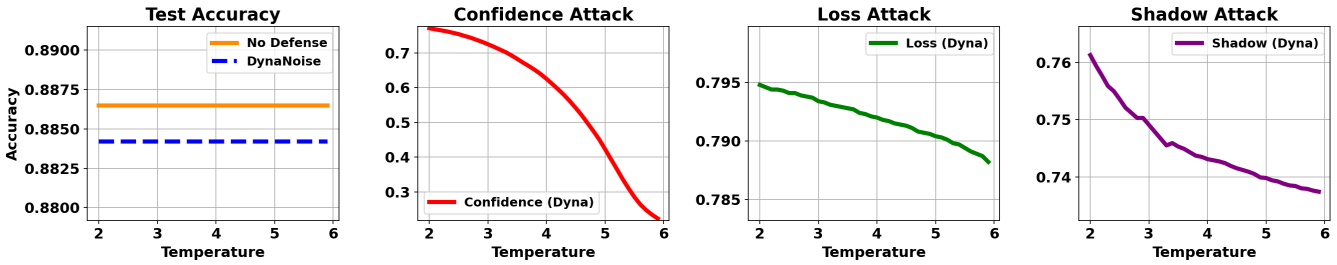
[7] Jinyuan Jia, Ahmed Salem, Michael Backes, Yang Zhang, and Neil Zhenqiang Gong. 2019. Memguard: Defending against black-box membership inference attacks via adversarial examples. In *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*. 259–274.



(a) Base Variance Comparison



(b) Lambda Scale Comparison



(c) Temperature Comparison

Figure 5: SST-2: Accuracy and ASR comparison over varying (a) Base Variance, (b) Lambda Scale, and (c) Temperature.

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* 25 (2012).

[9] Yunhui Long, Lei Wang, Diyue Bu, Vincent Bindschaedler, Xiaofeng Wang, Haixu Tang, Carl A Gunter, and Kai Chen. 2020. A pragmatic approach to membership inferences on machine learning models. In *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 521–534.

[10] Ismail A Mageed and Qichun Zhang. 2022. An introductory survey of entropy applications to information theory, queuing theory, engineering, computer science, and statistical mechanics. In *2022 27th international conference on automation and computing (ICAC)*. IEEE, 1–6.

[11] Milad Nasr, Reza Shokri, and Amir Houmansadr. 2018. Machine learning with membership privacy using adversarial regularization. In *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*. 634–646.

[12] Ke Pan, Yew-Soon Ong, Maoguo Gong, Hui Li, A Kai Qin, and Yuan Gao. 2024. Differential privacy in deep learning: A literature survey. *Neurocomputing* (2024), 127663.

[13] Md Eshmam Rayed, SM Sajibul Islam, Sadia Islam Niha, Jamin Rahman Jim, Md Mohsin Kabir, and MF Mridha. 2024. Deep learning for medical image segmentation: State-of-the-art advancements and challenges. *Informatics in Medicine Unlocked* (2024), 101504.

[14] Santosh Kumar Sahu, Anil Mokhade, and Neeraj Dhanraj Bokde. 2023. An overview of machine learning, deep learning, and reinforcement learning-based techniques in quantitative finance: recent progress and challenges. *Applied Sciences* 13, 3 (2023), 1956.

[15] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. 2018. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. *arXiv preprint arXiv:1806.01246* (2018).

[16] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108* (2019).

[17] Avi Schwarzschild, Zhili Feng, Pratyush Maini, Zachary Lipton, and J Zico Kolter. 2024. Rethinking llm memorization through the lens of adversarial compression. *Advances in Neural Information Processing Systems* 37 (2024), 56244–56267.

[18] Salomé A Sepúlveda-Fontaine and José M Amigó. 2024. Applications of Entropy in Data Analysis and Machine Learning: A Review. *Entropy* 26, 12 (2024), 1126.

[19] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*. IEEE, 3–18.

[20] Liwei Song and Prateek Mittal. 2021. Systematic evaluation of privacy risks of machine learning models. In *30th USENIX Security Symposium (USENIX Security 21)*. 2615–2632.

[21] Xinyu Tang, Saeed Mahloujifar, Liwei Song, Virat Shejwalkar, Milad Nasr, Amir Houmansadr, and Prateek Mittal. 2022. Mitigating membership inference attacks by {Self-Distillation} through a novel ensemble architecture. In *31st USENIX security symposium (USENIX security 22)*. 1433–1450.

[22] Iswarya Kannoth Veetil, Divi Eswar Chowdary, Paleti Nikhil Chowdary, V Sowmya, and EA Gopalakrishnan. 2024. An analysis of data leakage and generalizability in MRI based classification of Parkinson's Disease using explainable 2D Convolutional Neural Networks. *Digital Signal Processing* 147 (2024), 104407.

[23] Chugui Xu, Ju Ren, Deyu Zhang, Yaoxue Zhang, Zhan Qin, and Kui Ren. 2019. GANobfuscator: Mitigating information leakage under GAN via differential privacy. *IEEE Transactions on Information Forensics and Security* 14, 9 (2019), 2358–2371.

[24] Zhou Yang, Zhipeng Zhao, Chenyu Wang, Jieke Shi, Dongsun Kim, Donggyun Han, and David Lo. 2024. Unveiling memorization in code models. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*. 1–13.

[25] Jiayuan Ye. 2024. Privacy Analyses in Machine Learning. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*. 5110–5112.

[26] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. 2018. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*. IEEE, 268–282.

[27] Chaoyun Zhang, Paul Patras, and Hamed Haddadi. 2019. Deep learning in mobile and wireless networking: A survey. *IEEE Communications surveys & tutorials* 21, 3 (2019), 2224–2287.