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

Tianqu Zhuang[1*], Hongyao Yu[2*], Yixiang Qiu[1*], Hao Fang[1*], Bin Chen[2#], Shu-Tao Xia[1]
[1]Shenzhen International Graduate School, Tsinghua University, China
[2]Harbin Institute of Technology, Shenzhen, China
{zhuangtq23, qiu-yx24, fang-h23}@mails.tsinghua.edu.cn; yuhongyao@stu.hit.edu.cn;
chenbin2021@hit.edu.cn; xiast@sz.tsinghua.edu.cn; [*]Equal contribution [#]Corresponding author

ABSTRACT

Model inversion attacks (MIAs) aim to reconstruct the private training data by accessing a public model, raising concerns about privacy leakage. Black-box MIAs, where attackers can only query the model and obtain outputs, are closer to real-world scenarios. The latest black-box attacks have outperformed the state-of-the-art white-box attacks, and existing defenses cannot resist them effectively. To fill this gap, we propose Stealthy Shield Defense (SSD), a post-processing algorithm against black-box MIAs. Our idea is to modify the model's outputs to minimize the conditional mutual information (CMI). We mathematically prove that CMI is a special case of information bottlenecks (IB), and thus inherits the advantages of IB—making predictions less dependent on inputs and more dependent on ground truths. This theoretically guarantees our effectiveness, both in resisting MIAs and preserving utility. For minimizing CMI, we formulate a convex optimization problem and solve it via the water-filling method. Adaptive rate-distortion is introduced to constrain the modification to the outputs, and the water-filling is implemented on GPUs to address computation cost. Without the need to retrain the model, our algorithm is plug-and-play and easy to deploy. Experimental results indicate that SSD outperforms existing defenses, in terms of MIA resistance and model's utility, across various attack algorithms, training datasets, and model architectures. Our code is available at `https://github.com/ZhuangQu/Stealthy-Shield-Defense`.

1 INTRODUCTION

Deep neural networks (DNNs) have driven widespread deployment in multiple mission-critical domains, such as computer vision (He et al., 2015), natural language processing (Devlin et al., 2019) and dataset distillation (Zhong et al., 2024b;a). However, their integration with sensitive training data has raised concerns about privacy breaches. Recent studies (Fang et al., 2024b;a; 2025) have explored various attack methods to probe these privacy, such as gradient inversion (Fang et al., 2023; Yu et al., 2024b) and membership inference (Hu et al., 2021). Among the emergent threats, model inversion attacks (MIAs) aim to reconstruct the private training data by accessing a public model, posing the greatest risk (Qiu et al., 2024b). For instance, consider a face recognition access control system with a publicly accessible interface. Through carefully crafted malicious queries, model inversion attackers can infer the sensitive facial images stored in the system, along with the associated user identities.

MIAs are divided into *white-box* and *black-box* (Fang et al., 2024c). White-box attackers know the details of the model, whereas black-box attackers can only query the model and obtain outputs. Black-box MIAs become more threatening than white-box because: **(1) Black-box scenarios are more common.** As models grow larger nowadays, they are mostly stored on servers and can only be accessed online, which are typical black-box scenarios. **(2) Black-box attacks are more powerful.** The latest soft-label attack RLBMI (Han et al., 2023) and hard-label attack LOKT (Nguyen et al., 2023) have outperformed the state-of-the-art white-box attacks. **(3) Existing defenses cannot resist**

black-box attacks effectively. Existing defenses focus on modifying the weights and structure of the model, but black-box attackers only exploit the outputs, and thus are less susceptible.

To address these concerns, we propose Stealthy Shield Defense (SSD), a post-processing algorithm against black-box MIAs. As shown in Figure 1, the idea of SSD is to modify the model's outputs to minimize the conditional mutual information (CMI) (Yang et al., 2024). CMI quantifies the dependence between inputs and predictions when ground truths are given. In Theorem 1, we prove that CMI is a special case of information bottlenecks (IB), and thus inherits the advantages of IB—making predictions less dependent on inputs and more dependent on ground truths. Under this theoretical guarantee, SSD achieves a better trade-off between MIA resistance and model's utility. Without the need to retrain the model, SSD is plug-and-play and easy to deploy.

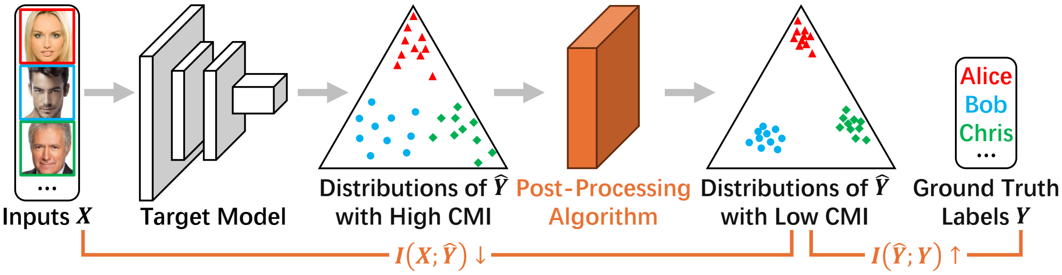

Figure 1: An overview of Stealthy Shield Defense. The probability simplex is a triangle when the number of classes is three. CMI is defined as $\mathcal{I}(X;\hat{Y}|Y)$. According to our Theorem 1, minimizing CMI makes the mutual information $\mathcal{I}(X;\hat{Y})$ minimized and $\mathcal{I}(\hat{Y};Y)$ maximized. As shown by Yang et al. (2024), minimizing CMI makes the outputs more concentrated class-wisely.

The contributions of this paper are:

- We introduce CMI into model inversion defense for the first time, and theoretically prove its effectiveness.
- We propose a post-processing algorithm to minimize CMI without retraining models. In our algorithm, temperature is introduced to calibrate the probabilities and adaptive rate-distortion is introduced to constrain the modification to the outputs. We speed up our algorithm by GPU-based water-filling method as well.
- Our experiments indicate that we outperform all competitors, in terms of MIA-resistance and model's utility, exhibiting good generalizability across various attack algorithms, training datasets, and model architectures.

2 Related Work

2.1 Model Inversion Attacks and Defenses

Model inversion attacks (MIAs) are a serious privacy threat to released models (Fang et al., 2024c). MIAs are categorized as *white-box* (Zhang et al., 2019; Chen et al., 2020; Struppek et al., 2022; Yuan et al., 2023; Qiu et al., 2024a) and *black-box*. We focus on black-box MIAs, where attackers can only query the model and obtain outputs. In this scenario, BREP (Kahla et al., 2022) utilizes zero-order optimization to drive the latent vectors away from the decision boundary. Mirror (An et al., 2022) and C2F (Ye et al., 2024b) explore genetic algorithms. LOKT (Nguyen et al., 2023) trains multiple surrogate models and applies white-box attacks to them.

To address the threat of MIAs, a variety of defenses have been proposed. MID (Wang et al., 2020), BiDO (Peng et al., 2022), and LS (Struppek et al., 2023) change the training losses, TL (Ho et al., 2024) freezes some layers of the model, and CA-FaCe (Yu et al., 2024a) change the structure of the model. However, black-box attackers only exploit the outputs, and thus are rarely hindered. The defense against black-box MIAs is still limited.

In this paper, we propose a novel black-box defense based on post-processing, without retraining the model. Experimental results indicate that we outperform the existing defenses.

2.2 Information Bottleneck and Conditional Mutual Information

Tishby et al. (2000) proposed the Information Bottleneck (IB) principle: a good machine learning model should compress the redundant information in inputs while preserving the useful information for tasks. They later highlighted that information is compressed layer-by-layer in DNNs (Tishby & Zaslavsky, 2015; Shwartz-Ziv & Tishby, 2017). Alemi et al. (2017) proposed Variational Information Bottleneck (VIB) to estimate the bounds of IB, and Wang et al. (2020) applied VIB in their Mutual Information-based Defense (MID).

Yang et al. (2024) proposed to use conditional mutual information (CMI) as a performance metric for DNNs, providing the calculation formula and geometric interpretation of CMI. By minimizing CMI, they improve classifiers (Yang et al., 2025) and address class imbalance (Hamidi et al., 2024). By maximizing CMI, they improve knowledge distillation (Ye et al., 2024a) and address nasty teachers (Yang & Ye, 2024).

In this paper, we theoretically prove that CMI is a special case of IB and thus inherits the advantages of IB. Furthermore, we propose a novel model inversion defense based on CMI.

3 Preliminary

3.1 Notation

Let $f\colon \mathbb{X} \to \mathbb{Y}$ be a neural classifier, $X \in \mathbb{X}$ be an input to f, $Y \in \mathbb{Y}$ be the ground truth label, $\hat{Y} \in \mathbb{Y}$ be the label predicted by f, and $Z \in \mathbb{Z}$ be the intermediate representation in f. Note that $Y \to X \to Z \to \hat{Y}$ is a Markov chain. Let $\mathcal{P}$ be the probability function and $\mathcal{P}(x) := \mathcal{P}\{X = x\}$, $\mathcal{P}(y) := \mathcal{P}\{Y = y\}$, $\mathcal{P}(x, \hat{y}|y) := \mathcal{P}\{X = x, \hat{Y} = \hat{y} \mid Y = y\}$, etc. Note that $\mathcal{P}(x, y)$ is the private data distribution.

Let $\Delta^{\mathbb{Y}}$ be the probability simplex with $|\mathbb{Y}|$ vertices. Let $\boldsymbol{f}(x) \in \Delta^{\mathbb{Y}}$ be the output from the softmax layer of f when x is input to f, and $f_{\hat{y}}(x) \in [0, 1]$ be the $\hat{y}$-th component of $\boldsymbol{f}(x)$, $\hat{y} \in \mathbb{Y}$. Note that $f(x) = \arg\max_{\hat{y} \in \mathbb{Y}} f_{\hat{y}}(x)$.

3.2 Model Inversion Attacks

Let $D \subseteq \mathbb{X} \times \mathbb{Y}$ be the dataset learned by f. Note that the samples in D are i.i.d. to $\mathcal{P}(x, y)$. MIAs aim to reconstruct $\hat{D}$ as close to D as possible. Based on the access to f, MIAs are categorized as:

Hard-label: Attackers can query any $x \in \mathbb{X}$ and obtain $f(x) \in \mathbb{Y}$.

Soft-label: Attackers can query any $x \in \mathbb{X}$ and obtain $\boldsymbol{f}(x) \in \Delta^{\mathbb{Y}}$.

White-box: Attackers know the details of f.

Hard-label and *soft-label*, collectively called *black-box*,[1] are defended against in this paper.

3.3 Mutual Information-Based Defense (MID)

Wang et al. (2020) proposed to resist MIAs by reducing the dependence between X and $\hat{Y}$. The dependence is quantified by the mutual information, which is defined as

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

They reduced $\mathcal{I}(X; \hat{Y})$ to prevent attackers from inferring the information of D. However, low $\mathcal{I}(X; \hat{Y})$ hurts the model's utility. Especially, $\mathcal{I}(X; \hat{Y}) = 0$ iff X and $\hat{Y}$ are independent, in which case f is immune to any attack but useless at all.

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

As an alternative, they introduced the information bottleneck (IB), which is defined as

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

where $\lambda > 0$. They used it as a regularizer to train f, minimizing $\mathcal{I}(X;Z)$ to resist MIAs while maximizing $\mathcal{I}(Z;Y)$ to preserve the model's utility.

4 Methodology

4.1 Conditional Mutual Information-Based Defense

We aim to resist black-box MIAs where attackers cannot access Z, so we still minimize $\mathcal{I}(X;\hat{Y})$ instead of $\mathcal{I}(X;Z)$.

Furthermore, we observe that all MIA algorithms target one fixed label. Formally, let

$$D^y \coloneqq \{x \in \mathbb{X} : (x,y) \in D\}$$

be the sub-dataset whose ground truth label is y. For a given $y \in \mathbb{Y}$, all attackers aim to reconstruct $\hat{D}^y$ as close to D^y as possible. Against their intention, we propose to minimize

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

$\mathcal{I}(X;\hat{Y}|Y=y)$ quantifies the dependence between X and $\hat{Y}$ when $Y=y$. We minimize it to prevent attackers from inferring the information of D^y.

To protect the complete D, we minimize (3) for each $y \in \mathbb{Y}$ with the weight of $\mathcal{P}(y)$. It is equivalent to minimizing the conditional mutual information (CMI), which is defined as

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

Theorem 1. *CMI is a special case of the information bottleneck (2) when $Z=\hat{Y}$ and $\lambda = 1$, i.e.*

$$\mathcal{I}(X;\hat{Y}|Y) = \mathcal{I}(X;\hat{Y}) - \mathcal{I}(\hat{Y};Y).$$

Our proof is provided in Appendix A. Our theorem proves that CMI inherits the benefits of IB in two aspects:

- Minimize $\mathcal{I}(X;\hat{Y})$ to compress the redundant information in inputs, and decrease the dependence between inputs and predictions. It improves the resistance to MIAs as shown by Wang et al. (2020).
- Maximize $\mathcal{I}(\hat{Y};Y)$ to preserve the useful information for tasks, and increase the dependence between predictions and ground truths. It improves the utility obviously.

$\mathcal{I}(X;Z)$ in IB is challenging to calculate because the input space $\mathbb{X}$ and representation space $\mathbb{Z}$ are both high-dimensional. Previous work could only approximate IB by variational bounds (Alemi et al., 2017). Fortunately, as a special case of IB, CMI can be calculated directly (Yang et al., 2024).

4.2 Minimize CMI via Post-Processing

Previous work used CMI as a regularizer and minimized it during training models (Yang et al., 2024; Hamidi et al., 2024; Yang et al., 2025). In contrast to them, we minimize CMI via post-processing.

We transform CMI as follows:

$$\begin{aligned}\mathcal{I}(X;\hat{Y}|Y) &= \sum_{y\in\mathbb{Y}}\mathcal{P}(y)\sum_{x\in\mathbb{X}}\sum_{\hat{y}\in\mathbb{Y}}\mathcal{P}(x,\hat{y}|y)\log\frac{\mathcal{P}(x,\hat{y}|y)}{\mathcal{P}(x|y)\mathcal{P}(\hat{y}|y)}, && \text{by definitions (3-4),}\\ &= \sum_{x\in\mathbb{X}}\sum_{\hat{y}\in\mathbb{Y}}\sum_{y\in\mathbb{Y}}\mathcal{P}(x,\hat{y},y)\log\frac{\mathcal{P}(\hat{y}|x,y)}{\mathcal{P}(\hat{y}|y)},\\ &= \sum_{x\in\mathbb{X}}\mathcal{P}(x)\sum_{y\in\mathbb{Y}}\mathcal{P}(y|x)\sum_{\hat{y}\in\mathbb{Y}}\mathcal{P}(\hat{y}|x,y)\log\frac{\mathcal{P}(\hat{y}|x,y)}{\mathcal{P}(\hat{y}|y)},\\ &= \sum_{x\in\mathbb{X}}\mathcal{P}(x)\sum_{y\in\mathbb{Y}}\mathcal{P}(y|x)\sum_{\hat{y}\in\mathbb{Y}}\mathcal{P}(\hat{y}|x)\log\frac{\mathcal{P}(\hat{y}|x)}{\mathcal{P}(\hat{y}|y)}, && \text{by Markov chain } Y\to X\to\hat{Y}.\end{aligned}$$

Thus minimizing $\mathcal{I}(X;\hat{Y}|Y)$ is equivalent to minimizing $\sum_{y\in\mathbb{Y}}\mathcal{P}(y|x)\sum_{\hat{y}\in\mathbb{Y}}\mathcal{P}(\hat{y}|x)\log\frac{\mathcal{P}(\hat{y}|x)}{\mathcal{P}(\hat{y}|y)}$ for each x input to f. For simplicity, we sample $y\in\mathbb{Y}$ with the probability of $\mathcal{P}(y|x)$ and minimize $\sum_{\hat{y}\in\mathbb{Y}}\mathcal{P}(\hat{y}|x)\log\frac{\mathcal{P}(\hat{y}|x)}{\mathcal{P}(\hat{y}|y)}$ instead.[2] It is equal to the original objective in terms of mathematical expectation. Next we need $\mathcal{P}(\hat{y}|x)$, $\mathcal{P}(y|x)$ and $\mathcal{P}(\hat{y}|y)$.

To get $\mathcal{P}(\hat{y}|x)$, we have $\mathcal{P}(\hat{y}|x)=f_{\hat{y}}(x)$ by the design of neural classifiers.

To get $\mathcal{P}(y|x)$, an intuitive idea is that $\mathcal{P}(y|x)=\mathcal{P}(\hat{y}|x)$ for $y=\hat{y}$. But Guo et al. (2017) have demonstrated that it is inaccurate in modern neural networks. Inspired by their work, we introduce the temperature mechanism to adjust it.

To get $\mathcal{P}(\hat{y}|y)$, we have

$$\begin{aligned}\mathcal{P}(\hat{y}|y) &= \sum_{x\in\mathbb{X}}\mathcal{P}(x,\hat{y}|y) = \sum_{x\in\mathbb{X}}\mathcal{P}(x|y)\mathcal{P}(\hat{y}|x,y) = \sum_{x\in\mathbb{X}}\mathcal{P}(x|y)\mathcal{P}(\hat{y}|x),\\ &= \sum_{x\in\mathbb{X}}\mathcal{P}(x|y)f_{\hat{y}}(x) = \mathbb{E}_{X|Y=y}[f_{\hat{y}}(X)] \approx \operatorname*{mean}_{x'\in D^y} f_{\hat{y}}(x'),\end{aligned}$$

where the "$\approx$" is based on the fact that the samples in D^y are i.i.d. to $\mathcal{P}(x|y)$, and thus the sample mean can estimate the conditional expectation. In practice we use the validation set as D^y, because the training samples are overfitted by f, causing inaccurate estimation.

Now the objective becomes

$$\sum_{\hat{y}\in\mathbb{Y}}\mathcal{P}(\hat{y}|x)\log\frac{\mathcal{P}(\hat{y}|x)}{\mathcal{P}(\hat{y}|y)} \approx \sum_{\hat{y}\in\mathbb{Y}}f_{\hat{y}}(x)\log\frac{f_{\hat{y}}(x)}{\operatorname*{mean}_{x'\in D^y}f_{\hat{y}}(x')} = \mathrm{KL}(\boldsymbol{f}(x)||\operatorname*{\mathbf{mean}}_{x'\in D^y}\boldsymbol{f}(x')),$$

where KL is Kullback-Leibler divergence, a binary convex function. To minimize it, we fix $\operatorname*{\mathbf{mean}}_{x'\in D^y}\boldsymbol{f}(x')$ for simplicity and modify $\boldsymbol{f}(x)$. Let $\boldsymbol{p}\in\Delta^{\mathbb{Y}}$ be the modified version of $\boldsymbol{f}(x)$ and our objective is $\mathrm{KL}(\boldsymbol{p}||\operatorname*{\mathbf{mean}}_{x'\in D^y}\boldsymbol{f}(x'))$. Additionally, we constrain $\|\boldsymbol{p}-\boldsymbol{f}(x)\|_1\le\varepsilon$ to preserve the model's utility, where $\varepsilon>0$ is the distortion controller.

In information theory, minimizing mutual information under bounded distortion constraints is known as the rate-distortion problem (Shannon, 1959) for signal compression. If a signal has less information, it is easier to compress, and a stricter distortion bound can be applied. Inspired by their work, we introduce Shannon entropy to quantify the information in $\hat{Y}$ when $X=x$, which is defined as

$$\mathcal{H}(\hat{Y}|X=x) \coloneqq -\sum_{\hat{y}\in\mathbb{Y}}\mathcal{P}(\hat{y}|x)\log\mathcal{P}(\hat{y}|x).$$

Our new constraint is $\|\boldsymbol{p}-\boldsymbol{f}(x)\|_1\le\varepsilon\cdot\mathcal{H}(\hat{Y}|X=x)$, where the distortion bound is proportional to the amount of information. It reduces the modification when the information is limited, and enhances the compression when the information is abundant. We refer to it as *adaptive rate-distortion*.

[2] After sampling, we only need to consider one $y\in\mathbb{Y}$ and all $\hat{y}\in\mathbb{Y}$, so we can solve it in $O(|\mathbb{Y}|\log|\mathbb{Y}|)$ time (Algorithm 2). Without sampling, we have to consider all $y,\hat{y}\in\mathbb{Y}$. The time complexity is $\Omega(|\mathbb{Y}|^2)$, which is unacceptable when $|\mathbb{Y}|$ is large.

Algorithm 1: post-processing to minimize CMI.

```
Input: original output $\boldsymbol{f}(x)$, temperature $T$, distortion controller $\varepsilon$, validation set $D$.
Output: modified output $\boldsymbol{p}$.
Sample $y \in \mathbb{Y}$ with the probability of softmax($\frac{\boldsymbol{f}(x)}{T}$);
$\boldsymbol{q}^y \leftarrow \underset{x' \in D^y}{\mathbf{mean}} \, \boldsymbol{f}(x')$;
$\mathcal{H} \leftarrow -\sum_{\hat{y} \in \mathbb{Y}} f_{\hat{y}}(x) \log f_{\hat{y}}(x)$;
Solve the convex optimization problem and return the optimal $\boldsymbol{p}$:
    $$\begin{aligned} &\min \mathrm{KL}(\boldsymbol{p} || \boldsymbol{q}^y), \\ &\text{s.t. } \|\boldsymbol{p} - \boldsymbol{f}(x)\|_1 \le \varepsilon \cdot \mathcal{H}, \\ &\quad \boldsymbol{p} \in \Delta^{\mathbb{Y}}. \end{aligned} \tag{5}$$
```

Our defense is implemented by Algorithm 1. Without the need to retrain the model, it is plug-and-play and easy to deploy.

Note that the $\boldsymbol{q}^y, y \in \mathbb{Y}$ can be calculated and stored in advance to reduce computation cost. If the model owner differs from the defender, the owner only needs to provide the defender with $\boldsymbol{q}^y$ instead of D, avoiding communication cost and privacy risks.

(5) is a convex optimization problem that can be solved by existing optimizers. Furthermore, we derive the explicit solution in Appendix B, calculate it by Water-Filling in Algorithm 2, accelerate it by GPUs in Algorithm 3, and evaluate the computation cost in Appendix C.

5 Experiment

5.1 Settings

Datasets. Following the previous work, we use CelebA (Liu et al., 2014) and FaceScrub (Ng & Winkler, 2014) as private datasets. CelebA contains 10,177 labels and we only take 1000 labels with the most images (Kahla et al., 2022). FaceScrub contains 530 labels and 43,147 images.[3] All images are cropped and resized to 64×64 pixels. We use 80% of the data for training, 10% for validation, and 10% for testing. The validation set is used to select the best trained models, training hyperparameters, and defense hyperparameters.

Models. VGG-16 (Simonyan & Zisserman, 2014) and IR-152 (He et al., 2015) are selected as target models. They are trained with various defenses. The evaluation model is a FaceNet (Cheng et al., 2017).

Attacks. We focus on state-of-the-art black-box MIAs, including BREP (Kahla et al., 2022), Mirror (An et al., 2022), C2FMI (Ye et al., 2024b), LOKT (Nguyen et al., 2023) and RLBMI (Han et al., 2023). We attack the first 100 labels in the private dataset, reconstructing 5 images for each label. For BREP and LOKT, we use the FFHQ (Karras et al., 2019) to train GANs and surrogate models under official settings. For Mirror and C2FMI, we adopt the 256×256 GAN trained on FFHQ provided by (Karras et al., 2019). The generated images are center-cropped to 176×176 and then resized to 64×64.

Metrics. To evaluate the MIA resistance and model's utility, 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 compute the top-1 and top-5 classification accuracies, denoted as "acc" and "acc5", respectively.

[3] The original FaceScrub contains 106,863 images, but some images are unavailable because their URLs are invalid.

- **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 (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.
- **Distortion.** This metric is used to quantify the modification to the predicted probability vectors by defenses. We take the L_1 distance between the outputs with and without defense. It is denoted as "dist".

All experiments are conducted by MIBench (Qiu et al., 2024b).

5.2 Comparison with State-of-the-art Defenses

Table 1: MIA resistance of various defenses under soft-label attacks.

		Mirror				**C2FMI**			
		$\downarrow$ acc	$\downarrow$ acc5	$\uparrow \sigma_{eval}$	$\uparrow \sigma_{face}$	$\downarrow$ acc	$\downarrow$ acc5	$\uparrow \sigma_{eval}$	$\uparrow \sigma_{face}$
IR-152 CelebA	**None**	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
	SSD	**1.2%**	**3.0%**	2527	**1.56**	**0%**	**0.4%**	2377	**1.67**
IR-152 FaceScrub	**None**	39.6%	63.2%	2135	0.88	17.6%	41.2%	2196	1.03
	MID	40.0%	61.2%	2152	0.96	**3.2%**	7.6%	3055	1.36
	BiDO	31.0%	55.6%	2168	0.92	12.2%	25.6%	2528	1.14
	LS	28.8%	56.8%	2286	0.90	11.0%	30.4%	2390	1.07
	TL	31.2%	51.6%	2175	0.98	7.4%	21.0%	2341	1.24
	SSD	**22.8%**	**35.8%**	**2753**	**1.18**	**3.2%**	**7.2%**	**3107**	**1.38**
VGG-16 FaceScrub	**None**	9.2%	24.8%	2740	1.02	5.8%	13.2%	2907	1.14
	MID	17.4%	38.0%	2518	0.95	2.0%	7.0%	2986	1.22
	BiDO	5.2%	17.2%	2911	1.06	5.0%	14.8%	2625	1.11
	LS	13.8%	30.4%	2557	0.99	7.0%	20.4%	2662	1.08
	TL	7.0%	18.6%	2777	1.06	7.6%	19.8%	2565	1.11
	SSD	**5.0%**	**13.8%**	**2970**	**1.17**	**0.8%**	**5.0%**	**3223**	**1.37**

We compare our SSD with other state-of-the-art defenses, including MID (Wang et al., 2020), BiDO (Peng et al., 2022), LS (Struppek et al., 2023) and TL (Ho et al., 2024). We adhere to the official implementations for each defense, and the corresponding hyperparameters are detailed in Appendix D.

For soft-label attacks, the results are listed in Table 1. It can be seen that our SSD significantly reduces the attack effect of Mirror and C2FMI, surpassing the other four defenses. In particular, for the IR-152 model trained on FaceScrub, we increase the face feature distance from 0.88 to 1.18, while the other defenses fall below 1.0.

For hard-label attacks, our SSD also shows a strong effect presented in Table 2. Note that LOKT is the most powerful hard-label attack, and existing defenses cannot resist it. Our SSD drops the acc of LOKT to 1/5 of the original. In addition, judging from the attack results of BREP, some defenses are even advantageous to the hard-label attackers. This demonstrates the need to specifically design black-box defenses.

Table 2: MIA resistance of various defenses under hard-label attacks.

		BREP				LOKT			
		↓ acc	↓ acc5	↑ σ_{eval}	↑ σ_{face}	↓ acc	↓ acc5	↑ σ_{eval}	↑ σ_{face}
IR-152 CelebA	**None**	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
	SSD	**0.4%**	**1.6%**	**2362**	**1.61**	**9.4%**	**17.0%**	**2077**	**1.30**
IR-152 FaceScrub	**None**	31.8%	52.0%	2325	0.94	87.2%	94.8%	1209	0.68
	MID	33.2%	52.4%	2177	1.09	63.8%	81.8%	1550	0.82
	BiDO	24.8%	50.8%	2320	1.01	79.6%	93.6%	1345	0.77
	LS	22.8%	44.6%	2506	1.00	81.2%	94.2%	1285	0.71
	TL	17.8%	39.8%	2440	1.05	88.6%	98.0%	1213	0.73
	SSD	**5.2%**	**8.4%**	**2636**	**1.47**	**13.0%**	**22.8%**	**2279**	**1.29**
VGG-16 FaceScrub	**None**	12.0%	29.2%	2643	1.06	68.2%	86.0%	1382	0.57
	MID	14.6%	37.4%	2460	1.02	51.8%	78.0%	1521	0.60
	BiDO	8.4%	25.0%	2676	1.10	63.2%	84.0%	1523	0.59
	LS	13.4%	29.8%	2578	1.04	60.6%	83.6%	1467	0.65
	TL	8.4%	26.0%	2626	1.08	50.0%	78.8%	1556	0.70
	SSD	**2.6%**	**7.6%**	**2689**	**1.48**	**23.4%**	**34.2%**	**2162**	**1.14**

Figure 2 shows the reconstructed face images of attackers. It can be seen that with our defense, the images reconstructed by attackers are largely different from the private images, which indicates that our SSD is effective.

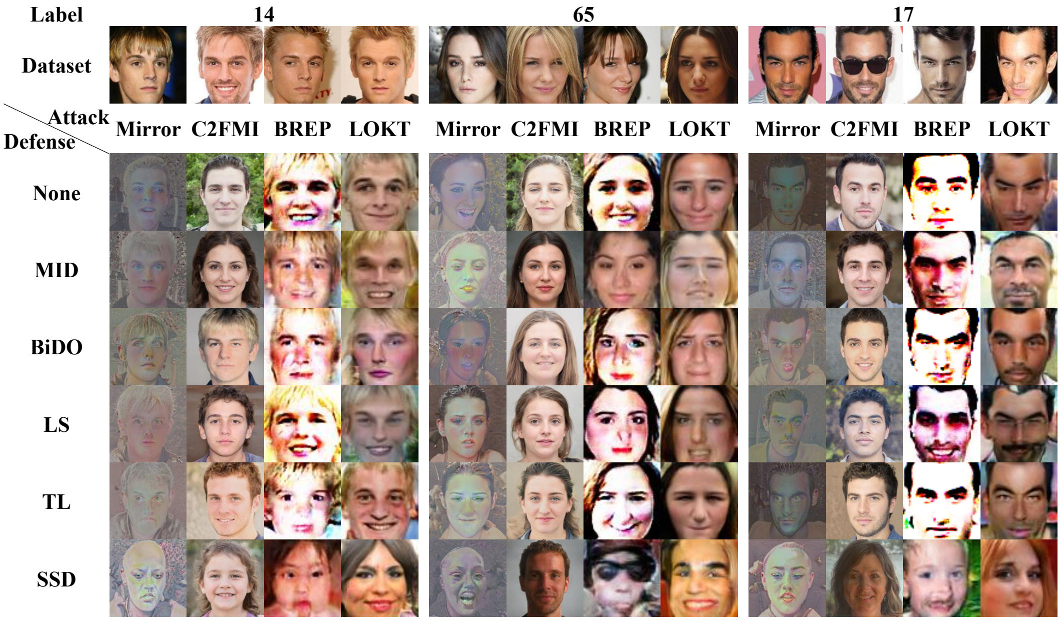

Figure 2: The reconstructed images from IR-152 models trained on CelebA. Above are the ground truth labels y and private sub-datasets D^y (4 samples shown). Below are the reconstructed datasets $\hat{D}^y$ (1 sample shown), over various attacks and defenses.

The evaluation results for the target model's utility are presented in Table 3. Our SSD not only holds the highest test accuracy, but also the smallest distortion, which is only 1/2 to 1/4 of other defenses.

Table 3: Model's utility with various defenses.

	IR-152 & CelebA		IR-152 & FaceScrub		VGG-16 & FaceScrub	
	↑ acc	↓ dist	↑ acc	↓ dist	↑ acc	↓ dist
None	92.1%	0	98.5%	0	92.9%	0
MID	86.8%	0.60	96.0%	0.31	87.2%	0.73
BiDO	86.6%	0.37	95.7%	0.13	88.6%	0.31
LS	86.9%	0.31	95.7%	0.10	88.7%	0.26
TL	86.5%	0.35	95.8%	0.12	88.0%	0.29
SSD	**87.0%**	**0.18**	**97.4%**	**0.05**	**89.4%**	**0.18**

5.3 Ablation Studies

We conduct ablation experiments to explore the effects of temperature T and distortion controller ε in our SSD. The results are shown in Figure 3.

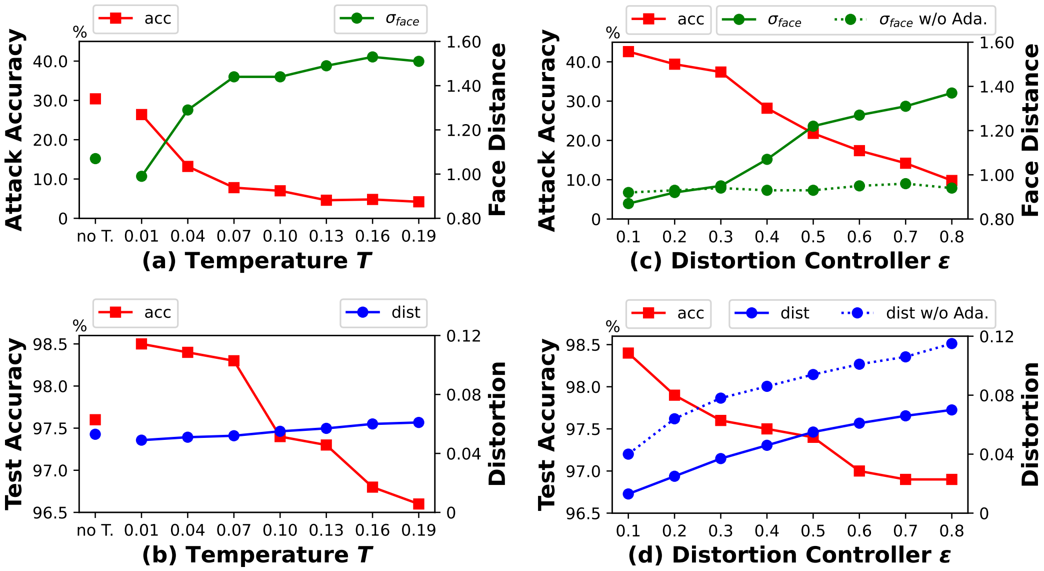

Figure 3: Ablation experiment results of IR-152 models trained on FaceScrub. (a) shows the result attacked by BREP, and (c) shows the result attacked by Mirror. "no T." denotes "no temperature mechanism", and "w/o Ada." denotes "without adaptive rate-distortion".

Figure 3(a) shows that as the temperature increases, the attacker's accuracy decreases, and the reconstructed images become more distant from the private images. This is because the sampling probability in Algorithm 1 is closer to the uniform distribution, which makes it easier to return misleading labels to hard-label attackers. However, high temperature impairs the model's accuracy, which is shown in Figure 3(b). In addition, we find that neither the MIA resistance nor the model's utility is satisfactory without the temperature mechanism. This illustrates the necessity of our introduction of the temperature mechanism.

Figure 3(c) and (d) similarly show that higher distortion controller ε strengthens MIA resistance but impairs the model's utility. In addition, without the adaptive mechanism, the attacker's reconstruction distance is unaffected by ε, and the model's distortion is very severe. This illustrates the necessity of our introduction of the adaptive rate-distortion.

6 Conclusion

In contrast to previous researches on model inversion defense with a focus on white-box attacks, we conduct a specific study on black-box attacks. Specifically, we investigate the impact of conditional mutual information (CMI) and develop a CMI-based defense strategy. We conduct our defense in the post-processing stage instead of re-training the model. Our method modify the model output by reducing the dependence between model inputs and outputs. To further reduce the modifications to outputs, we introduce an adaptive rate-distortion framework and optimize it by the water-filling method. Experimental results demonstrate that our defense method achieves state-of-the-art (SOTA) performance against black-box attacks. We hope that our findings will help shift attention toward robust defense mechanisms in black-box settings and inspire further researches in this area.

7 Acknowledgement

This work is supported in part by National Natural Science Foundation of China (62171248, 62301189), Peng Cheng Laboratory (PCL2023A08), and Shenzhen Science and Technology Program (KJZD20240903103702004, JCYJ20220818101012025, RCBS20221008093124061, GXWD20220811172936001).

References

Alexander A. Alemi, Ian Fischer, and Joshua V. Dillon. Deep variational information bottleneck. *International Conference on Learning Representations (ICLR)*, 2017.

Shengwei An, Guanhong Tao, Qiuling Xu, Yingqi Liu, Guangyu Shen, Yuan Yao, Jingwei Xu, and X. Zhang. Mirror: Model inversion for deep learning network with high fidelity. *Network and Distributed System Security Symposium (NDSS)*, 2022.

Si Chen, Mostafa Kahla, R. Jia, and Guo-Jun Qi. Knowledge-enriched distributional model inversion attacks. *International Conference on Computer Vision (ICCV)*, 2020.

Yu Cheng, Jian Zhao, Zhecan Wang, Yan Xu, Jayashree Karlekar, Shengmei Shen, and Jiashi Feng. Know you at one glance: A compact vector representation for low-shot learning. *International Conference on Computer Vision Workshops (ICCVW)*, 2017.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

Hao Fang, Bin Chen, Xuan Wang, Zhi Wang, and Shutao Xia. Gifd: A generative gradient inversion method with feature domain optimization. *International Conference on Computer Vision (ICCV)*, 2023.

Hao Fang, Jiawei Kong, Bin Chen, Tao Dai, Hao Wu, and Shutao Xia. Clip-guided generative networks for transferable targeted adversarial attacks. *European Conference on Computer Vision (ECCV)*, 2024a.

Hao Fang, Jiawei Kong, Wenbo Yu, Bin Chen, Jiawei Li, Shutao Xia, and Ke Xu. One perturbation is enough: On generating universal adversarial perturbations against vision-language pre-training models. *ArXiv*, 2024b.

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

Hao Fang, Xiaohang Sui, Hongyao Yu, Jiawei Kong, Sijin Yu, Bin Chen, Hao Wu, and Shutao Xia. Retrievals can be detrimental: A contrastive backdoor attack paradigm on retrieval-augmented diffusion models. *ArXiv*, 2025.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. *International Conference on Machine Learning (ICML)*, 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. *International Symposium on Information Theory (ISIT)*, 2024.

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

Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Computer Vision and Pattern Recognition (CVPR)*, 2015.

Sy-Tuyen Ho, Koh Jun Hao, Keshigeyan Chandrasegaran, Ngoc-Bao Nguyen, and Ngai-Man Cheung. Model inversion robustness: Can transfer learning help? *Computer Vision and Pattern Recognition (CVPR)*, 2024.

Hongsheng Hu, Zoran A. Salcic, Lichao Sun, Gillian Dobbie, P. Yu, and Xuyun Zhang. Membership inference attacks on machine learning: A survey. *ACM Computing Surveys (CSUR)*, 2021.

Mostafa Kahla, Si Chen, Hoang A. Just, and R. Jia. Label-only model inversion attacks via boundary repulsion. *Computer Vision and Pattern Recognition (CVPR)*, 2022.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. *Computer Vision and Pattern Recognition (CVPR)*, 2019.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. *International Conference on Computer Vision (ICCV)*, 2014.

Hongwei Ng and Stefan Winkler. A data-driven approach to cleaning large face datasets. *International Conference on Information Photonics (ICIP)*, 2014.

Ngoc-Bao Nguyen, Keshigeyan Chandrasegaran, Milad Abdollahzadeh, and Ngai-Man Cheung. Label-only model inversion attacks via knowledge transfer. *Neural Information Processing Systems (NeurIPS)*, 2023.

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

Yixiang Qiu, Hao Fang, Hongyao Yu, Bin Chen, Meikang Qiu, and Shutao Xia. A closer look at gan priors: Exploiting intermediate features for enhanced model inversion attacks. *European Conference on Computer Vision (ECCV)*, 2024a.

Yixiang Qiu, Hongyao Yu, Hao Fang, Wenbo Yu, Bin Chen, Xuan Wang, Shutao Xia, and Ke Xu. Mibench: A comprehensive benchmark for model inversion attack and defense. *ArXiv*, 2024b.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. *Computer Vision and Pattern Recognition (CVPR)*, 2015.

Claude Elwood Shannon. Coding theorems for a discrete source with a fidelity criteria. *Ire National Convention Record*, 1959.

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

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations (ICLR)*, 2014.

Lukas Struppek, Dominik Hintersdorf, Antonio De Almeida Correia, Antonia Adler, and Kristian Kersting. Plug & play attacks: Towards robust and flexible model inversion attacks. *International Conference on Machine Learning (ICML)*, 2022.

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. *International Conference on Learning Representations (ICLR)*, 2023.

Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. *Information Theory Workshop (ITW)*, 2015.

Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *ArXiv*, 2000.

Tianhao Wang, Yuheng Zhang, and R. Jia. Improving robustness to model inversion attacks via mutual information regularization. *AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

En-Hui Yang and Linfeng Ye. Markov knowledge distillation: Make nasty teachers trained by self-undermining knowledge distillation fully distillable. *European Conference on Computer Vision (ECCV)*, 2024.

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

En-Hui Yang, Shayan Mohajer Hamidi, Linfeng Ye, Renhao Tan, and Beverly Yang. Conditional mutual information constrained deep learning for classification. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 2025.

Linfeng Ye, Shayan Mohajer Hamidi, Renhao Tan, and En-Hui Yang. Bayes conditional distribution estimation for knowledge distillation based on conditional mutual information. *International Conference on Learning Representations (ICLR)*, 2024a.

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 (TDSC)*, 2024b.

Hongyao Yu, Yixiang Qiu, Hao Fang, Bin Chen, Sijin Yu, Bin Wang, Shutao Xia, and Ke Xu. Calor: Towards comprehensive model inversion defense. *ArXiv*, 2024a.

Wenbo Yu, Hao Fang, Bin Chen, Xiaohang Sui, Chuan Chen, Hao Wu, Shutao Xia, and Ke Xu. Gi-nas: Boosting gradient inversion attacks through adaptive neural architecture search. *ArXiv*, 2024b.

Xiaojian Yuan, Kejiang Chen, Jie Zhang, Weiming Zhang, Neng H. Yu, and Yangyi Zhang. Pseudo label-guided model inversion attack via conditional generative adversarial network. *AAAI Conference on Artificial Intelligence (AAAI)*, 2023.

Yuheng Zhang, R. Jia, Hengzhi Pei, Wenxiao Wang, Bo Li, and Dawn Xiaodong Song. The secret revealer: Generative model-inversion attacks against deep neural networks. *Computer Vision and Pattern Recognition (CVPR)*, 2019.

Xinhao Zhong, Bin Chen, Hao Fang, Xulin Gu, Shutao Xia, and En-Hui Yang. Going beyond feature similarity: Effective dataset distillation based on class-aware conditional mutual information. *ArXiv*, 2024a.

Xinhao Zhong, Hao Fang, Bin Chen, Xulin Gu, Tao Dai, Meikang Qiu, and Shutao Xia. Hierarchical features matter: A deep exploration of gan priors for improved dataset distillation. *ArXiv*, 2024b.

A Proof of Theorem 1

$$
\begin{aligned}
&\mathcal{I}(X;\hat{Y}|Y),\\
=&\sum_{y\in\mathbb{Y}}\mathcal{P}(y)\sum_{x\in\mathbb{X}}\sum_{\hat{y}\in\mathbb{Y}}\mathcal{P}(x,\hat{y}|y)\log\frac{\mathcal{P}(x,\hat{y}|y)}{\mathcal{P}(x|y)\mathcal{P}(\hat{y}|y)}, && \text{by definitions (3-4),}\\
=&\sum_{x\in\mathbb{X}}\sum_{\hat{y}\in\mathbb{Y}}\sum_{y\in\mathbb{Y}}\mathcal{P}(x,\hat{y},y)\log\frac{\mathcal{P}(\hat{y}|x,y)}{\mathcal{P}(\hat{y}|y)},\\
=&\sum_{x\in\mathbb{X}}\sum_{\hat{y}\in\mathbb{Y}}\sum_{y\in\mathbb{Y}}\mathcal{P}(x,\hat{y},y)\log\frac{\mathcal{P}(\hat{y}|x)}{\mathcal{P}(\hat{y}|y)}, && \text{by Markov chain } Y\to X\to\hat{Y},\\
=&\sum_{x\in\mathbb{X}}\sum_{\hat{y}\in\mathbb{Y}}\sum_{y\in\mathbb{Y}}\mathcal{P}(x,\hat{y},y)\log\left(\frac{\mathcal{P}(x,\hat{y})}{\mathcal{P}(x)}\Big/\frac{\mathcal{P}(\hat{y},y)}{\mathcal{P}(y)}\right),\\
=&\sum_{x\in\mathbb{X}}\sum_{\hat{y}\in\mathbb{Y}}\sum_{y\in\mathbb{Y}}\mathcal{P}(x,\hat{y},y)\log\left(\frac{\mathcal{P}(x,\hat{y})}{\mathcal{P}(x)\mathcal{P}(\hat{y})}\Big/\frac{\mathcal{P}(\hat{y},y)}{\mathcal{P}(\hat{y})\mathcal{P}(y)}\right),\\
=&\sum_{x\in\mathbb{X}}\sum_{\hat{y}\in\mathbb{Y}}\mathcal{P}(x,\hat{y})\log\frac{\mathcal{P}(x,\hat{y})}{\mathcal{P}(x)\mathcal{P}(\hat{y})}-\sum_{\hat{y}\in\mathbb{Y}}\sum_{y\in\mathbb{Y}}\mathcal{P}(\hat{y},y)\log\frac{\mathcal{P}(\hat{y},y)}{\mathcal{P}(\hat{y})\mathcal{P}(y)},\\
=&\mathcal{I}(X;\hat{Y})-\mathcal{I}(\hat{Y};Y), && \text{by definition (1).}
\end{aligned}
$$

B Water-Filling algorithm to solve (5)

For brevity, let $\boldsymbol{q} \coloneqq \boldsymbol{q}^y$, $\boldsymbol{f} \coloneqq \boldsymbol{f}(x)$, and $\varepsilon \coloneqq \varepsilon\cdot\mathcal{H}$. Then (5) is restated as

$$
\begin{aligned}
&\min\ \mathrm{KL}(\boldsymbol{p}||\boldsymbol{q}),\\
&\text{s.t. } \|\boldsymbol{p}-\boldsymbol{f}\|_1\le\varepsilon,\\
&\boldsymbol{p}\in\Delta^{\mathbb{Y}}.
\end{aligned}
\tag{6}
$$

Note that KL($\boldsymbol{p}||\boldsymbol{q}$) quantifies the difference between $\boldsymbol{p}$ and $\boldsymbol{q}$. When $\|\boldsymbol{q}-\boldsymbol{f}\|_1\le\varepsilon$, the optimal solution is $\boldsymbol{p}=\boldsymbol{q}$ trivially. When $\|\boldsymbol{q}-\boldsymbol{f}\|_1>\varepsilon$, the optimal $\boldsymbol{p}$ satisfies $\|\boldsymbol{p}-\boldsymbol{f}\|_1=\varepsilon$. We consider the case $\|\boldsymbol{q}-\boldsymbol{f}\|_1>\varepsilon$ in the following.

Obviously, the optimal $\boldsymbol{p}$ must be between $\boldsymbol{q}$ and $\boldsymbol{f}$, i.e.

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

Additionally, due to $\|\boldsymbol{p}-\boldsymbol{f}\|_1=\varepsilon$ and $\boldsymbol{p},\boldsymbol{f}\in\Delta^{\mathbb{Y}}$, the optimal $\boldsymbol{p}$ satisfies

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

Assume that $\{i\in\mathbb{Y}:q_i\ge f_i\}=\{1,2,\dots,n\}$ and $\{i\in\mathbb{Y}:q_i<f_i\}=\{n+1,n+2,\dots,|\mathbb{Y}|\}$. Then (6) can be divided into two sub-problems, (7) and (8).

$$
\begin{aligned}
&\min\ \sum_{i=1}^{n}p_i\log\frac{p_i}{q_i},\\
&\text{s.t. } \sum_{i=1}^{n}p_i-f_i=\frac{\varepsilon}{2},\\
&\quad p_i\ge f_i,\quad i=1,2,\dots,n.
\end{aligned}
\tag{7}
$$

$$
\begin{aligned}
&\min\ \sum_{i=n+1}^{|\mathbb{Y}|}p_i\log\frac{p_i}{q_i},\\
&\text{s.t. } \sum_{i=n+1}^{|\mathbb{Y}|}p_i-f_i=-\frac{\varepsilon}{2},\\
&\quad p_i\le f_i,\quad i=n+1,n+2,\dots,|\mathbb{Y}|.
\end{aligned}
\tag{8}
$$

To solve (7), we introduce Lagrange multipliers $\boldsymbol{\lambda}\in\mathbb{R}_+^n$ and $v\in\mathbb{R}$. The KKT conditions are

$$
\begin{aligned}
(p_i-f_i)\lambda_i&=0,\\
1+\log\frac{p_i}{q_i}-v-\lambda_i&=0.
\end{aligned}
$$

Eliminating $\lambda_i \geq 0$ yields

$$(p_i - f_i)\left(1 + \log\frac{p_i}{q_i} - v\right) = 0, \tag{9}$$

$$1 + \log\frac{p_i}{q_i} \geq v. \tag{10}$$

When $v > 1 + \log\frac{f_i}{q_i}$, (10) implies $p_i > f_i$, and then (9) implies $p_i = q_i \exp(v-1)$.

When $v \leq 1 + \log\frac{f_i}{q_i}$, $p_i > f_i$ implies $\left(1 + \log\frac{p_i}{q_i} - v\right) > 0$ contradicting (9), so $p_i = f_i$.

In summary,

$$p_i = \begin{cases} q_i \exp(v-1), & v > 1 + \log\dfrac{f_i}{q_i}, \\ f_i, & \text{other.} \end{cases}$$

Let $w \coloneqq \exp(v-1)$ and the optimal solution is

$$p_i = \max(wq_i, f_i), \quad i = 1, 2, \dots, n, \tag{11}$$

where w satisfies $\sum_{i=1}^{n} p_i - f_i = \frac{\varepsilon}{2}$.

Solving (8) similarly, the optimal solution is

$$p_i = \min(wq_i, f_i), \quad i = n+1, n+2, \dots, |\mathbb{Y}|,$$

where w satisfies $\sum_{i=n+1}^{|\mathbb{Y}|} p_i - f_i = -\frac{\varepsilon}{2}$.

We propose Algorithm 2 to calculate (11) efficiently. We ensure $\frac{f_1}{q_1} \leq \frac{f_2}{q_2} \leq \dots \leq \frac{f_n}{q_n}$ by sorting, and increase w to $\frac{f_1}{q_1}, \frac{f_2}{q_2}, \dots, \frac{f_n}{q_n}$ sequentially. Once $\sum_{i=1}^{j-1} wq_i - f_i > \frac{\varepsilon}{2}$ when $w = \frac{f_j}{q_j}$, we know that the proper $w \in [\frac{f_{j-1}}{q_{j-1}}, \frac{f_j}{q_j})$ and $w = \frac{\frac{\varepsilon}{2} + \sum_{i=1}^{j-1} f_i}{\sum_{i=1}^{j-1} q_i}$. Our algorithm is known as *Water-Filling*, because w is like a rising water level, $\frac{f_1}{q_1}, \frac{f_2}{q_2}, \dots, \frac{f_n}{q_n}$ are like ascending steps, and $\frac{\varepsilon}{2}$ is like the total volume of water. The time complexity is $O(n \log n)$ due to the sorting.

Algorithm 2: CPU-based Water-Filling.

```
Input: $q_i, f_i$ for $i = 1, 2, \dots, n$.
Output: $p_i$ for $i = 1, 2, \dots, n$.
Reindex $q_i, f_i$ so that $\frac{f_1}{q_1} \leq \frac{f_2}{q_2} \leq \dots \leq \frac{f_n}{q_n}$;
$j \leftarrow 1$;
$f_{\text{sum}} \leftarrow 0$;
$q_{\text{sum}} \leftarrow 0$;
while $\frac{f_j}{q_j} q_{sum} - f_{sum} \leq \frac{\varepsilon}{2}$ do
    $f_{\text{sum}} \leftarrow f_{\text{sum}} + f_j$;
    $q_{\text{sum}} \leftarrow q_{\text{sum}} + q_j$;
    $j \leftarrow j + 1$;
end
$w \leftarrow \dfrac{\frac{\varepsilon}{2} + f_{\text{sum}}}{q_{\text{sum}}}$;
Reindex $q_i, f_i$ back to the original;
return $\max(wq_i, f_i)$ for $i = 1, 2, \dots, n$;
```

Algorithm 3: GPU-based Water-Filling.

```
Input: PyTorch tensors $\boldsymbol{q}, \boldsymbol{f}$ of size $n$.
Output: PyTorch tensor $\boldsymbol{p}$ of size $n$.
Reindex $\boldsymbol{q}, \boldsymbol{f}$ by torch.sort($\frac{\boldsymbol{f}}{\boldsymbol{q}}$);

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


$j \leftarrow$ mask.argmin();

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

To further accelerate Algorithm 2, we propose Algorithm 3, a GPU-based Water-Filling. Using operators provided by PyTorch, we manage to eliminate the loop and branch in Algorithm 2, making it completely sequential and suitable for GPUs. Algorithm 3 fully leverages the parallelism of GPUs and reduces the computation cost, which is quantitatively described in the next section.

C EXPERIMENTS ON COMPUTATION COST

We quantitatively demonstrate the efficiency of our Algorithm 1 by experiments. The experiment settings are consistent with the main text. 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 4.

Table 4: Computation cost of Algorithm 1.

	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 $\boldsymbol{s} \in \mathbb{R}^{|\mathbb{Y}|} \sim N(\mathbf{0}, \boldsymbol{I})$ and let $\boldsymbol{r} \leftarrow \textbf{softmax}(10 \cdot \boldsymbol{s})$. 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 $\boldsymbol{f}(x)$ 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 $(\boldsymbol{f}(x), \boldsymbol{q}^y)$ and solve in parallel. The time costs are shown in Table 5.

Table 5: The relationship between $|\mathbb{Y}|$ and time cost of our Algorithm 3.

$\lvert\mathbb{Y}\rvert$	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.

D THE HYPERPARAMETERS FOR EACH DEFENSE

Table 6: The hyperparameters for each defense.

Defense	IR-152 & CelebA	IR-152 & FaceScrub	VGG-16 & FaceScrub
MID	$\beta = 0.005$	$\beta = 0.02$	$\beta = 0.015$
BiDO	$\lambda_x = 0.001, \lambda_y = 0.01$	$\lambda_x = 0.005, \lambda_y = 0.05$	$\lambda_x = 0.0005, \lambda_y = 0.005$
LS	$\alpha = -0.05$	$\alpha = -0.05$	$\alpha = -0.05$
TL	Freeze the first 50% of the layers.		
SSD	$T = 0.1, \varepsilon = 0.35$	$T = 0.1, \varepsilon = 0.5$	$T = 0.3, \varepsilon = 0.8$

E EXPERIMENT UNDER RLB ATTACK

We conduct the experiment under RLBMI attack (Han et al., 2023) in this section. Consistent with the main text, the target models are IR-152 trained on CelebA. Since RLBMI is computationally expensive, we only attack the first 10 labels and reconstruct 5 images for each label. The results are shown in Table 7.

Table 7: MIA resistance of various defenses under RLBMI attack.

	$\downarrow$ acc	$\downarrow$ acc5	$\uparrow \sigma_{eval}$	$\uparrow \sigma_{face}$
None	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	**8%**	**12%**	**2480**	**1.26**

It can be seen that our SSD is superior to other defenses.

F 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 and resized to 224×224. The target models are ResNet-152 trained on FaceScrub, and the evaluation model is an Inception-v3. Since high resolution is computationally expensive, we only attack the first 10 labels and reconstruct 5 images for each label. The attack results are shown in Table 8 and the models' utility and settings are shown in Table 9. It can be seen that our SSD still achieves the best MIA robustness, with a good utility.

Table 8: MIA resistance of various defenses under high-resolution Mirror attack.

	$\downarrow$ acc	$\downarrow$ acc5	$\uparrow \sigma_{eval}$	$\uparrow \sigma_{face}$
None	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	**42%**	**66%**	**211**	**1.13**

Table 9: Model's utility with various defenses.

	$\uparrow$ acc	$\downarrow$ dist	Settings
None	98.5%	0	–
MID	96.7%	0.30	$\beta = 0.005$
BiDO	96.3%	0.09	$\lambda_x = 0.15, \lambda_y = 1.5$
LS	96.5%	0.11	$\alpha = -0.01$
TL	96.7%	0.19	First 70% layers
SSD	**97.0%**	**0.05**	$T = 0.3, \varepsilon = 1$

G DISCUSSION ON ADAPTIVE ATTACKS

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

In our opinion, if attackers know our defense, their best strategy is:

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

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 x, we take $h(x)$ as the random seed for sampling, ensuring same-input-same-output. However, attackers can add subtle perturbations to x, therefore our h needs to be robust. For example, it can be

$$h(x) \coloneqq \sum_{i=1}^{m} \lfloor k \cdot z_i(x) \rfloor, \tag{12}$$

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