STEALTHY SHIELD DEFENSE: A CONDITIONAL MUTUAL INFORMATION-BASED APPROACH AGAINST BLACK-BOX MODEL INVERSION ATTACKS

Anonymous authors
Paper under double-blind review

ABSTRACT

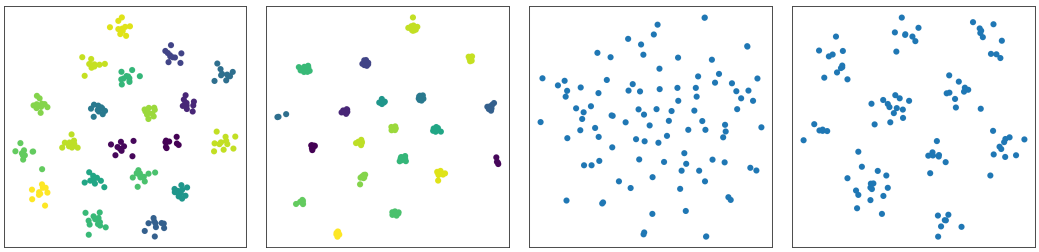
Model inversion attacks (MIAs) aim to reconstruct private training data by accessing public models, raising increasing concerns about privacy breaches. Black-box MIA, where attackers can generate inputs and obtain the model's outputs arbitrarily, has gained more attention due to its closer alignment with real-world scenarios and greater potential threats. Existing defenses primarily focus on white-box attacks, with a lack of specialized defenses to address the latest black-box attacks. To fill this technological gap, we propose a post-processing defense based on conditional mutual information (CMI). We have theoretically proven that our CMI framework serves as a special information bottleneck, making outputs less dependent on inputs and more dependent on true labels. To further reduce the modifications to outputs, we introduce an adaptive rate-distortion framework and optimize it with the water-filling method. Experimental results show that our approach outperforms existing defenses, in terms of both MIA robustness and model utility, across various attack algorithms, training datasets, and model architectures.

1 INTRODUCTION

Deep Neural Networks (DNNs) have witnessed remarkable advancements in recent years, driving significant breakthroughs across multiple applications, including face recognition (He et al., 2016) and personalized recommendations (Wu & Yan, 2017). Despite their efficacy, these powerful models are vulnerable to malicious attacks, particularly in the realm of privacy. A paramount concern is the Model Inversion Attack (MIA) (Fang et al., 2024), which exploits the output information of released models to reconstruct sensitive input features. Even in the absence of direct access to the original training data, the adversary still succeeds in recovering the private information. This capability poses a substantial threat, as it enables malicious actors to potentially replicate private features of confidential identities, thereby undermining both the privacy and security of the whole system.

According to the access ability to the target model, MIAs can be categorized into *white-box* and *black-box* attacks. black-box MIAs can deploy an attack without any access to the parameters or structure of target models, demonstrating more threats than white-box MIAs. However, most existing defenses against MIAs concentrate on the white-box attacks and apply various techniques to enhance the inner robustness for target models, which tend to lose efficacy when transferred into the black-box MIAs. Moreover, the black-box MIAs are hard to be mitigated through improvements on the inner structure of target models as they never rely on the intermediate information available in the white-box settings. In fact, the experiments in this paper have shown that existing defense methods can no longer withstand the latest and most powerful black-box attacks. Specialized black-box defenses are necessary and urgently needed.

To fill this gap, we propose **S**tealthy **S**hield **D**efense (SSD), a black-box model inversion defense method based on conditional mutual information (CMI). The CMI measures the dependence between the model's input and output when the ground truth is given. Our approach involves modifying the model's output to reduce this dependence, effectively protecting against model inversion attacks. We further show that minimizing CMI aligns with the information bottleneck principle, highlighting its potential to balance data privacy and utility.



(a) CMI=0.21 **w/o Def.** (b) CMI=0.14 **with Def.** (c) CMI=4.02 **w/o Def.** (d) CMI=1.96 **with Def.**

Figure 1: Visualize our defense effects via CMI. The detailed settings are in Sec 5.3.

Through empirical verification, we demonstrate that CMI has an intuitive geometric interpretation, which enhance model robustnest against MIAs. Figure 1(a)1(b) shows the probability vectors predicted by target model on test samples, with different colors indicating different classes. CMI quantifies the intra-class dispersion of these vectors, and a smaller CMI implies that they are more concentrated. With our defense, CMI is reduced to prevent attackers from obtaining more information about the private data distribution. Figure 1(c)1(d) displays the ones on samples generated by attackers. With our defense, these probability vectors are clustered into groups, misleading attackers' judgment on the classes their samples belong to.

In summary, the contributions of this paper are:

- Based on a review of existing attacks and defenses, we point out that the dependence between inputs and outputs should be minimized at the class level to defend against MIA. To achieve this, we introduce CMI into model inversion defense.
- We propose a post-processing algorithm to minimize CMI without retraining models. In our algorithm, we introduce a temperature mechanism to calibrate the probabilities and propose an adaptive rate-distortion mechanism to reduce the modifications to outputs. We speed up our algorithm by water-filling as well.
- Our experiments demonstrate that our defense outperforms all competitors in terms of MIA robustness and model utility, exhibiting good generalizability across various attack algorithms, private training datasets, and target model architectures.

2 Related Works

2.1 Model Inversion Attacks

Model inversion (MI) attacks are used to extracting training data from a trained model. Typically, MI attacks are categorized into two scenarios: white-box and black-box attacks. In white-box scenarios, attackers attackers have full access to the model's parameters and outputs. However, in black-box scenarios, the attackers can only observe the model's outputs.

Zhang et al. (2020) first propose a generative model inversion attack (GMI) to effectly attack deep neural networks (DNNs). The attackers train a GAN model to capture the similar structure of the private data. In the attack process, the attackers search the latent space of the GAN generator and minimize the classification loss to generate images that close to the private samples.

In this paper, we focus on the black-box scenarios. Recent black-box attacks show strong threaten against models, leading to a huge risk of privacy leakage. BREP (Kahla et al., 2022) utilizes zero-order optimization to urge the latent vector to gradually move away from the decision boundary. Mirror (An et al., 2022) and C2F (Ye et al., 2023) explores gradient-free techniques. They execute the optimization process with the genetic algorithm. LOKT (Nguyen et al., 2024) is the SOTA black-box method. It transfers the black-box attacks into white-box. They train multiple surrogate models and apply white-box attacks on them.

2.2 Mutual Information in Deep Learning

Tishby et al. (2001) established the information bottleneck (IB) theory for deep learning (DL), explaining the forward propagation from the perspective of mutual information (MI). However, the complex computation of MI restricts the application of IB. To overcome this, Alemi et al. (2017) proposed a variational method for calculating the MI in DL, and Kolchinsky et al. (2017) improved their method. Belghazi et al. (2018) proposed MINE to compute the MI in high-dimensional continuous space, and Hu et al. (2024) proposed InfoNet to reduce the time overhead of MINE. Shwartz-Ziv & Tishby (2017) pointed out that MI and IB may be the key to achieving explainable DL.

Yang et al. (2024) proposed a deep learning framework constrained by conditional mutual information (CMI). They also use CMI for knowledge distillation (Ye et al., 2024) and federated learning (Hamidi et al., 2024). We will point out in Proposition 1 that CMI is a special case of IB.

3 Preliminary

3.1 Notations

Let $f\colon \mathbb{X} \to \mathbb{Y}$ be a neural classifier, $\boldsymbol{X} \in \mathbb{X}$ be an input to f, $Y \in \mathbb{Y}$ be its ground truth label, $\hat{Y} \in \mathbb{Y}$ be the label predicted by f, and $\boldsymbol{Z} \in \mathbb{Z}$ be the intermediate feature in f. $\boldsymbol{X}, Y, \hat{Y}, \boldsymbol{Z}$ are random variables and let $\boldsymbol{x}, y, \hat{y}, \boldsymbol{z}$ donate their values, resp. For brevity, let $\mathbb{P}(\boldsymbol{x}) \coloneqq \mathbb{P}\{\boldsymbol{X} = \boldsymbol{x}\}$, $\mathbb{P}(y) \coloneqq \mathbb{P}\{Y = y\}$, $\mathbb{P}(\boldsymbol{x}, \hat{y}|y) \coloneqq \mathbb{P}\{\boldsymbol{X} = \boldsymbol{x}, \hat{Y} = \hat{y}|Y = y\}$, etc.

Let $\mathbb{P}(\mathbb{Y})$ be the set of probability vectors over $\mathbb{Y}$. When $\boldsymbol{x}$ is input to f, let $\boldsymbol{f}(\boldsymbol{x}) \in \mathbb{P}(\mathbb{Y})$ be the output from the softmax layer, and $f_{\hat{y}}(\boldsymbol{x})$ be its $\hat{y}$-th component. Note $f(\boldsymbol{x}) = \arg\max_{\hat{y} \in \mathbb{Y}} f_{\hat{y}}(\boldsymbol{x})$.

3.2 Model Inversion Attacks

Model inversion attacks (MIAs) aim to reconstruct the private dataset by accessing the public model. Formally, let $D_{\text{train}} \subseteq \mathbb{X} \times \mathbb{Y}$ be the training set for f. D_{train} is secret and attackers aim to construct $\hat{D}_{\text{train}}$ as close to D_{train} as possible. Based on their access to f, MIAs are categorized as:

Hard-label: Attackers can get $f(\boldsymbol{x}) \in \mathbb{Y}$ for any $\boldsymbol{x} \in \mathbb{X}$.

Soft-label: Attackers can get $\boldsymbol{f}(\boldsymbol{x}) \in \mathbb{P}(\mathbb{Y})$ for any $\boldsymbol{x} \in \mathbb{X}$.

White-box: Attackers can get the structure and weights of f, and thus the $\boldsymbol{z} \in \mathbb{Z}$ w.r.t. any $\boldsymbol{x} \in \mathbb{X}$.

Hard-label and soft-label, collectively called *black-box*,[1] are what we aim to defend against.

3.3 Mutual Information-Based Defense

Wang et al. (2021) proposed to defend against MIAs via mutual informations. Formally, the mutual information between $\boldsymbol{X}$ and $\hat{Y}$ is defined as

$$\mathbb{I}(\boldsymbol{X}; \hat{Y}) \coloneqq \sum_{\boldsymbol{x} \in \mathbb{X}} \sum_{\hat{y} \in \mathbb{Y}} \mathbb{P}(\boldsymbol{x}, \hat{y}) \log \frac{\mathbb{P}(\boldsymbol{x}, \hat{y})}{\mathbb{P}(\boldsymbol{x})\mathbb{P}(\hat{y})}. \tag{1}$$

$\mathbb{I}(\boldsymbol{X}; \hat{Y})$ quantifies the information of $\boldsymbol{X}$ carried by $\hat{Y}$. They reduced it to prevent attackers from knowing the private $\boldsymbol{X}$ by observing $\hat{Y}$. However, reducing it impairs the model's utility. Especially, $\mathbb{I}(\boldsymbol{X}; \hat{Y}) = 0$ iff $\boldsymbol{X}$ and $\hat{Y}$ are independent. In that case, f is immune to any attack but useless at all.

As an alternative, they introduced *information bottlenecks* (Tishby et al., 2001), which is defined as

$$\mathbb{I}(\boldsymbol{X}; \boldsymbol{Z}) - \lambda \cdot \mathbb{I}(Y; \boldsymbol{Z}), \tag{2}$$

where $\lambda > 0$. They used (2) as a regularizer to train f, minimizing $\mathbb{I}(\boldsymbol{X}; \boldsymbol{Z})$ to defend against MIAs while maximizing $\mathbb{I}(Y; \boldsymbol{Z})$ to preserve the utility. They achieved a better trade-off by adjusting λ.

Peng et al. (2022) replaced the $\mathbb{I}$ in (2) with other dependence metrics, COCO (Gretton et al., 2005b) and HSIC (Gretton et al., 2005a), to avoid the complex calculations of mutual informations.

[1] Some literature refer to *hard-label* as *label-only*, and *soft-label* as *black-box*.

4 METHODOLOGY

4.1 CONDITIONAL MUTUAL INFORMATION-BASED DEFENSE

We aim to defend against black-box MIAs, so we still focus on $\hat{Y}$ rather than $\boldsymbol{Z}$. Furthermore, we have observed that all MIA algorithms target one fixed label during attacking. Formally, given $y \in \mathbb{Y}$, they aim to reconstruct $D^y_{\text{train}} \coloneqq \{\boldsymbol{x} \in \mathbb{X} \mid (\boldsymbol{x}, y) \in D_{\text{train}}\}$, constructing $\hat{D}^y_{\text{train}}$ as close to D^y_{train} as possible. Based on our observation, we propose to minimize the mutual information

$$\mathbb{I}(\boldsymbol{X}; \hat{Y}|Y = y) \coloneqq \sum_{\boldsymbol{x}\in\mathbb{X}} \sum_{\hat{y}\in\mathbb{Y}} \mathbb{P}(\boldsymbol{x}, \hat{y}|y) \log \frac{\mathbb{P}(\boldsymbol{x}, \hat{y}|y)}{\mathbb{P}(\boldsymbol{x}|y)\mathbb{P}(\hat{y}|y)}. \tag{3}$$

$\mathbb{I}(\boldsymbol{X}; \hat{Y}|Y = y)$ quantifies the information of $\boldsymbol{X}$ carried by $\hat{Y}$ when $Y = y$. We minimize it to prevent attackers from knowing the private $\boldsymbol{X}|Y = y$ by observing $\hat{Y}$, where $\boldsymbol{X}|Y = y$ denotes the input whose ground truth label is y.

Minimizing (3) on each $y \in \mathbb{Y}$ is equivalent to minimizing the conditional mutual information (CMI), which is defined as

$$\mathbb{I}(\boldsymbol{X}; \hat{Y}|Y) \coloneqq \sum_{y\in\mathbb{Y}} \mathbb{P}(y)\mathbb{I}(\boldsymbol{X}; \hat{Y}|Y = y). \tag{4}$$

Proposition 1. *CMI is a special case of information bottlenecks, taking $\boldsymbol{Z} = \hat{Y}$ and $\lambda = 1$, i.e.*

$$\mathbb{I}(\boldsymbol{X}; \hat{Y}|Y) = \mathbb{I}(\boldsymbol{X}; \hat{Y}) - \mathbb{I}(Y; \hat{Y}). \tag{5}$$

Our proof is provided in Appendix A. Our proposition shows that CMI inherits the benefits of information bottlenecks—minimizing $\mathbb{I}(\boldsymbol{X}; \hat{Y})$ to defend against MIAs while maximizing $\mathbb{I}(Y; \hat{Y})$ to preserve the model's utility.

The $\mathbb{I}(\boldsymbol{X}; \boldsymbol{Z})$ in (2) is challenging to calculate because the input space $\mathbb{X}$ and feature space $\mathbb{Z}$ are both high-dimensional. Previous works can only estimate its bound by variational methods (Alemi et al., 2017), and can not calculate its value directly. Fortunately, as a special case of information bottlenecks, our CMI can be calculated and minimized directly, as described in the next section.

4.2 POST-PROCESSING ALGORITHM TO MINIMIZE CMI

Without retraining models, we propose to minimize CMI by post-processing. The CMI can be calculated as follows:

$$\mathbb{I}(\boldsymbol{X}; \hat{Y}|Y) = \sum_{y\in\mathbb{Y}} \mathbb{P}(y) \sum_{\boldsymbol{x}\in\mathbb{X}} \sum_{\hat{y}\in\mathbb{Y}} \mathbb{P}(\boldsymbol{x}, \hat{y}|y) \log \frac{\mathbb{P}(\boldsymbol{x}, \hat{y}|y)}{\mathbb{P}(\boldsymbol{x}|y)\mathbb{P}(\hat{y}|y)} \tag{6}$$

$$= \sum_{y\in\mathbb{Y}} \sum_{\boldsymbol{x}\in\mathbb{X}} \sum_{\hat{y}\in\mathbb{Y}} \mathbb{P}(\boldsymbol{x})\mathbb{P}(y|\boldsymbol{x})\mathbb{P}(\hat{y}|y, \boldsymbol{x}) \log \frac{\mathbb{P}(\hat{y}|\boldsymbol{x}, y)}{\mathbb{P}(\hat{y}|y)} \tag{7}$$

$$= \sum_{\boldsymbol{x}\in\mathbb{X}} \mathbb{P}(\boldsymbol{x}) \sum_{y\in\mathbb{Y}} \mathbb{P}(y|\boldsymbol{x}) \sum_{\hat{y}\in\mathbb{Y}} \mathbb{P}(\hat{y}|\boldsymbol{x}) \log \frac{\mathbb{P}(\hat{y}|\boldsymbol{x})}{\mathbb{P}(\hat{y}|y)}, \tag{8}$$

where (6) is by definitions (3-4), (7) is by conditional probability, and (8) is by the Markov chain of $Y \to \boldsymbol{X} \to \hat{Y}$.

Based on (8), minimizing $\mathbb{I}(\boldsymbol{X}; \hat{Y}|Y)$ is equivalent to minimizing $\sum_{y\in\mathbb{Y}} \mathbb{P}(y|\boldsymbol{x}) \sum_{\hat{y}\in\mathbb{Y}} \mathbb{P}(\hat{y}|\boldsymbol{x}) \log \frac{\mathbb{P}(\hat{y}|\boldsymbol{x})}{\mathbb{P}(\hat{y}|y)}$ for any $\boldsymbol{x}$ inputted to f. However, this objective function is too complex for convex optimizer to minimize. For simplicity, we sample $y \in \mathbb{Y}$ with the probability of $\mathbb{P}(y|\boldsymbol{x})$ and minimize $\sum_{\hat{y}\in\mathbb{Y}} \mathbb{P}(\hat{y}|\boldsymbol{x}) \log \frac{\mathbb{P}(\hat{y}|\boldsymbol{x})}{\mathbb{P}(\hat{y}|y)}$ instead, which is equivalent to the original formula in terms of mathematical expectation.

Let $\boldsymbol{q}^y \in \mathbb{P}(\mathbb{Y})$ and its $\hat{y}$-th component $q^y_{\hat{y}} \coloneqq \mathbb{P}(\hat{y}|y)$, consider $f_{\hat{y}}(\boldsymbol{x}) = \mathbb{P}(\hat{y}|\boldsymbol{x})$, and then we have

$$\sum_{\hat{y}\in\mathbb{Y}} \mathbb{P}(\hat{y}|\boldsymbol{x}) \log \frac{\mathbb{P}(\hat{y}|\boldsymbol{x})}{\mathbb{P}(\hat{y}|y)} = \sum_{\hat{y}\in\mathbb{Y}} f_{\hat{y}}(\boldsymbol{x}) \log \frac{f_{\hat{y}}(\boldsymbol{x})}{q^y_{\hat{y}}} = \mathbb{KL}(\boldsymbol{f}(\boldsymbol{x})||\boldsymbol{q}^y), \tag{9}$$

where $\mathbb{KL}$ is the Kullback-Leibler divergence, a binary strictly convex function.

We can get $\boldsymbol{f}(\boldsymbol{x})$ from the softmax layer when $\boldsymbol{x}$ is inputted to f. To determine $\boldsymbol{q}^y$, we note that

$$\begin{aligned} q^y_{\hat{y}} &= \sum_{\boldsymbol{x}\in\mathbb{X}} \mathbb{P}(\boldsymbol{x},\hat{y}|y) = \sum_{\boldsymbol{x}\in\mathbb{X}} \mathbb{P}(\boldsymbol{x}|y)\mathbb{P}(\hat{y}|\boldsymbol{x},y) = \sum_{\boldsymbol{x}\in\mathbb{X}} \mathbb{P}(\boldsymbol{x}|y)\mathbb{P}(\hat{y}|\boldsymbol{x}) = \sum_{\boldsymbol{x}\in\mathbb{X}} \mathbb{P}(\boldsymbol{x}|y) f_{\hat{y}}(\boldsymbol{x}) \\ &= \mathbb{E}_{\boldsymbol{X}|Y=y}[f_{\hat{y}}(\boldsymbol{X})], \text{ and thus } \boldsymbol{q}^y = \mathbb{E}_{\boldsymbol{X}|Y=y}[\boldsymbol{f}(\boldsymbol{X})]. \end{aligned} \tag{10}$$

Based on (10), we can estimate $\boldsymbol{q}^y$ by private samples whose ground truth label is y. Specifically, we can find a sample $\boldsymbol{x}'$ that labeled y in the validation set, and let $\tilde{\boldsymbol{q}}^y \coloneqq \boldsymbol{f}(\boldsymbol{x}')$.

To determine the sampling probability $\mathbb{P}(y|\boldsymbol{x})$, a simple idea is to consider

$$\mathbb{P}(y|\boldsymbol{x}) = \mathbb{P}(\hat{y}|\boldsymbol{x}) = f_{\hat{y}}(\boldsymbol{x}) \text{ for } y = \hat{y} \in \mathbb{Y}. \tag{11}$$

But Guo et al. (2017) have demonstrated that (11) is inaccurate for modern neural networks. Inspired by their work, we introduce temperature mechanism to calibrate it.

When $\boldsymbol{x}$ is inputted to f, we minimize CMI by modifying the prediction $\boldsymbol{f}(\boldsymbol{x})$. Let $\boldsymbol{p} \in \mathbb{P}(\mathbb{Y})$ be the modified prediction, and our objective function is $\mathbb{KL}(\boldsymbol{p}||\boldsymbol{q}^y)$ based on the above derivation. To preserve the model's utility, we constrain $\|\boldsymbol{p} - \boldsymbol{f}(\boldsymbol{x})\|_1 \le \varepsilon$ where $\varepsilon > 0$ is the distortion bound.

In rate-distortion theory (Shannon, 1959), minimizing mutual information under bounded distortion constraint is for signal compression. If a signal has less information, it is easier to compress, and a stricter distortion bound can be applied. Regarding $\boldsymbol{f}(\boldsymbol{x})$ as a signal and $\boldsymbol{p}$ as the compressed signal, we can use normalized entropy to quantify the information in $\boldsymbol{f}(\boldsymbol{x})$, which is defined as

$$\bar{\mathbb{H}}(\boldsymbol{x}) \coloneqq -\frac{1}{\log|\mathbb{Y}|} \sum_{\hat{y}\in\mathbb{Y}} f_{\hat{y}}(\boldsymbol{x}) \log f_{\hat{y}}(\boldsymbol{x}). \tag{12}$$

Smaller $\bar{\mathbb{H}}(\boldsymbol{x})$ means less information in $\boldsymbol{f}(\boldsymbol{x})$, and smaller modification it need. Thus we propose the adaptive constraint $\|\boldsymbol{p} - \boldsymbol{f}(\boldsymbol{x})\|_1 \le \varepsilon \cdot \bar{\mathbb{H}}(\boldsymbol{x})$ to further control the distortion. Note that the old constraint $\|\boldsymbol{p} - \boldsymbol{f}(\boldsymbol{x})\|_1 \le \varepsilon$ still holds due to the property of $0 \le \bar{\mathbb{H}}(\boldsymbol{x}) \le 1$. This method is called adaptive rate distortion.

Our defense is summarized as Algorithm 1:

Algorithm 1: Our post-processing to minimize CMI.

Input: Original prediction $\boldsymbol{f}(\boldsymbol{x})$, target model f, validation set D_{valid}, distortion bound ε, temperature T.
Output: Modified prediction $\boldsymbol{p}$.

$y \leftarrow$ Sample in $\mathbb{Y}$ with the probability of softmax $\left(\frac{\boldsymbol{f}(\boldsymbol{x})}{T}\right)$

Find one $(\boldsymbol{x}', y) \in D_{\text{valid}}$ and $\tilde{\boldsymbol{q}}^y \leftarrow \boldsymbol{f}(\boldsymbol{x}')$

Calculate $\bar{\mathbb{H}}(\boldsymbol{x})$ by (12)

Solve the convex optimization problem and return the optimal solution $\boldsymbol{p}$:

$$\begin{aligned} &\min \mathbb{KL}(\boldsymbol{p}||\tilde{\boldsymbol{q}}^y), \\ &\text{s.t. } \|\boldsymbol{p} - \boldsymbol{f}(\boldsymbol{x})\|_1 \le \varepsilon \cdot \bar{\mathbb{H}}(\boldsymbol{x}), \\ &\quad \boldsymbol{p} \in \mathbb{P}(\mathbb{Y}). \end{aligned} \tag{13}$$

Note that the $\tilde{\boldsymbol{q}}^y$ can be calculated and stored in advance, making our algorithm not require f and D_{valid}.

5 EXPERIMENT

5.1 EXPERIMENT SETTINGS

Datasets. Following previous researches of model inversion attacks, we use the FaceScrub (Ng & Winkler, 2014) and CelebFaces Attributes (CelebA) (Liu et al., 2015) as private datasets. FaceScrub consists of 530 identities. CelebA contains 10177 identities and we only take 1000 identities with the most images (Kahla et al., 2022). To adapt to the classifiers, all images in the various datasets are cropped and resized to 64×64 pixels in our experiment.

Models. We train a variety of classifiers using the private datasets mentioned above, measuring robust model performance under various conditions. For target models, we employ VGG-16 (Simonyan & Zisserman, 2014) and IR-152 (He et al., 2016), both of which are trained with different defense methods. We select FaceNet (Cheng et al., 2017) as the evaluation model.

Model inversion attacks. In our experiments, we foucus on four state-of-the-art (SOTA) black-box model inversion attacks, including BREP (Kahla et al., 2022), Mirror (An et al., 2022), C2F (Ye et al., 2023) and LOKT (Nguyen et al., 2024).

Metrics. Following previous works (Struppek et al., 2024), we utilize multiple metrics to comprehensively evaluate the performance of our defense methods. To measure the model robustness and utility of each method, we consider the following metrics:

- **Attack Accuracy.** The metric is used to imitate a human to determine whether reconstructed images correspond to the target identity or not. Specifically, we employ an evaluation model trained on the same dataset as the target model to re-classify the reconstructed images. We then compute the top-1 and top-5 classification accuracies, denoted as $Acc@1$ and $Acc@5$, respectively.
- **Feature Distance.** The feature is extracted from the second-to-last layer of the model. This distance metric measures the average l_2 distance between the features of reconstructed images and the nearest private images. Consistent with previous research, we use both the evaluation model and a pre-trained FaceNet model Schroff et al. (2015) to generate the features. The corresponding feature distances are denoted as σ_{eval} and σ_{face}. A lower feature distance indicates a closer semantic similarity between the reconstructed images and private samples.
- **Test Accuracy.** The top-1 classification accuracy on the private test set. This metric is used to evaluate the utility of the target model with defense.
- **Predictive Bias.** This metric is used to quantify the modification to the predicted probability vectors by defense methods. For an identical input, we take the L_1 norm of the difference between the outputs with and without defense. Avg L_1 is the average norm over all private test samples, and Max L_1 is the largest one. Lower values of both suggest that the defense method causes less harm to the predicted probability vectors.

5.2 COMPARISON WITH PREVIOUS STATE-OF-THE-ART DEFENSES

In this section, we evaluate the robustness of our defense method by comparing it against an undefended model and prior state-of-the-art (SOTA) defense strategies, including MID (Wang et al., 2021), BiDO (Peng et al., 2022), LS (Struppek et al., 2024) and TL (Ho et al., 2024). We adhere to the official configurations for each defense method, and the corresponding hyperparameters are detailed in Appendix B.

We evaluate the model's robustness under various types of black-box model inversion attacks, including both soft-label and hard-label attacks. We conduct experiments on different target models and private datasets to demonstrate that our approach performs effectively across diverse scenarios.

For soft-label attacks, we compare our method with previous defense strategies under the Mirror and C2F attacks. The attack results are listed in Table 1. We can observe that our method achieves significant improvements over existing defense strategies, especially when the attack has a strong

performace. Specifically, under the Mirror attack against IR-152 trained on the FaceScrub dataset, our method reduces the attack accuracy from 52.3% to 19.4%, achieving a 3.6% greater reduction compared to the previous SOTA method TL. For C2F attacks against VGG16 models trained on the FaceScrub dataset, our method reduces the attack accuracy to approximately $1/9$ of that without defense, which is only a quarter of the accuracy achieved under the TL defense.

Table 1: Experiment results against soft-label attacks.

Model Dataset	Defense	Mirror				C2F			
		$\downarrow Acc@1$	$\downarrow Acc@5$	$\uparrow \delta_{eval}$	$\uparrow \delta_{face}$	$\downarrow Acc@1$	$\downarrow Acc@5$	$\uparrow \delta_{eval}$	$\uparrow \delta_{face}$
IR-152 CelebA	**No Def.**	10.0%	18.8%	2526	1.31	3.6%	8.0%	2521	1.36
	MID	9.0%	17.6%	2448	1.23	0.2%	0.4%	2382	1.56
	BiDO	4.8%	11.4%	**2758**	1.17	0.8%	3.8%	**2598**	1.31
	LS	3.2%	7.8%	2602	1.33	1.4%	4.2%	2536	1.39
	TL	6.6%	14.4%	2613	1.27	2.6%	7.0%	2528	1.37
	Ours	**1.2%**	**3.0%**	2527	**1.56**	**0%**	**0.3%**	2377	**1.67**
IR-152 FaceScrub	**No Def.**	52.4%	74.6%	1893	0.79	27.0%	49.8%	1952	0.98
	MID	43.6%	63.4%	2067	0.86	3.0%	9.6%	**2754**	1.44
	BiDO	27.6%	53.0%	2132	0.99	14.2%	24.4%	2242	1.20
	LS	33.4%	56.6%	2153	0.88	21.8%	46.8%	2022	1.02
	TL	23.0%	47.2%	2155	0.95	6.8%	16.8%	2191	1.23
	Ours	**19.4%**	**28.2%**	**2415**	**1.31**	**2.0%**	**6.4%**	2517	**1.49**
VGG-16 FaceScrub	**No Def.**	8.0%	15.0%	2577	0.78	23.8%	37.0%	2315	0.93
	MID	6.4%	12.2%	2627	0.79	18.4%	31.8%	2239	0.93
	BiDO	11.4%	21.0%	2530	0.79	10.6%	19.2%	2552	0.94
	LS	10.2%	18.4%	2526	0.75	17.0%	29.2%	2424	0.95
	TL	6.8%	12.0%	2624	**0.88**	10.4%	17.6%	2602	1.03
	Ours	**5.6%**	**10.6%**	**2665**	0.80	**8.8%**	**15.2%**	**2681**	**1.07**

In hard-label scenarios with BREP and LOKT attacks. We provided a quantitive results in Table 2. Note that LOKT is the SOTA black-box attack method. It demonstrates very high attack performance across various kinds of settings. While previous defenses only showed limited defensive capabilities, our approach almost completely defeats this attack. Especially in the attack against IR-152 with FaceScrub dataset, without any defense, LOKT showed an attack accuracy of up to 82.9%. However, our defense method reduce it to only 1.7%, making it almost impossible to launch a successful attack. Moreover, our defense largely enhance the feature distance σ_{face} from 0.66 to 1.53, which indicate that our defense method make the attack failed to capture the privacy characteristics.

Table 2: Experiment results against hard-label attacks.

Model Dataset	Defense	BREP				LOKT			
		$\downarrow Acc@1$	$\downarrow Acc@5$	$\uparrow \delta_{eval}$	$\uparrow \delta_{face}$	$\downarrow Acc@1$	$\downarrow Acc@5$	$\uparrow \delta_{eval}$	$\uparrow \delta_{face}$
IR-152 CelebA	**No Def.**	7.2%	24.4%	1654	0.95	51.6%	74.4%	1469	0.85
	MID	12.6%	28.8%	1973	1.28	29.8%	51.0%	1713	1.04
	BiDO	13.0%	30.6%	1670	1.03	48.4%	66.8%	1551	0.95
	LS	15.6%	40.0%	1584	0.97	52.0%	73.6%	1489	0.88
	TL	10.2%	27.2%	1643	1.05	56.4%	74.6%	1510	0.92
	Ours	**0.4%**	**1.6%**	**2362**	**1.61**	**0.2%**	**1.0%**	**2321**	**1.54**
IR-152 FaceScrub	**No Def.**	32.8%	56.6%	2161	1.00	83.0%	93.2%	1488	0.66
	MID	34.0%	51.0%	2178	1.06	54.0%	74.4%	1856	0.82
	BiDO	24.2%	39.4%	2235	1.07	59.8%	77.6%	1694	0.77
	LS	22.8%	45.8%	2384	1.07	60.0%	77.6%	1748	0.74
	TL	14.2%	27.2%	2353	1.15	62.6%	78.2%	1682	0.73
	Ours	**3.4%**	**7.0%**	**2622**	**1.51**	**1.8%**	**4.4%**	**2694**	**1.53**
VGG-16 FaceScrub	**No Def.**	33.6%	56.6%	2327	0.94	93.8%	98.0%	1359	0.57
	MID	37.4%	58.2%	2249	0.90	82.4%	92.8%	1526	0.60
	BiDO	30.4%	51.8%	2349	0.96	78.8%	87.4%	1567	0.63
	LS	29.6%	49.0%	2402	0.94	78.2%	88.6%	1573	0.65
	TL	29.0%	47.8%	2381	0.98	58.2%	74.0%	1771	0.71
	Ours	**9.8%**	**15.0%**	**2586**	**1.45**	**12.6%**	**21.4%**	**2370**	**1.18**

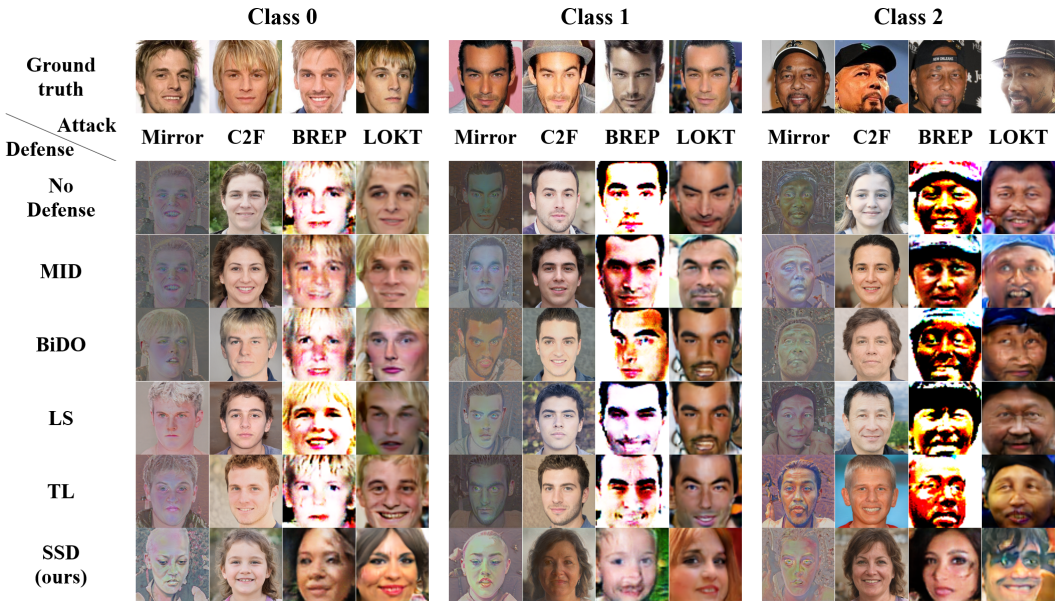


Figure 2: Visual comparison of reconstructed images using various black-box attack methods against an IR-152 model trained on CelebA, evaluated under different defense strategies. The top row displays the ground truth images of the target class from the private train dataset for reference.

Visualization results of the reconstructed images with different defense methods under different black-box attacks are shown in Fig. 2. Compared to previous approaches, our defense strategy produces reconstructed images that deviate more significantly from the private images, demonstrating its effectiveness in increasing the challenge for attackers to extract sensitive visual features and thereby enhancing privacy protection.

Table 3: Evaluation results on model's utility.

Defense	IR-152 & CelebA			IR-152 & FaceScrub			VGG-16 & FaceScrub		
	↑Acc	↓Avg L_1	↓Max L_1	↑Acc	↓Avg L_1	↓Max L_1	↑Acc	↓Avg L_1	↓Max L_1
No Def.	94.2%	0	0	98.6%	0	0	97.9%	0	0
MID	88.9%	0.44	1.93	96.5%	0.32	1.96	95.1%	0.36	1.78
BiDO	88.2%	0.37	1.96	94.0%	0.58	1.95	94.3%	0.27	1.90
LS	90.1%	0.37	1.99	94.9%	0.18	1.96	94.9%	0.19	1.88
TL	89.1%	0.35	1.84	95.3%	0.33	1.97	94.5%	0.15	1.96
Ours	**90.3%**	**0.15**	**0.95**	**96.7%**	**0.06**	**0.94**	**96.3%**	**0.05**	**0.74**

The evaluation results for the target model's utility are presented in Table 3. The results indicate that our defense holds the best utility, outperforming all competitors across different metrics, training datasets and model structures. Thanks to our bounded distortion constraint, our Max $L_1 \leq \varepsilon$ always holds strictly, where the competitors' are close to the maximum of 2. In particular, thanks to our adaptive rate-distortion, our Avg L_1 is only 1/5 to 1/2 of the competitors'.

5.3 Shaping Effects on Model Output with our SSD Defense Method

To see what modifications our Algorithm 1 makes to the outputs, we utilize t-SNE (van der Maaten & Hinton, 2008) method to visualize the predicted probability vectors from the VGG-16 trained on the FaceScrub dataset, and compute the Conditional Mutual Information (CMI) values by (Yang et al., 2024).

Private test samples that are correctly classified are shown in Figure 1(a)-1(b), with different colors indicating different labels. Without defense, these outputs are dispersed and their CMI is 0.21. With

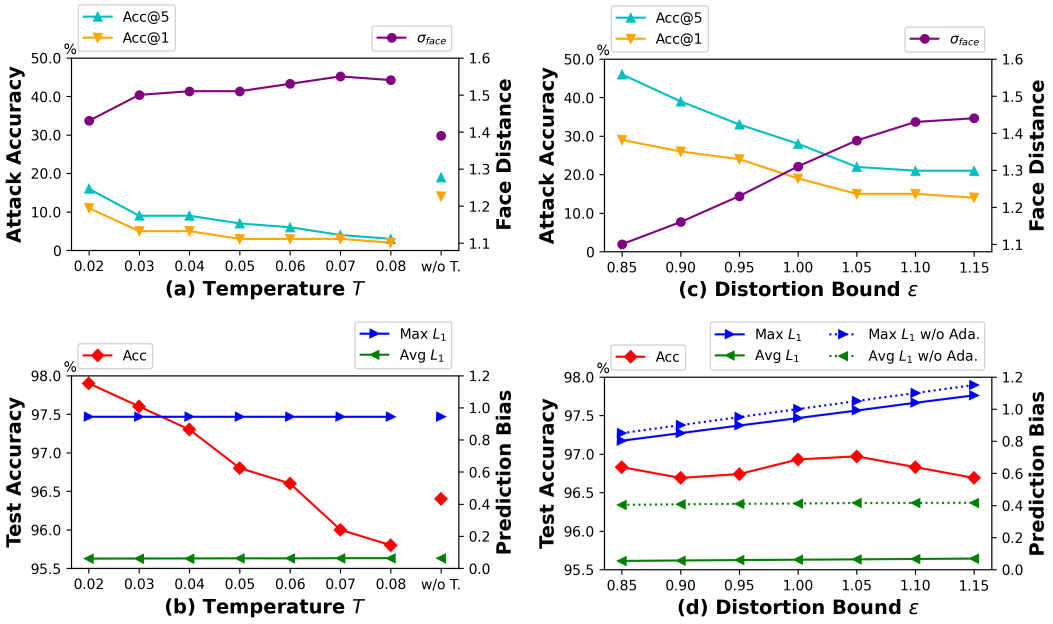


Figure 3: Ablation Study on temperature T and distortion bound ε.

our defense, these outputs are concentrated at their centers, and the CMI is reduced to 0.14. The latter reduces the dependence between inputs and outputs, preventing attackers from obtaining more information about the private data distribution. Meanwhile, we keep the labels of the outputs, which has little impact on the model's utility.

Figure 1(c)-1(d) shows the result on the samples generated by the attacker (selected from the initial phase of Mirror (An et al., 2022)). Without defense, these samples are uniformly distributed because the attacker does not know the training data distribution. With our defense, these samples are clustered into $|\mathbb{Y}| = 530$ groups (only 10 groups displayed for brevity). The CMI is reduced from 4.02 to 1.96 accordingly. Such grouping misleads the attacker's judgment on the classes their samples belong to, and makes it impossible for them to distinguish between private and non-private samples.

5.4 Ablation Studies

In this section, we conduct ablation experiments to explore the effects of the temperature and distortion bound in our defense. The target model is IR-152 trained on FaceScrub.

Figure 3 shows the experimental results on temperature, where the attack accuracy is measured on BREP. It can be seen that as the temperature T rises, our MIA robustness becomes stronger. This is because the y in Algorithm 1 is closer to uniform distribution, which makes it easier to return misleading labels to hard-label attackers. However, high temperature impairs the model's utility. In particular, the "NO" in Figure 3 represents the case without temperature mechanism. In that case, neither MIA robustness nor model's utility is ideal, which demonstrates the necessity of introducing a temperature mechanism.

For the distortion bound, the experiment results are displayed in Figure 3. The attack accuracy is measured on Mirror. As the distortion bound goes up, our defense can make more alterations to the output, resulting in better MIA robustness. It can be seen that relaxing the distortion bound mainly affects the maximum distortion Max L_1, while having almost no effect on the average distortion Avg L_1. Especially, without the adaptive mechanism, our Avg L_1 would become as high as that of other defenses. This demonstrates the necessity of introducing the adaptive mechanism.

6 CONCLUSION

In contrast to prior works that predominantly address white-box model inversion attacks, our study focuses specifically on defending against black-box attacks. We propose a novel defense strategy based on conditional mutual information (CMI) that operates entirely in the post-processing stage, eliminating the need for costly model retraining. By strategically modifying the model's outputs, our approach minimizes the dependency between inputs and outputs, thereby enhancing the model's resilience against inversion attacks. To further reduce output distortions, we incorporate an adaptive rate-distortion framework, optimized using a water-filling technique. Experimental results validate the effectiveness of our approach, achieving state-of-the-art performance in defending against black-box attacks. We believe that our findings will help shift attention towards robust defense mechanisms in black-box settings and inspire further research in this area.

REFERENCES

Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. In *International Conference on Learning Representations*, 2017.

Shengwei An, Guanhong Tao, Qiuling Xu, Yingqi Liu, Guangyu Shen, Yuan Yao, Jingwei Xu, and Xiangyu Zhang. Mirror: Model inversion for deep learning network with high fidelity. In *NDSS*, 2022.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

Yu Cheng, Jian Zhao, Zhecan Wang, Yan Xu, Karlekar Jayashree, Shengmei Shen, and Jiashi Feng. Know you at one glance: A compact vector representation for low-shot learning. In *ICCVW*, pp. 1924–1932, 2017.

Hao Fang, Yixiang Qiu, Hongyao Yu, Wenbo Yu, Jiawei Kong, Baoli Chong, Bin Chen, Xuan Wang, and Shu-Tao Xia. Privacy leakage on dnns: A survey of model inversion attacks and defenses. *ArXiv*, 2024.

Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Scholkopf. Measuring statistical dependence with hilbert-schmidt norms. In *International Conference on Algorithmic Learning Theory*, 2005a.

Arthur Gretton, Ralf Herbrich, Alex Smola, Olivier Bousquet, and Bernhard Scholkopf. Kernel methods for measuring independence. *J. Mach. Learn. Res.*, 2005b.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, 2017.

Shayan Mohajer Hamidi, Renhao Tan, Linfeng Ye, and En-Hui Yang. Fed-it: Addressing class imbalance in federated learning through an information- theoretic lens. *2024 IEEE International Symposium on Information Theory (ISIT)*, 2024.

Gyojin Han, Jaehyun Choi, Haeil Lee, and Junmo Kim. Reinforcement learning-based black-box model inversion attacks. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Sy-Tuyen Ho, Koh Jun Hao, Keshigeyan Chandrasegaran, Ngoc-Bao Nguyen, and Ngai-Man Cheung. Model inversion robustness: Can transfer learning help? *arXiv preprint arXiv:2405.05588*, 2024.

Zhengyang Hu, Song Kang, Qunsong Zeng, Kaibin Huang, and Yanchao Yang. InfoNet: Neural estimation of mutual information without test-time optimization. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Mostafa Kahla, Si Chen, Hoang Anh Just, and Ruoxi Jia. Label-only model inversion attacks via boundary repulsion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15045–15053, 2022.

Artemy Kolchinsky, Brendan D. Tracey, and David H. Wolpert. Nonlinear information bottleneck. *Entropy*, 2017.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pp. 3730–3738, 2015.

Hong-Wei Ng and Stefan Winkler. A data-driven approach to cleaning large face datasets. In *2014 IEEE international conference on image processing (ICIP)*, pp. 343–347, 2014.

Bao-Ngoc Nguyen, Keshigeyan Chandrasegaran, Milad Abdollahzadeh, and Ngai-Man Man Cheung. Label-only model inversion attacks via knowledge transfer. *Advances in Neural Information Processing Systems*, 36, 2024.

Xiong Peng, Feng Liu, Jingfeng Zhang, Long Lan, Junjie Ye, Tongliang Liu, and Bo Han. Bilateral dependency optimization: Defending against model-inversion attacks. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815–823, 2015.

Claude E. Shannon. *Coding Theorems for a Discrete Source With a Fidelity Criterion*. 1959.

Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *CoRR*, 2017.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Lukas Struppek, Dominik Hintersdorf, and Kristian Kersting. Be careful what you smooth for: Label smoothing can be a privacy shield but also a catalyst for model inversion attacks. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

Naftali Tishby, Fernando Pereira, and William Bialek. The information bottleneck method. *Allerton Conference on Communication, Control and Computation*, 2001.

Laurens van der Maaten and Geoffrey E. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, pp. 2579–2605, 2008.

Tianhao Wang, Yuheng Zhang, and Ruoxi Jia. Improving robustness to model inversion attacks via mutual information regularization. *AAAI Conference on Artificial Intelligence*, 2021.

Chen Wu and Ming Yan. Session-aware information embedding for e-commerce product recommendation. In *Proceedings of the 2017 ACM on conference on information and knowledge management*, pp. 2379–2382, 2017.

En-Hui Yang, Shayan Mohajer Hamidi, Linfeng Ye, Renhao Tan, and Beverly Yang. Conditional mutual information constrained deep learning: Framework and preliminary results. *IEEE International Symposium on Information Theory*, 2024.

Linfeng Ye, Shayan Mohajer Hamidi, Renhao Tan, and En-Hui Yang. Bayes conditional distribution estimation for knowledge distillation based on conditional mutual information. *ArXiv*, 2024.

Zipeng Ye, Wenjian Luo, Muhammad Luqman Naseem, Xiangkai Yang, Yuhui Shi, and Yan Jia. C2fmi: Corse-to-fine black-box model inversion attack. *IEEE Transactions on Dependable and Secure Computing*, 21(3):1437–1450, 2023.

Yuheng Zhang, Ruoxi Jia, Hengzhi Pei, Wenxiao Wang, Bo Li, and Dawn Song. The secret revealer: Generative model-inversion attacks against deep neural networks. In *CVPR*, 2020.

A PROOF OF PROPOSITION 1

Proof.

$$
\begin{aligned}
&\mathbb{I}(\boldsymbol{X};\hat{Y}|Y) \\
=&\sum_{y}\mathbb{P}(y)\sum_{\boldsymbol{x}\in\mathbb{X}}\sum_{\hat{y}\in\mathbb{Y}}\mathbb{P}(\boldsymbol{x},\hat{y}|y)\log\frac{\mathbb{P}(\boldsymbol{x},\hat{y}|y)}{\mathbb{P}(\boldsymbol{x}|y)\mathbb{P}(\hat{y}|y)} && \text{From definitions (3-4)} \\
=&\sum_{\boldsymbol{x}\in\mathbb{X}}\sum_{\hat{y}\in\mathbb{Y}}\sum_{y\in\mathbb{Y}}\mathbb{P}(\boldsymbol{x},\hat{y},y)\log\frac{\mathbb{P}(\hat{y}|\boldsymbol{x},y)}{\mathbb{P}(\hat{y}|y)} && \text{By conditional probability} \\
=&\sum_{\boldsymbol{x}\in\mathbb{X}}\sum_{\hat{y}\in\mathbb{Y}}\sum_{y\in\mathbb{Y}}\mathbb{P}(\boldsymbol{x},\hat{y},y)\log\frac{\mathbb{P}(\hat{y}|\boldsymbol{x})}{\mathbb{P}(\hat{y}|y)} && \text{By Markov chain: } Y\to\boldsymbol{X}\to\hat{Y} \\
=&\sum_{\boldsymbol{x}\in\mathbb{X}}\sum_{\hat{y}\in\mathbb{Y}}\sum_{y\in\mathbb{Y}}\mathbb{P}(\boldsymbol{x},\hat{y},y)\log\frac{\mathbb{P}(\boldsymbol{x},\hat{y})}{\mathbb{P}(\boldsymbol{x})}\frac{\mathbb{P}(y)}{\mathbb{P}(y,\hat{y})} && \text{By conditional probability} \\
=&\sum_{\boldsymbol{x}\in\mathbb{X}}\sum_{\hat{y}\in\mathbb{Y}}\mathbb{P}(\boldsymbol{x},\hat{y})\log\frac{\mathbb{P}(\boldsymbol{x},\hat{y})}{\mathbb{P}(\boldsymbol{x})\mathbb{P}(\hat{y})}-\sum_{y\in\mathbb{Y}}\sum_{\hat{y}\in\mathbb{Y}}\mathbb{P}(y,\hat{y})\log\frac{\mathbb{P}(y,\hat{y})}{\mathbb{P}(y)\mathbb{P}(\hat{y})} && \text{Toward definition (1)} \\
=&\mathbb{I}(\boldsymbol{X};\hat{Y})-\mathbb{I}(Y;\hat{Y}). && \text{Is special case of (2)}
\end{aligned}
$$

□

B SETTINGS FOR DEFENSES

Table 4: The parameter settings for all defenses.

Defense	IR-152 & CelebA	IR-152 & FaceScrub	VGG-16 & FaceScrub
MID	$\beta = 0.005$	$\beta = 0.01$	$\beta = 0.02$
BiDO	$\lambda_x = 0.001, \lambda_y = 0.01$	$\lambda_x = 0.01, \lambda_y = 0.1$	$\lambda_x = 0.002, \lambda_y = 0.02$
LS	$\alpha = -0.05$	$\alpha = -0.1$	$\alpha = -0.1$
TL	Freeze the first 50% of the layers.		
Ours	$T = 0.03, \varepsilon = 1$	$T = 0.05, \varepsilon = 1$	$T = 0.3, \varepsilon = 1$

C OUR WATER-FILLING ALGORITHM TO OPTIMIZE (13)

For brevity, we donate $\boldsymbol{q} \coloneqq \tilde{\boldsymbol{q}}^y$, $\boldsymbol{f} \coloneqq \boldsymbol{f}(\boldsymbol{x})$, and $\varepsilon \coloneqq \varepsilon \cdot \bar{\mathbb{H}}(\boldsymbol{x})$. The problem (13) is restated as

$$
\begin{aligned}
&\min \mathbb{KL}(\boldsymbol{p}||\boldsymbol{q}), \\
&\text{s.t. } \|\boldsymbol{p}-\boldsymbol{f}\|_1 \le \varepsilon, \\
&\quad \boldsymbol{p} \in \mathbb{P}(\mathbb{Y}).
\end{aligned} \tag{14}
$$

Note that Kullback-Leibler divergence is a metric. $\mathbb{KL}(\boldsymbol{p}||\boldsymbol{q}) \ge 0$ always holds and $\mathbb{KL}(\boldsymbol{p}||\boldsymbol{q}) = 0$ iff $\boldsymbol{p} = \boldsymbol{q}$. Trivially, when $\|\boldsymbol{q}-\boldsymbol{f}\|_1 \le \varepsilon$, the optimal solution is $\boldsymbol{p} = \boldsymbol{q}$.

When $\|\boldsymbol{q}-\boldsymbol{f}\|_1 > \varepsilon$, the optimal $\boldsymbol{p}$ must be between $\boldsymbol{f}$ and $\boldsymbol{q}$ due to the properties of $\mathbb{KL}$, i.e.

$$
\text{Either } f_i \le p_i \le q_i \text{ or } f_i \ge p_i \ge q_i, \text{ for each } i \in \mathbb{Y}. \tag{15}
$$

Furthermore, due to $\boldsymbol{f}, \boldsymbol{p} \in \mathbb{P}(\mathbb{Y})$, there must be

$$
\sum_{i\in\mathbb{Y}:f_i<q_i} p_i - f_i = \sum_{i\in\mathbb{Y}:f_i>q_i} f_i - p_i = \frac{\varepsilon}{2}. \tag{16}
$$

In the following we consider the case $f_i < q_i$ (another is symmetric). Assuming that $f_i < q_i$ iff $i \in \{1, 2, \ldots, n\}$, a semi-problem of (14) is

$$\begin{aligned} \min \ & \sum_{i=1}^{n} p_i \log \frac{p_i}{q_i}, \\ \text{s.t.} \ & \sum_{i=1}^{n} p_i - f_i = \frac{\varepsilon}{2}, \\ & p_i \geq f_i, \qquad i = 1, 2, \ldots, n. \end{aligned} \tag{17}$$

Introducing Lagrange multipliers $\boldsymbol{\lambda} \in \mathbb{R}^n_{\geq 0}$ and $v \in \mathbb{R}$, the KKT conditions are

$$(p_i - f_i)\lambda_i = 0, \tag{18}$$

$$1 + \log \frac{p_i}{q_i} - v - \lambda_i = 0, \tag{19}$$

where $i = 1, 2, \ldots, n$. Eliminating $\lambda_i \geq 0$ yields

$$(p_i - f_i)\left(1 + \log \frac{p_i}{q_i} - v\right) = 0, \tag{20}$$

$$1 + \log \frac{p_i}{q_i} \geq v. \tag{21}$$

When $v > 1 + \log \frac{f_i}{q_i}$, (21) implies $p_i > f_i$, and (20) implies $p_i = q_i \exp(v - 1)$.

When $v \leq 1 + \log \frac{f_i}{q_i}$, $p_i > f_i$ implies $\left(1 + \log \frac{p_i}{q_i} - v\right) > 0$ that against (20), so $p_i = f_i$.

In summary, the optimal solution is

$$p_i = \begin{cases} q_i \exp(v - 1) & v > 1 + \log \dfrac{f_i}{q_i} \\ f_i & \text{other} \end{cases}, \qquad i = 1, 2, \ldots, n, \tag{22}$$

where v is determined by the constraint $\sum_{i=1}^{n} p_i - f_i = \frac{\varepsilon}{2}$.

Let $w := \exp(v - 1) \in \mathbb{R}_{>0}$ and (22) is simplified to

$$p_i = \max(f_i, q_i w), \qquad i = 1, 2, \ldots, n. \tag{23}$$

We propose Algorithm 2 to calculate (23) efficiently. Our algorithm is known as "water-filling", because w is like a rising water level and $\frac{\varepsilon}{2}$ is like the maximum volume of water. Its time complexity is $O(n \log n)$ due to the sorting at the beginning.

Algorithm 2: Our water-filling on CPU.

Input: f_i, q_i for $i = 1, 2, \ldots, n$, and ε.
Output: p_i for $i = 1, 2, \ldots, n$.

Reindex f_i, q_i so that $\frac{f_1}{q_1} \leq \frac{f_2}{q_2} \leq \ldots \leq \frac{f_n}{q_n}$
$i \leftarrow 1$
$f_{\text{sum}} \leftarrow 0$
$q_{\text{sum}} \leftarrow 0$
while $q_{sum} \frac{f_i}{q_i} - f_{sum} < \frac{\varepsilon}{2}$ **do**
  $i \leftarrow i + 1$
  $f_{\text{sum}} \leftarrow f_{\text{sum}} + f_i$
  $q_{\text{sum}} \leftarrow q_{\text{sum}} + q_i$
end
$w \leftarrow \dfrac{f_{\text{sum}} + \frac{\varepsilon}{2}}{q_{\text{sum}}}$
Reindex f_i, q_i back to the original
return $\max(f_i, q_i w)$ for $i = 1, 2, \ldots, n$

Algorithm 3: Our water-filling on GPU.

Input: $\boldsymbol{f}, \boldsymbol{q}$ which are PyTorch tensors, and ε.
Output: $\boldsymbol{p}$ which is a PyTorch tensor.

Reindex $\boldsymbol{f}, \boldsymbol{q}$ by torch.sort($\frac{\boldsymbol{f}}{\boldsymbol{q}}$)

$\boldsymbol{f}_{\text{sum}} \leftarrow \boldsymbol{f}$.cumsum()
$\boldsymbol{q}_{\text{sum}} \leftarrow \boldsymbol{q}$.cumsum()
$\boldsymbol{mask} \leftarrow \boldsymbol{q}_{\text{sum}} \frac{\boldsymbol{f}}{\boldsymbol{q}} - \boldsymbol{f}_{\text{sum}} < \frac{\varepsilon}{2}$
$i \leftarrow \boldsymbol{mask}$.argmax()

$w \leftarrow \dfrac{\boldsymbol{f}_{\text{sum}}[i] + \frac{\varepsilon}{2}}{\boldsymbol{q}_{\text{sum}}[i]}$
Reindex $\boldsymbol{f}, \boldsymbol{q}$ back to the original
return torch.max($\boldsymbol{f}, w\boldsymbol{q}$)

To further speed up, we also propose Algorithm 3, a GPU-based water-filling. Specifically, we manage to eliminate the loop and branch in Algorithm 2, making it completely sequential and suitable for GPUs. By utilizing the operators of PyTorch tensors, we fully leverage the parallelism capabilities of GPUs.

D EXPERIMENTS ON COMPUTATIONAL COST

We quantitatively demonstrate the efficiency of our post-processing Algorithm 1 by experiments. The target models, training sets, and defense settings are consistent with Table 4. We take a batch with 512 test samples and let the model infer 100 times on it. We record the time cost by torch.profiler, an official tool provided by PyTorch. We exclude the time for I/O (i.e. the time from disk to memory, and from CPU to GPU), and only include the time for forward propagation on GPU. Our experiment is conducted on one NVIDIA GeForce RTX 3090. The results are in Table 5.

Table 5: The time cost of our post-processing algorithm.

	IR-152 & CelebA	IR-152 & FaceScrub	VGG-16 & FaceScrub
Time without defense	18.63 s	17.70 s	5.65 s
Time with our defense	19.22 s	18.16 s	6.07 s
Percent of increased time	3.1%	2.5%	7.4%

It can be seen that we only increase the time by 2.5% to 7.4%. The higher percent on VGG is due to the shallower model structure. In absolute terms, modifying 512 predictions for 100 times only needs 0.5 seconds. If we take the I/O time into account, the percents will be small enough to be ignored.

We further investigate the relationship between $|\mathbb{Y}|$ and the time cost of our Algorithm 3. We generate $\boldsymbol{s} \in \mathbb{R}^{|\mathbb{Y}|} \sim N(\boldsymbol{0}, \boldsymbol{I})$ and let $\boldsymbol{r} \leftarrow \mathrm{softmax}(10\boldsymbol{s})$. It is observed that the $\boldsymbol{r}$ generated in this way is close to the real probability distributions. We use these $\boldsymbol{r}$ to simulate the real $\boldsymbol{f}(\boldsymbol{x})$ and $\boldsymbol{q}^y$, and let our GPU-based water-filling to find the optimal solution $\boldsymbol{p}$. We take a batch with 256 pairs $(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{q}^y)$ and solve in parallel. The time costs are shown in Table 6.

Table 6: The relationship between $|\mathbb{Y}|$ and the time cost of our GPU-based water-filling.

$\lvert\mathbb{Y}\rvert$	10^1	10^2	10^3	10^4	10^5	10^6
Time	131 ms	132 ms	143 ms	163 ms	249 ms	1301 ms

It shows that even when $|\mathbb{Y}|$ reaches a million, solving 256 convex optimization problems only takes 1.3 seconds. We believe that at this point, our post-processing will not be the performance bottleneck, but the slow inferring and massive parameters of the target model will be.

E ESTIMATE $\boldsymbol{q}^y$ VIA TRAINING SAMPLES

In our Algorithm 1, we estimate $\boldsymbol{q}^y$ by finding one $(\boldsymbol{x}', y) \in D_{\text{valid}}$ and let $\tilde{\boldsymbol{q}}^y := \boldsymbol{f}(\boldsymbol{x}')$. Actually, based on the

$$\boldsymbol{q}^y = \mathbb{E}_{\boldsymbol{X}|Y=y}[\boldsymbol{f}(\boldsymbol{X})] \tag{24}$$

in (10), we can propose another estimate

$$\tilde{\boldsymbol{q}}^y := \operatorname*{mean}_{(\boldsymbol{x},y)\in D_{\text{train}}} \boldsymbol{f}(\boldsymbol{x}), \tag{25}$$

which is the average prediction of the samples that labeled y in the training set.

We explore which estimation is better through experiments. All other settings are consistent with Tables 1-4, where the target model is IR-152 and the private dataset is CelebA. The results on MIA robustness are shown in Table 7, and the results on model's utility are shown in Table 8.

Table 7: The MIA robustness on different estimations of $\boldsymbol{q}^y$.

Estimate $\boldsymbol{q}^y$ by	Mirror				BREP			
	$\downarrow Acc@1$	$\downarrow Acc@5$	$\uparrow \delta_{eval}$	$\uparrow \delta_{face}$	$\downarrow Acc@1$	$\downarrow Acc@5$	$\uparrow \delta_{eval}$	$\uparrow \delta_{face}$
D_{valid}	**1.2%**	**2.9%**	2527	**1.56**	**0.4%**	**1.6%**	**2362**	1.61
D_{train}	**1.2%**	3.0%	**2531**	1.54	**0.4%**	**1.6%**	2355	**1.62**

Table 8: The model's utility on different estimations of $\boldsymbol{q}^y$.

Estimate $\boldsymbol{q}^y$ by	$\uparrow$ Acc	$\downarrow$ Avg L_1	$\downarrow$ Max L_1
D_{valid}	**90.3%**	**0.15**	**0.95**
D_{train}	90.0%	0.18	0.97

We find that there is almost no difference in MIA robustness between the two. Only in terms of model's utility, D_{valid} is a little better than D_{train}. We believe the reason is that the estimation of D_{train} is not accurate enough due to overfitting. However, the gap is very small, so we suggest that: when no validation set available, the training set can be used to estimate $\boldsymbol{q}^y$ by (25).

F EXPERIMENTS UNDER RLB ATTACK

We evaluate the all defenses' MIA robustness against RLB (Han et al., 2023), a SOTA soft-label attack method. All settings are consistent with Tables 1-4, where the target model is IR-152 and the private dataset is CelebA. The first 10 classes of CelebA are attacked and each class reconstructed 5 images. The results are shown in Table 9.

Table 9: The MIA robustness of all defense under RLB attack.

	$\downarrow Acc@1$	$\downarrow Acc@5$	$\uparrow \delta_{eval}$	$\uparrow \delta_{face}$
No Defense	32%	64%	2006	0.77
MID	30%	48%	2088	0.84
BiDO	16%	28%	2254	0.94
LS	12%	34%	2204	0.85
TL	22%	34%	2107	0.82
SSD (ours)	**8%**	**12%**	**2480**	**1.26**

It can be seen that our defense has the best MIA robustness against RLB. The models' utility and defenses' settings are consistent with the Tables 3-4, which shows that we also preserve the best model's utility.

G EXPERIMENTS ON HIGH RESOLUTION

To adapt to high resolution, we choose Mirror as the attacker. The prior distribution is StyleGAN2 trained on FFHQ with a resolution of 1024×1024. The generated images are center-cropped to 800×800, resized to 224×224, and inputted to the target model. The target model is ResNet-152 and the evaluation model is Inception-v3. The first 10 classes of FaceScrub are attacked and each class reconstructed 5 images. The attack results are shown in Table 10 and the models' utility are shown in Table 11. Although models are more vulnerable on high resolution, our defense still achieves the best MIA robustness, with a good utility.

Table 10: The MIA robustness of all defenses under Mirror attack on high resolution.

	$\downarrow Acc@1$	$\downarrow Acc@5$	$\uparrow \delta_{eval}$	$\uparrow \delta_{face}$
No Defense	70%	94%	195	0.84
MID	62%	90%	183	0.76
BiDO	66%	86%	194	0.90
LS	48%	82%	202	0.87
TL	58%	92%	191	0.80
SSD (ours)	**42%**	**66%**	**211**	**1.13**

Table 11: The target models' utility and defenses' settings on high resolution.

	$\uparrow$ Acc	$\downarrow$ Avg L_1	$\downarrow$ Max L_1	Settings
No Defense	98.5%	0	0	–
MID	96.7%	0.30	**1.97**	$\beta = 0.005$
BiDO	96.3%	0.09	1.99	$\lambda_x = 0.15, \lambda_y = 1.5$
LS	96.5%	0.11	1.99	$\alpha = -0.01$
TL	96.7%	0.19	1.99	First 70% layers
SSD (ours)	**96.9%**	**0.07**	1.98	$T = 1, \varepsilon = 20$

H Discussion on adaptive attacks

In this section we discuss adaptive attacks, where attackers are aware of our defense and take targeted actions.

Firstly, we believe that launching adaptive attacks in black-box scenarios is unrealistic, because attackers don't know the target model, and naturally don't know its defense strategy. If they were to guess the defense strategy based on the model's behavior, they would need to consume a large number of queries.

Step back and consider, if attackers know our defense, their best strategy is:

1. Query the same $\boldsymbol{x}$ repeatedly and count the frequency of different outputs.
2. Estimate our sampling probability $\mathbb{P}(y|\boldsymbol{x})$ by the frequency they count.
3. Infer our true prediction $\mathbb{P}(\hat{y}|\boldsymbol{x})$ by the $\mathbb{P}(y|\boldsymbol{x})$ they estimate and the temperature T (assuming they know).

If an online server detects such pattern of queries, it can block them. Step back and consider again, we propose a memory-free and low-cost improvement to block such adaptive attacks:

Design a hash function $h : \mathbb{X} \to \mathbb{N}$, where $\mathbb{X}$ is the input space and $\mathbb{N}$ is the set of integers. When users/attackers query $\boldsymbol{x}$, we take $h(\boldsymbol{x})$ as the random seed for sampling, ensuring same-input-same-output. However, attackers can add subtle perturbations to $\boldsymbol{x}$, therefore our h needs to be robust. For example, it can be

$$h(\boldsymbol{x}) \coloneqq \sum_{i=1}^{m} \lfloor k \cdot z_i(\boldsymbol{x}) \rfloor, \tag{26}$$

where $\boldsymbol{z}(\boldsymbol{x}) \in \mathbb{R}^m$ is the penultimate layer feature in target model, and k is the sensitivity coefficient. Note that $\boldsymbol{z}(\boldsymbol{x})$ are commonly used to evaluate the similarity between two images, i.e., the closer the two $\boldsymbol{z}(\boldsymbol{x})$ are, the more similar the two $\boldsymbol{x}$ look. The larger k is, the more numerically sensitive h is, and the more random our defense is.

How to evaluate and improve h is a new and interesting topic, worth studying deeply in the future.