STEALTHY SHIELD DEFENSE: A CONDITIONAL MUTUAL INFORMATION-BASED APPROACH 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**

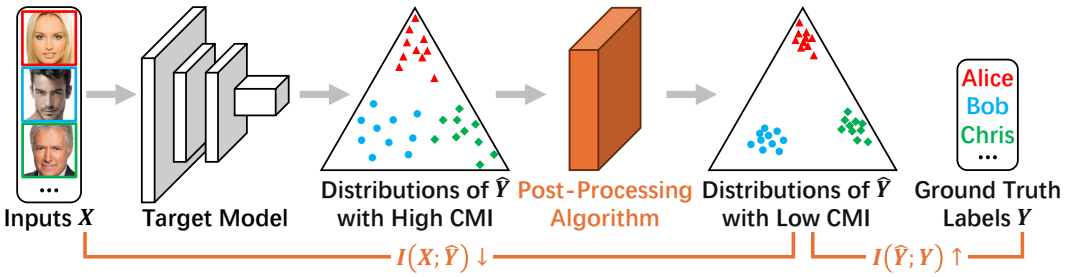


Figure 1: An overview of Stealthy Shield Defense. The probability simplex is a triangle when the number of classes is three. CMI is defined as $\mathcal{I}(X;\hat{Y}|Y)$. According to our Theorem 1, minimizing CMI makes the mutual information $\mathcal{I}(X;\hat{Y})$ minimized and $\mathcal{I}(\hat{Y};Y)$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

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.

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 Works

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 Notations

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 $\mathcal{P}(x) \coloneqq \mathcal{P}\{X = x\}$, $\mathcal{P}(y) \coloneqq \mathcal{P}\{Y = y\}$, $\mathcal{P}(x, \hat{y}|y) \coloneqq \mathcal{P}\{X = x, \hat{Y} = \hat{y} \mid Y = y\}$, etc.

Let $\Delta^{\mathbb{Y}}$ be the probability simplex over $\mathbb{Y}$. 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 $\boldsymbol{f}(x)$, $\hat{y} \in \mathbb{Y}$.

3.2 Model Inversion Attacks

Let $D \subseteq \mathbb{X} \times \mathbb{Y}$ be the dataset learned by f. 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 $f(x) \in \mathbb{Y}$, i.e. $\operatorname{argmax}_{\hat{y} \in \mathbb{Y}} f_{\hat{y}}(x)$.

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

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

They reduced $\mathcal{I}(X; \hat{Y})$ to prevent attackers from inferring the information about 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

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

where $\beta > 0$. They used (2) 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 during attacking. Formally, let $D^y \coloneqq \{x \in \mathbb{X} : (x,y) \in D\}$ be the sub-dataset with the ground truth label y. When y is given, 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 about 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 information bottlenecks (2) when $Z = \hat{Y}$ and $\beta = 1$, 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 MIA resistance 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 model's utility obviously.

The $\mathcal{I}(X;Z)$ in IB is challenging to calculate because the input space $\mathbb{X}$ and representation space $\mathbb{Z}$ are both high-dimensional. Wang et al. (2020) 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 works used CMI as a regularizer and minimized it during training (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 manage to calculate $\mathcal{P}(\hat{y}|x)$, $\mathcal{P}(y|x)$ 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 its inaccuracy in modern neural classifiers. Inspired by their work, we introduce the temperature mechanism to adjust it.

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

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

where the $\approx$ is by 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 the Kullback-Leibler divergence. 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), which is for signal compression. If a signal has less information, it is easier to compress, and a stricter distortion bound can be applied. Inspired by his 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 constraint becomes $\|\boldsymbol{p} - \boldsymbol{f}(x)\|_1 \le \varepsilon \cdot \mathcal{H}(\hat{Y}|X = x)$, 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 within $O(|\mathbb{Y}| \log |\mathbb{Y}|)$ 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 ε, validation set D.
Output: modified output $\boldsymbol{p}^*$.

Sample $y \in \mathbb{Y}$ with the probability of $\textbf{softmax}(\frac{\boldsymbol{f}(x)}{T})$;
$\boldsymbol{q}^y \leftarrow \underset{x' \in D^y}{\textbf{mean}}\, \boldsymbol{f}(x')$;
$\mathcal{H} \leftarrow -\sum_{\hat{y} \in \mathbb{Y}} f_{\hat{y}}(x) \log f_{\hat{y}}(x)$;
Solve the convex optimization problem and return the optimal solution:

$$\begin{aligned} \min \quad & \mathrm{KL}(\boldsymbol{p} || \boldsymbol{q}^y), \\ \text{s.t.} \quad & \|\boldsymbol{p} - \boldsymbol{f}(x)\|_1 \leq \varepsilon \cdot \mathcal{H}, \\ & \boldsymbol{p} \in \Delta^{\mathbb{Y}}. \end{aligned} \tag{5}$$

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

Note that $D' \coloneqq \{\boldsymbol{q}^y : y \in \mathbb{Y}\}$ can be calculated and stored in advance. If the model owner and the defender are not the same, the owner only needs to provide D' instead of D, avoiding communication costs 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 within $O(|\mathbb{Y}| \log |\mathbb{Y}|)$ time in Algorithm 2, accelerate it via GPUs in Algorithm 3, and evaluate the computation cost in Appendix C.

5 Experiments

5.1 Settings

Datasets. We select CelebA (Liu et al., 2015) and FaceScrub (Ng & Winkler, 2014) as private datasets, and FFHQ (Karras et al., 2018) as public dataset. CelebA has 10,177 labels and we only take 1,000 labels with the most images. FaceScrub has 530 labels and 106,863 images, but we only take 43,147 images because the other URLs are broken. FFHQ has 70,000 unlabeled images. All images are cropped and resized to 64×64. We use 80% of the private data for training, 10% for validation, and 10% for testing.

Models. We select IR-152 (He et al., 2015) and VGG-16 (Simonyan & Zisserman, 2014) as target models, and MaxViT (Tu et al., 2022) as evaluation model. IR-152 and MaxViT are pre-trained on MS-Celeb-1M (Guo et al., 2016), and VGG-16 is pre-trained on ImageNet (Deng et al., 2009). They are fine-tuned for 100 epochs on the training set, and we take the version with the highest accuracy on the validation set. The evaluation models achieve 97.3% test accuracy on CelebA and 99.3% on FaceScrub.

Attacks. We select Mirror (An et al., 2022) and C2FMI (Ye et al., 2024b) as soft-label attackers, and BREP (Kahla et al., 2022) and LOKT (Nguyen et al., 2023) as hard-label attackers. They attack the first 100 private labels and reconstruct 5 images for each label. For BREP and LOKT, we train GANs and surrogate models on FFHQ. For Mirror and C2FMI, we use the 256×256 StyleGAN2 (Karras et al., 2019) trained on FFHQ. The generated images are center-cropped to 176×176 and resized to 64×64.

Defenses. We select MID (Wang et al., 2020), BiDO (Peng et al., 2022), LS (Struppek et al., 2023) and TL (Ho et al., 2024) as competitors. They need to retrain the target model, whereas we only post-process the outputs from the undefended model. For fair comparison, we carefully tune their hyper-parameters to achieve similar accuracies on the validation set. All hyper-parameters are in Appendix D.

To evaluate the MIA resistance and model's utility, we consider the following metrics:

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

		Mirror				C2FMI			
		$\downarrow Acc1$	$\downarrow Acc5$	$\uparrow \delta_{eval}$	$\uparrow \delta_{face}$	$\downarrow Acc1$	$\downarrow Acc5$	$\uparrow \delta_{eval}$	$\uparrow \delta_{face}$
IR-152 CelebA	**None**	18.0%	31.0%	625	1.22	5.8%	15.4%	647	1.30
	MID	17.8%	31.6%	629	1.22	**1.2%**	4.2%	699	1.48
	BiDO	10.6%	25.8%	614	1.16	5.4%	13.0%	645	1.33
	LS	9.0%	15.2%	660	1.36	1.8%	8.0%	660	1.36
	TL	13.6%	28.6%	633	1.24	2.8%	5.0%	703	1.48
	SSD	**3.2%**	**8.2%**	**728**	**1.46**	**1.2%**	**1.8%**	**744**	**1.59**
IR-152 FaceScrub	**None**	55.2%	76.8%	496	0.89	19.6%	41.4%	549	1.02
	MID	38.0%	61.0%	534	0.96	**0.2%**	**1.8%**	**715**	**1.44**
	BiDO	34.4%	60.4%	526	0.93	13.2%	27.6%	588	1.13
	LS	45.8%	73.4%	503	0.90	13.6%	31.2%	564	1.05
	TL	39.2%	63.2%	535	0.99	7.0%	18.6%	617	1.22
	SSD	**32.2%**	**46.4%**	**604**	**1.15**	1.6%	5.6%	671	1.35
VGG-16 FaceScrub	**None**	29.0%	51.8%	544	1.00	12.8%	30.6%	588	1.11
	MID	34.6%	64.8%	520	0.94	4.4%	17.8%	611	1.19
	BiDO	20.0%	44.4%	556	1.03	12.0%	27.6%	575	1.09
	LS	30.2%	56.4%	531	0.97	15.6%	37.4%	551	1.05
	TL	17.8%	41.8%	558	1.03	8.4%	27.2%	574	1.10
	SSD	**16.4%**	**40.0%**	**604**	**1.13**	**3.2%**	**13.6%**	**656**	**1.29**

Attack Accuracy Let the evaluation model reclassify the reconstructed images. The top-1 and top-5 accuracies are denoted as $Acc1$ and $Acc5$ respectively. Lower percentages indicate better MIA resistance.

Feature Distance The image features are extracted from the penultimate layer of a model. We take the average L_2 distance between the reconstructed images and the nearest private images. The features are extracted by the evaluation model and a FaceNet (Schroff et al., 2015) trained on VGGFace2 (Cao et al., 2017), which are denoted as δ_{eval} and δ_{face} respectively. Higher distances indicate better MIA resistance.

Test Accuracy The accuracy of the target model on the test set, which is denoted as Acc. Higher percentage indicates better utility.

Distortion The L_1 distance between the probability vectors with and without defense. We take the average on the test set, which is denoted as $Dist$. Lower distance indicates better utility.

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

5.2 Comparisons with State-of-the-art Defenses

The evaluation results under soft-label attacks are in Table 1, and the ones under hard-label attacks are in Table 2. Our SSD exhibits the best MIA resistance and outperforms all state-of-the-art defenses, across various attack algorithms, private datasets, and model architectures. This is because we minimize the CMI to prevent attackers from inferring the information about the private dataset.

To our surprise, some metrics with defenses are worse than those with no defense. It demonstrates that existing defenses may fail to resist modern black-box attacks. They focus on modifying the weights and structure of the model, but black-box attackers only exploit the outputs and thus are rarely hindered. This highlights the necessity of designing specialized black-box defenses.

The visualization results are in Figure 2. Our SSD significantly enhances the difference between D^y and $\hat{D}^y$, forces attackers to reconstruct wrong images, and thus protects the private data. In contrast, other defenses fail to prevent attackers from reconstructing similar images, leading to privacy leakage.

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

		BREP				LOKT			
		$\downarrow Acc1$	$\downarrow Acc5$	$\uparrow \delta_{eval}$	$\uparrow \delta_{face}$	$\downarrow Acc1$	$\downarrow Acc5$	$\uparrow \delta_{eval}$	$\uparrow \delta_{face}$
IR-152 CelebA	**None**	26.8%	50.8%	541	1.02	50.2%	67.8%	527	0.93
	MID	20.8%	39.4%	596	1.13	32.4%	52.2%	551	1.00
	BiDO	11.4%	33.4%	577	1.09	30.8%	48.4%	576	1.06
	LS	19.4%	43.8%	547	1.03	33.6%	57.0%	537	1.01
	TL	20.2%	46.4%	550	1.05	43.0%	66.2%	525	0.93
	SSD	**1.8%**	**4.2%**	**769**	**1.62**	**1.2%**	**9.0%**	**769**	**1.61**
IR-152 FaceScrub	**None**	50.6%	77.2%	497	0.95	90.2%	96.2%	411	0.65
	MID	36.2%	52.6%	570	1.10	61.8%	81.4%	475	0.78
	BiDO	33.6%	62.6%	530	1.02	82.2%	94.8%	445	0.69
	LS	36.0%	67.0%	521	1.01	84.8%	95.4%	429	0.69
	TL	30.4%	59.0%	538	1.06	91.6%	98.4%	414	0.65
	SSD	**9.2%**	**13.4%**	**717**	**1.47**	**15.8%**	**25.0%**	**656**	**1.27**
VGG-16 FaceScrub	**None**	28.2%	54.0%	543	1.06	75.8%	91.4%	444	0.74
	MID	30.0%	56.8%	535	1.03	63.2%	86.8%	470	0.80
	BiDO	21.6%	44.4%	566	1.10	67.4%	87.4%	474	0.80
	LS	24.0%	50.4%	549	1.05	74.4%	90.6%	453	0.79
	TL	20.8%	46.8%	566	1.08	67.4%	88.8%	471	0.79
	SSD	**10.8%**	**18.6%**	**693**	**1.40**	**24.0%**	**35.4%**	**635**	**1.24**

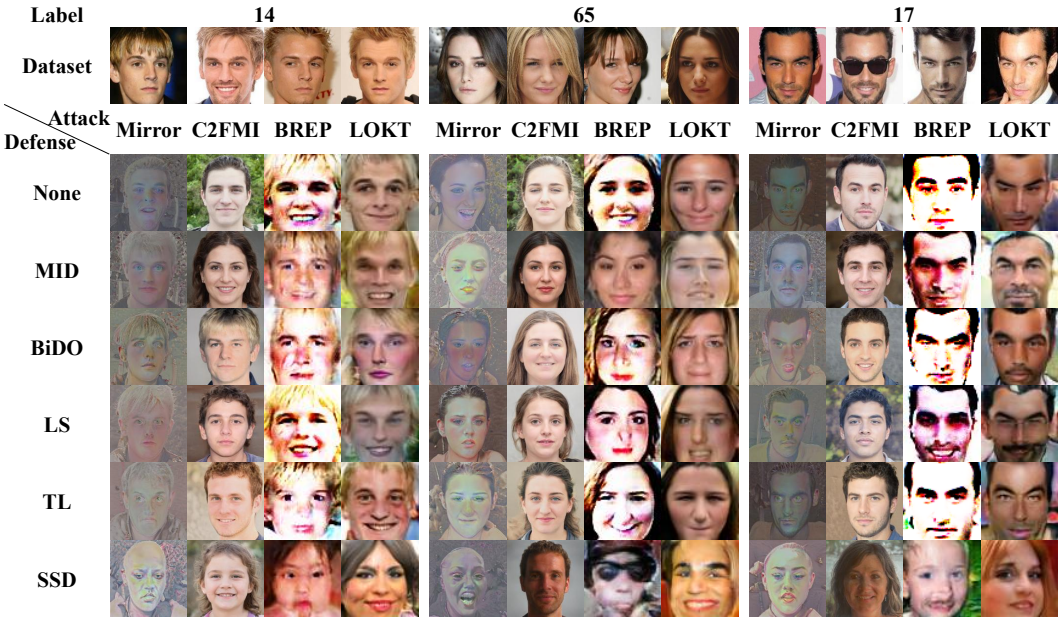


Figure 2: The reconstructed images from IR-152 models trained on CelebA. Above are the ground truth labels y and the private sub-datasets D^y (4 images shown). Below are the reconstructed datasets $\hat{D}^y$ (1 image shown), over various attacks and defenses.

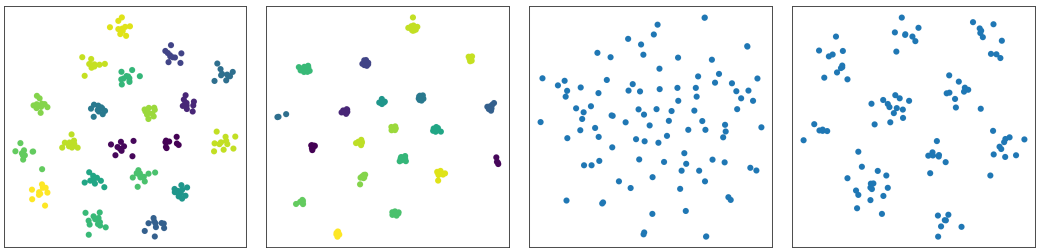
Table 3: Model's utility with various defenses.

	IR-152 & CelebA		IR-152 & FaceScrub		VGG-16 & FaceScrub	
	$\uparrow Acc$	$\downarrow Dist$	$\uparrow Acc$	$\downarrow Dist$	$\uparrow Acc$	$\downarrow Dist$
None	92.1%	0	98.2%	0	91.8%	0
MID	86.8%	0.607	95.5%	0.321	86.4%	0.744
BiDO	86.6%	0.371	95.3%	0.135	88.5%	0.313
LS	86.9%	0.317	95.6%	0.103	87.5%	0.295
TL	86.5%	0.352	95.8%	0.128	87.4%	0.306
SSD	**87.1%**	**0.191**	**97.0%**	**0.055**	**89.3%**	**0.176**

The evaluation results on model's utility are in Table 3. Our SSD achieves the highest test accuracy, because we minimize CMI to preserve the information useful for tasks. Additionally, we hold the smallest distortion due to our distortion controller ε and adaptive mechanism.

5.3 Visualize the outputs of SSD

To explore how SSD modifies the outputs, we visualize them using t-SNE (van der Maaten & Hinton, 2008). Figure 3 shows that minimizing CMI makes the outputs aggregate into several clusters, which is interpreted by Yang et al. (2024). For private samples, this will reduce the sensitivity between the inputs and outputs, preventing attackers from inferring the private information. For attack samples, this will mislead attackers towards the wrong label.



(a) Tested w/o defense. (b) Tested with SSD. (c) Attacked w/o defense. (d) Attacked with SSD.

Figure 3: Visualizing the outputs using t-SNE. The target model is a VGG-16 model trained on FaceScrub. For (a) and (b), the test samples are colored according to their ground truth labels. For (c) and (d), the attack samples are generated by Mirror. Only a few clusters are shown due to space limitations.

5.4 Ablation Study

To explore the effects of temperature T and distortion controller ε in SSD, we conduct ablation experiments.

Figure 4(a) shows that a higher temperature helps to resist hard-label attacks, because it makes the sampling probability more uniform, and attackers easier to get misleading labels. However, a higher temperature decreases the test accuracy, which is shown in Figure 4(b). Specifically, without the temperature mechanism, neither the MIA resistance nor model's accuracy is satisfactory, which highlights the necessity of our temperature mechanism.

Figure 4(c) shows that a larger distortion bound helps to resist soft-label attacks, because it allows greater modifications. However, a larger distortion bound decreases the test accuracy and increases the distortion, which is shown in Figure 4(d). Specifically, without the adaptive mechanism, neither the face distance nor the distortion is satisfactory, which highlights the necessity of our adaptive rate-distortion.

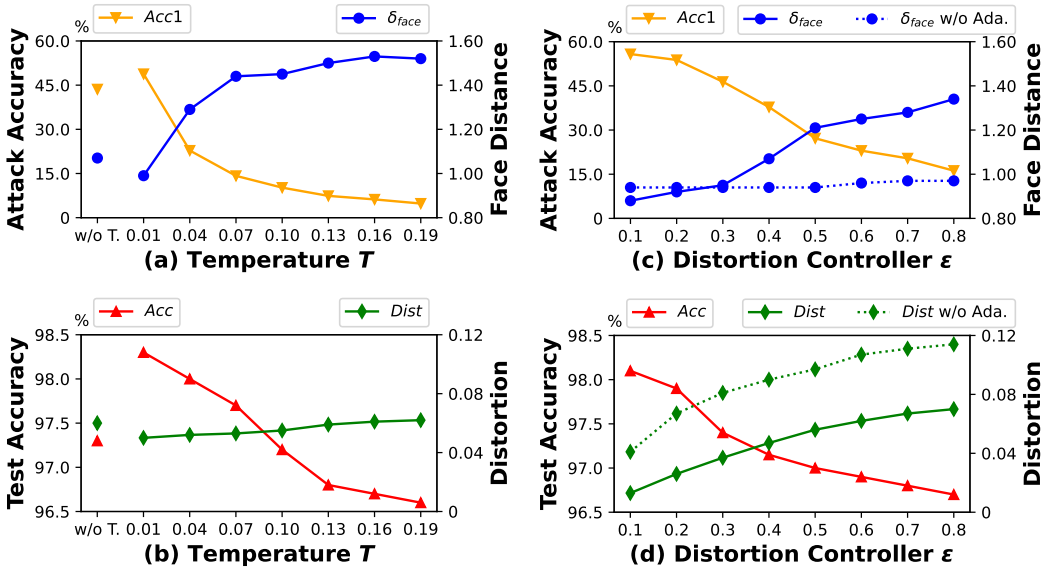


Figure 4: Ablation results of IR-152 models trained on FaceScrub. (a) is attacked under BREP, and (c) is attacked under Mirror. The "w/o T." denotes "without temperature mechanism", and "w/o Ada." denotes "without adaptive rate-distortion".

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.

Qiong Cao, Li Shen, Weidi Xie, Omkar M. Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *Automatic Face & Gesture Recognition (FG)*, 2017.

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

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition (CVPR)*, 2009.

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.

A. Gionis, Piotr Indyk, and Rajeev Motwani. Similarity search in high dimensions via hashing. In *Very Large Data Bases Conference (VLDB)*, 1999.

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

Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *European Conference on Computer Vision (ECCV)*, 2016.

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, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Computer Vision and Pattern Recognition (CVPR)*, 2018.

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)*, 2015.

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.

Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Conrad Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. In *European Conference on Computer Vision (ECCV)*, 2022.

Laurens van der Maaten and Geoffrey E. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research (JMLR)*, 2008.

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.

Ziqi Yang, Li-Juan Wang, D. Yang, Jie Wan, Ziming Zhao, Ee-Chien Chang, Fan Zhang, and Kui Ren. Purifier: Defending data inference attacks via transforming confidence scores. *AAAI Conference on Artificial Intelligence (AAAI)*, 2023.

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

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

Let $\boldsymbol{p}^*$ be the optimal solution. Note that $\mathrm{KL}(\boldsymbol{p}||\boldsymbol{q})$ is a convex function with a minimum point at $\boldsymbol{p}=\boldsymbol{q}$. We have $\boldsymbol{p}^*=\boldsymbol{q}$ if $\|\boldsymbol{q}-\boldsymbol{f}\|_1\le\varepsilon$, and $\|\boldsymbol{p}^*-\boldsymbol{f}\|_1=\varepsilon$ otherwise.

Consider $\|\boldsymbol{q}-\boldsymbol{f}\|_1>\varepsilon$ in the following. Since $\boldsymbol{p}^*,\boldsymbol{f}\in\Delta^{\mathbb{Y}}$, we have

$$
\sum_{i\in\mathbb{Y}:\ p_i^*\ge f_i}|p_i^*-f_i|=\sum_{i\in\mathbb{Y}:\ p_i^*<f_i}|p_i^*-f_i|=\frac{\varepsilon}{2}.
\tag{7}
$$

Lemma 1. *$\forall i\in\mathbb{Y}$, either $q_i\ge p_i^*\ge f_i$ or $q_i\le p_i^*\le f_i$.*

Proof. WLOG, assume $\exists i\in\mathbb{Y}, p_i^*>q_i\ge f_i$. Since $\boldsymbol{p}^*,\boldsymbol{q}\in\Delta^{\mathbb{Y}}$, there must $\exists j\in\mathbb{Y}, p_j^*<q_j$. Let

$$
g(\xi)\coloneqq(p_i^*-\xi)\log\frac{p_i^*-\xi}{q_i}+(p_j^*+\xi)\log\frac{p_j^*+\xi}{q_j}.
$$

Note that $g'(\xi)$ is continuous and $g'(0)=\log\frac{p_j^*}{q_j}-\log\frac{p_i^*}{q_i}<0$. So $\exists\xi^*>0$, $\forall\xi\in(0,\xi^*)$, $g(\xi)<g(0)$. It implies that decreasing p_i^* by ξ and increasing p_j^* by ξ will reduce $\mathrm{KL}(\boldsymbol{p}^*||\boldsymbol{q})$, contradicting the optimality of $\boldsymbol{p}^*$. No constraints are violated if ξ is chosen such that $p_i^*-\xi\ge f_i$ and $p_j^*+\xi\le 1$. □

Let $A\coloneqq\{i\in\mathbb{Y}:q_i\ge f_i\}$ and $B\coloneqq\{i\in\mathbb{Y}:q_i<f_i\}$. Based on (7) and Lemma 1, we divide (6) into two sub-problems, (8) and (9).

$$
\begin{aligned}
\min\quad & \sum_{i\in A}p_i\log\frac{p_i}{q_i},\\
\text{s.t.}\quad & \sum_{i\in A}p_i-f_i=\frac{\varepsilon}{2},\\
& q_i\ge p_i\ge f_i,\qquad i\in A.
\end{aligned}
\tag{8}
$$

$$
\begin{aligned}
\min\quad & \sum_{i\in B}p_i\log\frac{p_i}{q_i},\\
\text{s.t.}\quad & \sum_{i\in B}p_i-f_i=-\frac{\varepsilon}{2},\\
& q_i\le p_i\le f_i,\qquad i\in B.
\end{aligned}
\tag{9}
$$

To solve (8), we introduce Lagrange multipliers $v^* \in \mathbb{R}$ for $\sum_{i\in A} p_i^* - f_i = \frac{\varepsilon}{2}$, and $\lambda_i^* \geq 0$ for $p_i^* \geq f_i, i \in A$. The KKT conditions are

$$1 + \log \frac{p_i^*}{q_i} - v^* - \lambda_i^* = 0,$$
$$(p_i^* - f_i)\lambda_i^* = 0.$$

Eliminating $\lambda_i^* \geq 0$ yields

$$1 + \log \frac{p_i^*}{q_i} \geq v^*, \tag{10}$$
$$(p_i^* - f_i)\left(1 + \log \frac{p_i^*}{q_i} - v^*\right) = 0. \tag{11}$$

If $1 + \log \frac{f_i}{q_i} \geq v^*$, then $p_i^* > f_i$ implies $\left(1 + \log \frac{p_i^*}{q_i} - v^*\right) > 0$ contradicting (11), so $p_i^* = f_i$.

If $1 + \log \frac{f_i}{q_i} < v^*$, then (10) implies $p_i^* > f_i$ and (11) implies $p_i^* = q_i \exp(v^* - 1)$.

In summary,

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

Let $w_A^* \coloneqq \exp(v^* - 1)$ and it becomes

$$p_i^* = \max(f_i, w_A^* q_i), \quad i \in A, \tag{12}$$

where w_A^* satisfies $\sum_{i\in A} p_i^* - f_i = \frac{\varepsilon}{2}$. Let $c(w) \coloneqq \sum_{i\in A} \max(0, wq_i - f_i)$, which is a piecewise-linear increasing function with breakpoints at $w = \frac{f_i}{q_i}, i \in A$. Note that $c(0) = 0$ and $c(1) = \frac{\|\boldsymbol{q}-\boldsymbol{f}\|_1}{2} > \frac{\varepsilon}{2}$, so $\exists! w_A^* \in (0, 1), c(w_A^*) = \frac{\varepsilon}{2}$. The constraint $q_i \geq p_i^*, i \in A$ is naturally satisfied by $w_A^* < 1$.

Solving (9) similarly yields

$$p_i^* = \min(f_i, w_B^* q_i), \quad i \in B, \tag{13}$$

where w_B^* satisfies $\sum_{i\in B} p_i^* - f_i = -\frac{\varepsilon}{2}$ and $w_B^* \in (1, +\infty)$.

Combining (12-13) and $w_A^* < 1 < w_B^*$ yields

$$p_i^* = \min(\max(f_i, w_A^* q_i), w_B^* q_i), \quad i \in \mathbb{Y}.$$

We propose Algorithm 2 to solve (6) efficiently. Firstly we address the trivial cases. For non-trivial cases, we make $\frac{f_1}{q_1} \leq \frac{f_2}{q_2} \leq \ldots \leq \frac{f_{|\mathbb{Y}|}}{q_{|\mathbb{Y}|}}$ by sorting. To determine w_A^*, we check $w = \frac{f_1}{q_1}, \frac{f_2}{q_2}, \ldots$ successively. Once $c(w) > \frac{\varepsilon}{2}$ when $w = \frac{f_j}{q_j}$, we get $w_A^* = \frac{\frac{\varepsilon}{2} + \sum_{i=1}^{j-1} f_i}{\sum_{i=1}^{j-1} q_i} \in [\frac{f_{j-1}}{q_{j-1}}, \frac{f_j}{q_j})$. This process is known as *water-filling*, because w is like a rising water level, $\frac{f_1}{q_1}, \frac{f_2}{q_2}, \ldots$ are like ascending steps, and $\frac{\varepsilon}{2}$ is like the total volume of water. To determine w_B^*, we check $w = \frac{f_{|\mathbb{Y}|}}{q_{|\mathbb{Y}|}}, \frac{f_{|\mathbb{Y}|-1}}{q_{|\mathbb{Y}|-1}}, \ldots$ similarly. This symmetric process is known as *reverse water-filling*. Finally we restore the indices and output the optimal solution. Our time complexity is $O(|\mathbb{Y}| \log |\mathbb{Y}|)$ mainly due to the sorting.

We propose Algorithm 3 to accelerate Algorithm 2 by GPUs. Leveraging the tensor operators provided by PyTorch, we manage to eliminate the loops and branches in Algorithm 2, making it completely sequential and suitable for GPUs. Algorithm 3 can process numerous $(\boldsymbol{f}, \boldsymbol{q})$ pairs in parallel and reduce the computation cost, which is quantitatively described in the next section.

Algorithm 2: CPU-based water-filling.

Input: f_i, q_i for $i \in \mathbb{Y}$.
Output: p_i^* for $i \in \mathbb{Y}$.
if $\sum_{i\in\mathbb{Y}} |q_i - f_i| \leq \varepsilon$ **then**
  return q_i for $i \in \mathbb{Y}$;
Reindex f_i, q_i so that $\frac{f_1}{q_1} \leq \frac{f_2}{q_2} \leq \ldots \leq \frac{f_{|\mathbb{Y}|}}{q_{|\mathbb{Y}|}}$;
$j \leftarrow 1$;
$F \leftarrow 0$;
$Q \leftarrow 0$;
while $\frac{f_j}{q_j} Q - F \leq \frac{\varepsilon}{2}$ **do**
  $F \leftarrow F + f_j$;
  $Q \leftarrow Q + q_j$;
  $j \leftarrow j + 1$;
$w_A^* \leftarrow \dfrac{F + \frac{\varepsilon}{2}}{Q}$;
$j \leftarrow |\mathbb{Y}|$;
$F \leftarrow 0$;
$Q \leftarrow 0$;
while $\frac{f_j}{q_j} Q - F \geq -\frac{\varepsilon}{2}$ **do**
  $F \leftarrow F + f_j$;
  $Q \leftarrow Q + q_j$;
  $j \leftarrow j - 1$;
$w_B^* \leftarrow \dfrac{F - \frac{\varepsilon}{2}}{Q}$;
Restore the indices of f_i, q_i;
return
 $\min(\max(f_i, w_A^* q_i), w_B^* q_i)$ for $i \in \mathbb{Y}$;

Algorithm 3: GPU-based water-filling.

Input: PyTorch tensors $\boldsymbol{f}, \boldsymbol{q}$ of size $|\mathbb{Y}|$.
Output: PyTorch tensor $\boldsymbol{p}^*$ of size $|\mathbb{Y}|$.
$m \leftarrow (\|\boldsymbol{q} - \boldsymbol{f}\|_1 \leq \varepsilon)$;

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

$\boldsymbol{F} \leftarrow \boldsymbol{f}$.cumsum();
$\boldsymbol{Q} \leftarrow \boldsymbol{q}$.cumsum();
$\boldsymbol{M} \leftarrow (\frac{\boldsymbol{f}}{\boldsymbol{q}} \boldsymbol{Q} - \boldsymbol{F} \leq \frac{\varepsilon}{2})$;

$j \leftarrow \boldsymbol{M}$.argmin();
$w_A^* \leftarrow \dfrac{\boldsymbol{F}[j] + \frac{\varepsilon}{2}}{\boldsymbol{Q}[j]}$;

$\boldsymbol{F} \leftarrow 1 - \boldsymbol{F}$;
$\boldsymbol{Q} \leftarrow 1 - \boldsymbol{Q}$;
$\boldsymbol{M} \leftarrow (\frac{\boldsymbol{f}}{\boldsymbol{q}} \boldsymbol{Q} - \boldsymbol{F} \geq -\frac{\varepsilon}{2})$;

$j \leftarrow \boldsymbol{M}$.argmax();
$w_B^* \leftarrow \dfrac{\boldsymbol{F}[j] - \frac{\varepsilon}{2}}{\boldsymbol{Q}[j]}$;
Restore the indices of $\boldsymbol{f}, \boldsymbol{q}$;
return
 $m\boldsymbol{q} + (1 - m)\boldsymbol{f}$.clip($w_A^* \boldsymbol{q}, w_B^* \boldsymbol{q}$);

C EVALUATIONS ON COMPUTATION COST

We quantitatively evaluate the computation cost of Algorithm 1, whose convex optimization problem is solved by Algorithm 3. We take a batch with 512 images and infer 100 times. The time cost is measured by torch.profiler, an official tool provided by PyTorch. We exclude the time for I/O (i.e. from disk to memory, and from CPU to GPU), and only record the time for computation. Our experiment is conducted on a NVIDIA GeForce RTX 3090.

Table 4: Time cost of Algorithm 1 equipped with Algorithm 3.

	IR-152 & CelebA	IR-152 & FaceScrub	VGG-16 & FaceScrub
None	18.63 s	17.70 s	5.65 s
SSD	19.22 s	18.16 s	6.07 s
Increment	3.1%	2.5%	7.4%

Table 4 shows that we increase little time. The higher percentage on VGG-16 is due to the shallower model architecture. In absolute terms, modifying 512 outputs for 100 times only needs 0.5 seconds. If we take the I/O time into account, the percentages will be low enough to be ignored.

D HYPER-PARAMETERS OF DEFENSES

Table 5: Hyper-parameters of the defenses in our experiments.

	IR-152 & CelebA	IR-152 & FaceScrub	VGG-16 & FaceScrub
MID	$\beta = 0.005$	$\beta = 0.02$	$\beta = 0.015$
BiDO	$\lambda_x = 0.0005, \lambda_y = 0.005$	$\lambda_x = 0.005, \lambda_y = 0.05$	$\lambda_x = 0.0005, \lambda_y = 0.005$
LS	$\alpha = -0.1$	$\alpha = -0.05$	$\alpha = -0.05$
TL	freeze 50% layers	freeze 50% layers	freeze 50% layers
SSD	$T = 0.1, \varepsilon = 0.35$	$T = 0.1, \varepsilon = 0.5$	$T = 0.1, \varepsilon = 0.8$

E EXPERIMENT UNDER RLBMI

We conduct the experiment under RLBMI (Han et al., 2023), a soft-label attack algorithm. Aligned with Tables 1-3, the target models are IR-152 trained on CelebA. Due to the computation cost of RLBMI, we only reconstruct 1 image for each label. Table 6 shows that our SSD still outperforms the other defenses.

Table 6: MIA resistance of various defenses under RLBMI attack.

	$\downarrow Acc1$	$\downarrow Acc5$	$\uparrow \delta_{eval}$	$\uparrow \delta_{face}$
None	44%	65%	505	0.89
MID	31%	53%	523	0.90
BiDO	24%	44%	552	1.01
LS	24%	52%	527	0.98
TL	37%	55%	523	0.94
SSD	**21%**	**42%**	**572**	**1.07**

F COMPARISON WITH PURIFIER

Table 7: Comparisons between Purifier and SSD.

		$\downarrow Acc1$	$\downarrow Acc5$	$\uparrow \delta_{eval}$	$\uparrow \delta_{face}$
Mirror	**Purifier**	3.6%	**8.2%**	721	1.43
	SSD	**3.2%**	**8.2%**	**728**	**1.46**
C2FMI	**Purifier**	**0.6%**	2.6%	723	1.56
	SSD	1.2%	**1.8%**	**744**	**1.59**
BREP	**Purifier**	22.8%	41.6%	591	1.14
	SSD	**1.8%**	**4.2%**	**769**	**1.62**
LOKT	**Purifier**	40.2%	57.6%	546	0.99
	SSD	**1.2%**	**9.0%**	**769**	**1.61**
		$\uparrow Acc$	$\downarrow Dist$		
Utility	**Purifier**	84.5%	0.362		
	SSD	**87.1%**	**0.191**		

Purifier (Yang et al., 2023) is a black-box defense against membership inference attacks and may also be effective against model inversion attacks. Despite the lack of details about λ and kNN, we

reproduce their work setting $\lambda = k = 1$. If the L_2 distance between the input and the nearest training sample is less than 0.00005, then we swap the top-1 and top-2 labels. The validation set is used to train the CVAE. Table 7 shows that we outperform Purifier.

G EXPERIMENT ON HIGH RESOLUTION

We use HD CelebA Cropper[3] to generate high resolution CelebA, whose images are cropped and resized to 224×224. The target models IR-152 and evaluation model MaxViT are retrained on new CelebA. The hyper-parameters of defenses are in Table 8 and the test accuracy of MaxViT is 97.2%.

We select Mirror as the attacker, using the 1024×1024 StyleGAN2 trained on FFHQ. The generated images are center-cropped to 800×800 and resized to 224×224. Since high resolution is computationally expensive, we only attack the first 20 labels and reconstruct 5 images for each label. Table 8 shows that our SSD still outperforms the other defenses.

Table 8: Comprehensive results on high resolution.

	Mirror				**Utility**		**Hyper-parameter**
	$\downarrow Acc1$	$\downarrow Acc5$	$\uparrow \delta_{eval}$	$\uparrow \delta_{face}$	$\uparrow Acc$	$\downarrow Dist$	
None	32%	53%	492	1.25	96.4%	0	
MID	37%	68%	477	1.28	93.1%	0.334	$\beta = 0.005$
BiDO	37%	55%	487	1.23	94.7%	0.112	$\lambda_x = 0.05, \lambda_y = 0.5$
LS	28%	43%	508	1.29	94.1%	0.118	$\alpha = -0.005$
TL	64%	96%	472	1.17	94.1%	0.719	freeze 80% layers
SSD	**17%**	**33%**	**522**	**1.30**	**95.0%**	**0.079**	$T = 0.1, \varepsilon = 1.5$

H DISCUSSION ON ADAPTIVE ATTACKS

A potential adaptive attack strategy is:

1. Query the same x repeatedly and count the frequency of different outputs.
2. Estimate our sampling probability by the frequency.
3. Infer the original output $\boldsymbol{f}(x)$ by the sampling probability.

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.

The h can employ a Locality Sensitive Hashing (Gionis et al., 1999) to cope with the minor perturbations in x.

[3] https://github.com/LynnHo/HD-CelebA-Cropper