The Devil's Advocate: Shattering the Illusion of Unexploitable Data using Diffusion Models

Hadi M. Dolatabadi, Sarah Erfani, and Christopher Leckie
School of Computing and Information Systems
The University of Melbourne
Victoria, Australia
{h.dolatabadi, sarah.erfani, caleckie}@unimelb.edu.au

***Abstract*—Protecting personal data against exploitation of machine learning models is crucial. Recently, availability attacks have shown great promise to provide an extra layer of protection against the unauthorized use of data to train neural networks. These methods aim to add imperceptible noise to clean data so that the neural networks cannot extract meaningful patterns from the protected data, claiming that they can make personal data "unexploitable." This paper provides a strong countermeasure against such approaches, showing that unexploitable data might only be an illusion. In particular, we leverage the power of diffusion models and show that a carefully designed denoising process can counteract the effectiveness of the data-protecting perturbations. We rigorously analyze our algorithm, and theoretically prove that the amount of required denoising is directly related to the magnitude of the data-protecting perturbations. Our approach, called AVATAR, delivers state-of-the-art performance against a suite of recent availability attacks in various scenarios, outperforming adversarial training even under distribution mismatch between the diffusion model and the protected data. Our findings call for more research into making personal data unexploitable, showing that this goal is far from over. Our implementation is available at this repository: https://github.com/hmdolatabadi/AVATAR.**

***Index Terms*—neural networks, availability attacks, diffusion models, facial recognition**

I. Introduction

Neural networks have achieved great success in various areas of computer vision including object detection [25, 14], semantic segmentation [78, 36], and photo-realistic image/video generation [31, 11, 54]. While the efforts of the community in the development of such models cannot be undermined, this unparalleled success would have been impossible without the abundance of data resources available today [9, 33, 49, 35]. In this regard, social media, and the internet in general, provides a platform that can be crawled easily to create massive datasets. This capability can act both as a blessing and a curse: while the collected data can facilitate learning larger, more accurate neural networks, the users lose control over protecting their personal data from being exploited. This issue has raised increasing concerns about misuse of personal data [27, 26, 5].

Recently, there has been an increasing number of studies on hindering the unauthorized use of personal data for neural network image classifiers [15, 30, 72, 17, 18, 71, 62, 50]. These methods tend to add an imperceptible amount of noise to the clean images so that while the data has the same appearance as the ground-truth, it cannot provide any meaningful patterns for the neural networks to learn. As a result, such approaches, collectively known as *availability attacks* [4], claim that personal image data can be made *unexploitable* for the neural networks [30, 71]. While there has been an abundance of research on designing better availability attacks, far too little attention has been paid to counter-attacks that might be employed by adversaries to break such precautionary measures.

Unfortunately, the assumptions of existing availability attacks are far too weak to make the data unexploitable. For example, consider a user who shares their protected photos over their social media. We can clearly see that once the photos are shared, they cannot be protected against *all* future countermeasures [47]. For instance, consider a corporate entity that aims to train face recognition models by crawling over social media without the consent of the users. While this unauthorized entity might not have unprotected versions of a particular person's image from his/her social media, they can have a large pre-trained model representing a facial image distribution. Given this threat model, shown in Figure 1, we aim to show that counteracting the protecting perturbations is indeed plausible.

To this end, we show that pre-trained density estimators are powerful tools that can be used to counteract the effects of the data-protecting perturbations, eventually enabling us to exploit protected data. We utilize the power of diffusion models in representing the image data distributions to show that reverse-engineering unexploitable data is easier than what is thought. In particular, given a training dataset, we first diffuse the images by adding a controlled amount of Gaussian noise following the forward process of a pre-trained diffusion model. Then, we denoise the noisy images using the reverse process of the aforementioned model, resulting in a dataset purified from data-protecting perturbations. Theoretically, using contraction properties of stochastic difference equations we prove that the number of diffusion steps required to cancel the data-protecting perturbations is directly influenced by the

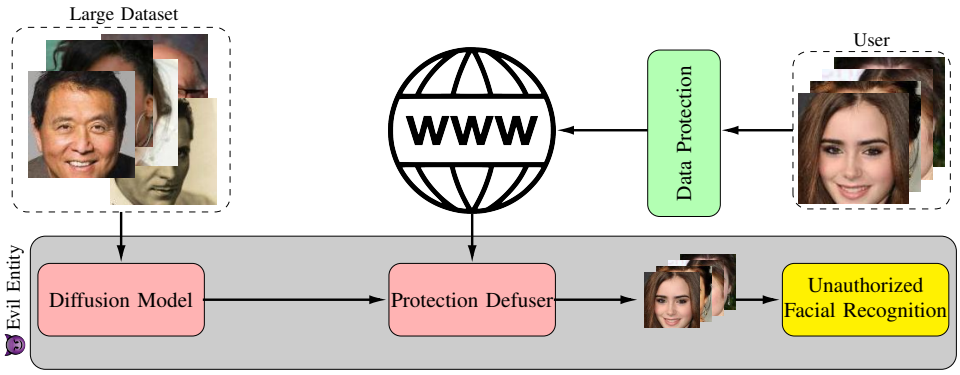


Fig. 1: The threat model considered in this paper. Availability attacks cannot guarantee to protect all the data that exists over the web. A data exploiter might use large density estimators to defuse the data-protecting perturbations and exploit the data.

magnitude of its norm. Thus, protecting personal data using imperceptible perturbations is not possible. We also empirically show that our approach is surprisingly powerful, being able to deliver the state-of-the-art (SOTA) performance against a wide variety of recent availability attacks. Our findings indicate the fragility of *unexploitable data*, calling for more research to protect personal data.

Diffusion models have been extensively used in various areas. Closely related to our work, Yoon *et al.* [70] and Nie *et al.* [42] have employed diffusion models to increase robustness against adversarial attacks. In contrast to these methods, in this paper, we investigate the capabilities of diffusion models as a threat against personal data protected by availability attacks. In particular, we leverage the SOTA diffusion models as a proxy for the true data distribution and argue why unlearnable examples provide a false sense of data privacy.

Our contributions can be summarized as follows:

- We introduce AVATAR as a countermeasure against data availability attacks. To the best of our knowledge, this is the first work that explores the use of diffusion models to circumvent such attacks.
- We show the power of AVATAR in breaking availability attacks over five datasets, four architectures, and seven of the most recent availability attacks. AVATAR achieves the SOTA performance against availability attacks, outperforming adversarial training.
- Our results indicate that even in the absence of the true data distribution, one can use a similar distribution to counteract availability attacks.
- Theoretically, we show that the amount of noise needed to diffuse the data-protecting perturbation is directly related to the magnitude of its norm. This result indicates that achieving both goals of availability attacks (data utility and protection) at the same time is impossible.

II. Related Work

In this section, we review the related work to our approach.

a) Poisoning and Backdoor Attacks: A considerable number of studies have been published on various types of *data poisoning attacks* [4, 51, 20]. These attacks aim to pollute the training data so that they can hinder the performance of the machine learning model at test-time [3, 32, 40]. While these methods are quite successful in achieving this goal, they often tend to perform weakly against neural networks [40] and appear to be distinguishable from the clean samples, damaging the utility of the underlying data [68]. *Backdoor attacks* are a popular family of data poisonings against deep neural networks [23, 2, 63, 12]. Unlike general poisoning attacks, these methods attach triggers to a small fraction of the clean training data so that the model creates an association between the existence of the trigger and a particular class. During inference, the neural network would behave normally on benign samples. However, if the trigger is activated, the model would output the attacker's desired value due to the existence of a backdoor in the model.

b) Availability Attacks: Motivated to address the lack of personal data privacy, an emerging type of poisoning attacks known as *availability attacks* have drawn considerable attention. Unlike previous types of poisoning attacks, availability attacks seek to add imperceptible perturbations to the clean training data with two goals in mind. First, the added perturbation should be able to protect the underlying data from being exploited by a neural network during training. Second, the perturbed data should still preserve its normal utility. To understand these constraints, consider a user sharing their photo over their social media. While the user wants to protect their photo from unauthorized use of web-crawlers to train a face recognition model [27] (first constraint), they still wish their photo to appear normal to their audience (second constraint) [30].

Feng *et al.* [15] propose to produce the poisoning perturbations by training an auto-encoder, whose aim is to get the lowest performance from an auxiliary classifier. In a similar spirit, Tian *et al.* [62] train a conditional generative adversarial network (GAN) [21] to generate the availability attacks' perturbation. The training objective is designed to create a spurious correlation between the noisy image and the ground-truth labels. Concurrently, Yu *et al.* [71] empirically investigate various types of availability attacks and show that

almost all of them leverage these spurious features to create a shortcut within neural networks [19]. Yu *et al.* [71] then propose a fast and scalable approach for perturbation generation by generating randomly-initialized linearly-separable perturbations which can generate availability attacks for an entire dataset in a few seconds. Concurrently, Sandoval-Segura *et al.* [50] proposed another approach that generates the random noise independent from the data. In this approach, first the beginning rows and columns of each channel are populated with Gaussian noise. Then, an autoregressive process is used to find the value of the remaining pixel values.

Another popular approach to generate availability attacks is via direct optimization. Huang *et al.* [30] define a bi-level optimization objective to generate error-minimizing noise for data samples and an auxiliary classifier. It is argued that since the perturbed images minimize the auxiliary classifier's loss, they contain no useful information for any other target classifier to learn, and as such, the model would not exploit them during training. In contrast, Fowl *et al.* [17] show that using adversarial examples [60, 22] as the poisoned data would make it hard for the classifier to learn any meaningful pattern, and thus, they can serve as a powerful family of availability attacks. While optimization-based availability attacks are potent, they are often computationally demanding and several attempts have been made to ease their computational burden [16, 76].

Compared to various types of availability attacks, preventative measures have received little attention. It has been shown that various data augmentation techniques (such as CutOut [10], Mixup [75], CutMix [73], and Fast Auto-augment [34]) are not able to prevent availability attacks [30, 17, 62, 71]. Tao *et al.* [61] show that adversarial training [39, 77, 13], originally proposed to enhance robustness against adversarial attacks [60, 22], can be used to train successful classifiers against availability attacks. Later, Fu *et al.* [18] extended the error-minimizing noise of Huang *et al.* [30] resulting in perturbations that can even prevent adversarial training from learning over the poisoned data. Despite this, adversarial training has remained one of the strongest defense baselines against availability attacks. In this work, we show that one can outperform adversarial training in an attempt to counteract availability attacks.

c) Diffusion Models: Denoising diffusion probabilistic modeling (DDPM) [55, 28] (also known as score-matching networks [56–58]) are a family of deep generative models that have achieved the SOTA performance in image [11, 67], text-to-image [48], video [54], and 3D-object [46] generation. Diffusion models generally comprise of a forward and a backward process [8]. In the forward process, the model gradually adds noise to the data until it is transformed into Gaussian noise. The backward process is the reverse of the forward process, where the model tries to gradually transform/denoise a Gaussian vector into a data point.

III. Proposed Method

This section formally introduces our proposed method, called Avatar (dAta aVailAbiliTy Attacks defuseR). First, we define our notation and problem settings. Next, we introduce our proposed approach that materializes our threat model and provide a theoretical analysis of our framework. Finally, we discuss the potential advantages of Avatar compared to existing methods such as adversarial training.

A. Problem Statement

Let $\mathcal{D} = \{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{n}$ be a labeled dataset consisting of n i.i.d. samples $\boldsymbol{x}^{(i)}$ each with a label $y^{(i)}$. Without loss of generality, in this paper, we consider image data $\boldsymbol{x}^{(i)} \in \mathbb{R}^d$ where d shows the data dimension. Also, we assume that $y^{(i)}$ takes one of the K possible class values $\{1, 2, \ldots, K\}$. Furthermore, let $f_{\boldsymbol{\theta}} : \mathbb{R}^d \to \mathbb{R}^K$ denote a neural network classifier parameterized by $\boldsymbol{\theta}$ that takes an image $\boldsymbol{x}$ and outputs a real-valued vector $\boldsymbol{z} = f_{\boldsymbol{\theta}}(\boldsymbol{x})$ known as the logit. The final decision of the classifier is determined via $\hat{y} = \arg\max_j z_j$. To train the classifier, one usually aims to minimize the empirical error between the ground-truth labels and the classifier predictions:

$$\arg\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x},y)\in\mathcal{D}}[\ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y)], \tag{1}$$

where $\ell(\cdot)$ denotes the cross-entropy loss.

Following the convention in availability attacks, we assume that there exists a data curator that manipulates the dataset $\mathcal{D}$ into $\mathcal{D}_{\text{pr}} = \{(\tilde{\boldsymbol{x}}^{(i)}, y^{(i)})\}_{i=1}^{n}$ such that once a neural network is trained over $\mathcal{D}_{\text{pr}}$, it performs poorly over the clean data $\mathcal{D}$:

$$\begin{aligned} &\arg\max_{\mathcal{D}_{\text{pr}}} \mathbb{E}_{(\boldsymbol{x},y)\in\mathcal{D}}[\ell(f_{\boldsymbol{\theta}^*}(\boldsymbol{x}), y)] \\ &\text{s.t. } \boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x},y)\in\mathcal{D}_{\text{pr}}}[\ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y)]. \end{aligned} \tag{2}$$

Since each image $\tilde{\boldsymbol{x}}^{(i)}$ needs to maintain its normal utility, it is assumed that $\tilde{\boldsymbol{x}}^{(i)} = \boldsymbol{x}^{(i)} + \boldsymbol{\delta}^{(i)}$. Here, $\boldsymbol{\delta}^{(i)}$'s are the data-protecting perturbations such that $\left\|\boldsymbol{\delta}^{(i)}\right\|_p \leq \varepsilon$, where $\|\cdot\|_p$ denotes the L_p norm.

B. dAta aVailAbiliTy Attacks defuseR (Avatar)

As discussed, large pre-trained generative models can pose a threat to availability attacks and personal data protection. In this section, we show how diffusion models, which are the SOTA in image generation, can be leveraged to cancel out the effects of availability attacks.

Recall that availability attacks provide a manipulated version of the original data $\boldsymbol{x}$ that is seemingly unexploitable. At the same time, the protected image $\tilde{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{\delta}$ should have its normal utility as it is going to be used by the users, e.g., to post over their social media. This condition reflects itself through the constraint that $\|\boldsymbol{\delta}\|_p \leq \varepsilon$.

A trivial idea would be to add *random* noise to the protected perturbation that might counteract the perturbation, but this is detrimental/ineffective in removing the unlearnable effect [30]. As such, we propose to use a diffusion model for denoising as outlined next.[1]

[1]Note that while here we use DDPMs [28] to demonstrate our method, it can be easily extended to other types of diffusion models as they are all different ways of representing the same process [58].

Specifically, let us assume that we have a pre-trained DDPM [28] model that represents the data distribution $\boldsymbol{x}_0 \sim p_{\text{data}}(\boldsymbol{x})$. The forward process of this model is represented using a Markov chain of length T, such that:

$$\boldsymbol{x}_t = \sqrt{1-\beta_t}\boldsymbol{x}_{t-1} + \sqrt{\beta_t}\boldsymbol{\epsilon}_t, \tag{3}$$

where $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is the normal distribution, and $t = 1, 2, \ldots, T$. The constants β_t, known as variance schedules, are selected such that $\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. If we set $\alpha_t := \prod_{s=1}^{t}(1-\beta_s)$, then this Markov process can also be performed via a single step [28]:

$$\boldsymbol{x}_t = \sqrt{\alpha_t}\boldsymbol{x}_0 + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}. \tag{4}$$

The reverse of this process is also a variational Markov chain which is represented by:

$$\boldsymbol{x}_{t-1} = \frac{1}{\sqrt{1-\beta_t}}\left(\boldsymbol{x}_t + \beta_t \mathbf{s}_{\boldsymbol{\phi}}(\boldsymbol{x}_t, t)\right) + \sqrt{\beta_t}\boldsymbol{\epsilon}_t. \tag{5}$$

Here, $\mathbf{s}_{\boldsymbol{\phi}}(\cdot, t)$ is a network parameterized by $\boldsymbol{\phi}$ representing the score of the noisy data distribution at scale t.

To cancel the effects of the data-protecting perturbations, we propose to first add Gaussian noise to the data. The amount of noise should be adjusted in a way that each image maintains its visual appearance. Otherwise, the semantic information of each image would be lost, and since the reverse process is probabilistic, the original image might not be recovered. In particular, let $\tilde{\boldsymbol{x}}$ be a protected image. We perform the forward process up to a step $t^* < T$ such that the semantic information of the image is preserved:

$$\bar{\boldsymbol{x}}_{t^*} = \sqrt{\alpha_{t^*}}\tilde{\boldsymbol{x}} + \sqrt{1-\alpha_{t^*}}\boldsymbol{\epsilon}. \tag{6}$$

Now, we have managed to diminish the effects of the data-protecting perturbation in $\boldsymbol{x}_{t^*}$. However, this way we would also damage the semantic features of the data which makes it hard to train a neural network model (see the ablation study in Figure 4). To revert to the normal image space, we use the reverse process of our diffusion model to denoise the data:

$$\bar{\boldsymbol{x}}_{t-1} = \frac{1}{\sqrt{1-\beta_t}}\left(\bar{\boldsymbol{x}}_t + \beta_t \mathbf{s}_{\boldsymbol{\phi}}(\bar{\boldsymbol{x}}_t, t)\right) + \sqrt{\beta_t}\boldsymbol{\epsilon}_t. \tag{7}$$

Recursively solving Equation (7) from t^* to 1, we get a denoised version of the data which we denote by $\bar{\boldsymbol{x}} = \bar{\boldsymbol{x}}_0$. Using this process, , shown in Figure 2, we unlock the entire dataset $\mathcal{D}_{\text{pr}}$, and construct a new one $\mathcal{D}_{\text{de}} = \{(\bar{\boldsymbol{x}}^{(i)}, y^{(i)})\}$ for neural network training. Algorithm 1 shows our final algorithm for training a neural network using AVATAR.

Algorithm 1 dAta aVailAbiliTy Attacks defuseR

Input: protected dataset $\mathcal{D}_{\text{pr}} = \{(\tilde{\boldsymbol{x}}^{(i)}, y^{(i)})\}_{i=1}^{n}$, pre-trained diffusion model $\mathbf{s}_{\boldsymbol{\phi}}(\cdot, t)$.

Output: trained neural network classifier $f_{\boldsymbol{\theta}}(\cdot)$.

Parameters: noise time-step t^*, learning rate α, total epochs E, and batch-size b.

```
Initialize θ randomly.
Set D_de = {}.
for (x̃, y) in D_pr do
    x̄_{t*} = √α_{t*} x̃ + √(1 − α_{t*}) ε.
    for t in t*, t* − 1, ··· , 0 do
        x̄_{t−1} = (1/√(1−β_t)) (x̄_t + β_t s_φ(x̄_t, t)) + √β_t ε_t.
    end for
    Add (x̄_0, y) to the dataset D_de.
end for
for i = 1, 2, ..., E do
    Assign D_de to batches of size b randomly.
    for batch in batches do
        θ ← SGD (batch, f_θ, α).
    end for
end for
```

C. Conflicting Assumptions in Availability Attacks

So far, we discussed how by using diffusion models we can nullify the effects of the data-protecting perturbations. Here, we take a theoretical perspective on our proposed solution and show that in this setting, the two constraints of availability attacks conflict with each other. Specifically, from the perspective of availability attacks our result indicates that for a better data protection against AVATAR, we need larger perturbation norms. However, enlarging the perturbation is in conflict with retaining data utility which is the ultimate aim of availability attacks as discussed in Section III-B.

Theorem 1. *Let $\boldsymbol{x} \in \mathbb{R}^d$ denote a clean image and $\tilde{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{\delta}$ its protected version, where $\boldsymbol{\delta}$ denotes any arbitrary data protection perturbation. Also, let $\bar{\boldsymbol{x}}_0$ be the sanitized image using the* AVATAR *denoising process given in Equations (6) and (7). If we set t^* such that*

$$2\log\left(\frac{2\left\|\boldsymbol{\delta}\right\|^2 + 4d}{\mu\Delta}\right) \leq t^*\beta_{t^*} \leq \frac{\mu\Delta}{4d},$$

then the estimation error between the sanitized $\bar{\boldsymbol{x}}_0$ and clean image $\boldsymbol{x}$ can be bounded as:

$$\mathbb{E}\left[\left\|\bar{\boldsymbol{x}}_0 - \boldsymbol{x}\right\|^2\right] \leq 2(\mu+1)\Delta,$$

where $\Delta = \mathbb{E}[\|\boldsymbol{x}_0 - \boldsymbol{x}\|^2]$ and $\mu > 0$ is a constant.

Proof. See Appendix A for our proof using the contraction property of stochastic difference equations. □

Theorem 1 states that for a protected image with a larger perturbation norm $\|\boldsymbol{\delta}\|$, a larger amount of noise (determined by $t^*\beta_{t^*}$) is required. However, the amount of noise cannot be arbitrarily large as the semantic information of the image might be lost in the process (as indicated by the presence of Δ in the upper-bound).

D. AVATAR vs. Adversarial Training

As Tao *et al.* [61] have demonstrated, adversarial training (AT) [39] could also be used to train successful models over unexploitable data. However, our approach has several key advantages compared to AT:

1) First, AT modifies the learning algorithm, and as such, it needs to be applied separately for training each neural network. In contrast, AVATAR sanitizes the data only *once*. As a result, AVATAR is more efficient.

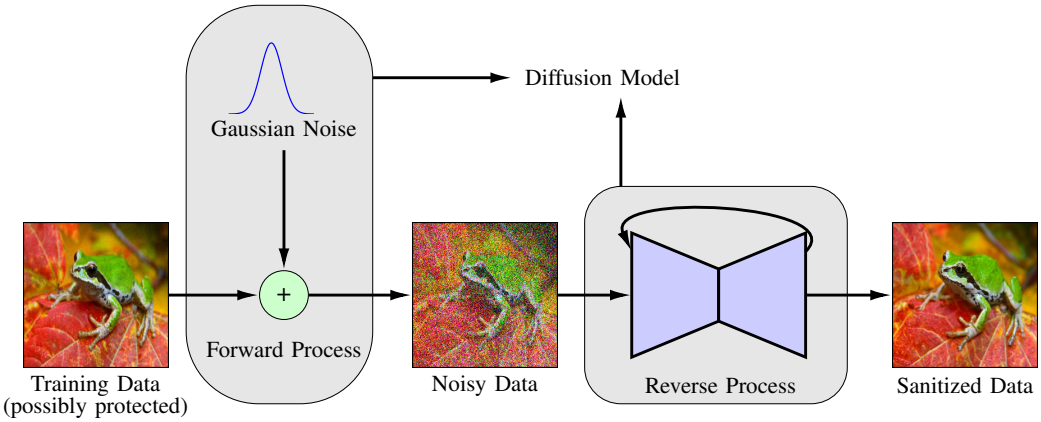


Fig. 2: Overview of AVATAR. According to a pre-trained diffusion model, we first add a controlled amount of Gaussian noise to the training data. Then, we use the reverse diffusion process to denoise the data which is later going to be used for neural network training.

2) Second, as shown by Tsipras *et al.* [64], AT greatly affects the clean accuracy in its learning process, and as such, might not be the ultimate method for defending against availability attacks.
3) Lastly, as Fu *et al.* [18] show, one can build unexploitable data against AT that would essentially render AT vulnerable to availability attacks. However, to the best of our knowledge, no adaptive availability attacks have been proposed against diffusion models so far.

IV. Experimental Results

In this section, we run various experiments to analyze the performance of AVATAR against availability attacks:

1) We conduct extensive experiments on seven SOTA availability attacks and show that given the data distribution, AVATAR can counteract them (Section IV-B).
2) We provide detailed comparisons against various preprocessing techniques (Section IV-C), early stopping (Section IV-D), and adversarial training (Section IV-E) to show that AVATAR delivers the best performance.
3) We provide extensive ablation studies into different assumptions made by AVATAR. First, we show that the training data overlap between the diffusion model and the unlearnable example generation has **no effect** on the performance of AVATAR (Section IV-G). Interestingly, we show that even a similar, different, or even poisoned distribution compared to the true data distribution can counteract availability attacks (Section IV-H).
4) We simulate our scenario given in Figure 1 for the real-world application of facial recognition to show the plausibility of our approach. Again, here we use a diffusion model trained on a different dataset, but we manage to counteract the unlearnable examples for another dataset (Section IV-I).

We also include an extended version of our experimental results in Appendix B.

TABLE I: The classes of CIFAR-10 and their matching ones in the ImageNet-10 dataset.

CIFAR-10	**IN-10**
Airplane	Airliner
Automobile	Wagon
Bird	Humming Bird
Cat	Siamese Cat
Deer	Ox
Dog	Golden Retriever
Frog	Tailed Frog
Horse	Zebra
Ship	Container Ship
Truck	Trailer Truck

A. Details of Experimental Settings

In this section, we provide the details of our experimental settings.

a) Datasets: In our experiments, we use four different datasets. CIFAR-10 & 100 [33] are 32×32 datasets of colored images, where the classes contain different objects, animals, plants, etc. SVHN [41] is a dataset of house numbers from 0 to 9 in a natural, street view setting. Finally, ImageNet [49] is a dataset of natural images of size 224×224 with 1000 classes. In our experiments, we use two simplified versions of this dataset. First, following the convention of prior research, we select the first 100 classes of this dataset, which we refer to as ImageNet (IN)-100. Second, for our distribution mismatch experiments, we follow Huang *et al.* [30] and select 10 classes of ImageNet that are closely aligned with CIFAR-10 and downscale them to 32×32 size. We call this dataset IN-10. The information on the selected classes can be found in Table I. Finally, we also use the 32×32 version of the ImageNet dataset for some of our experiments, which we denote by IN-1k-32×32.

b) Classifiers: In our experiments, we use four types of neural network image classifiers, namely: ResNet-18 (RN-

18) [25], VGG-16 [53], DenseNet-121 (DN-121) [29], and WideResNet-34 (WRN-34) [74]. For training these classifiers over different datasets and also training objectives (vanilla vs. adversarial training (AT)), we follow two different training conventions. The hyper-parameters of each setting are given in Table II. Furthermore, Table III indicates the setting used for each experiment in the paper.

c) Diffusion Models: For the diffusion models used during the denoising process of AVATAR (shown in Figure 2), we follow the implementation of DiffPure[2] and use score SDE [58] (for CIFAR-10, CIFAR-100, SVHN, IN-10) and the guided DDPM (for IN-100 and IN-1k-32×32.) [11]. For CIFAR-10 and IN-100, we download the pre-trained versions available online.[3] for IN-1k-32×32 dataset. Additionally, for CIFAR-100, IN-10, and SVHN we use the PyTorch repository of score SDE [58], and train variance-preserving diffusion models with continuous DDPM++ architecture, similar to the one used for CIFAR-10. The FID score of the trained diffusion models is given in Table IV.

d) Availability Attacks: We use seven SOTA availability attacks in our experiments: DeepConfuse (CON) [15], Neural Tangent Generalization Attacks (NTGA) [72], Error-minimizing Noise (EMN) [30], Targeted Adversarial Poisoning (TAP) [17], Robust EMN (REMN) [18], Shortcut (SHR) [19], and Autoregressive attacks (AR) [50]. The details of each availability attack are given below:

- For CON [15], we use the released protected CIFAR-10 dataset, available online at SHR [71] repository.[4] Note that since generating this attack for the CIFAR-10 dataset would take 5-7 days, we just used the available data for CIFAR-10 and skipped generating the attack for the other datasets.
- For NTGA [72], we use their code[5] to generate availability attacks for our datasets. For CIFAR-10, we used the data published online. For CIFAR-100 and SVHN, we used the online repository, and generate NTGA protected data using the `CNN` surrogate model, `time-step` of 64, and `block-size` of 100 to generate perturbations of magnitude $\|\boldsymbol{\delta}\|_\infty \leq 8/255$. Due to limited GPU memory, we used the `FNN` surrogate model to generate perturbations of magnitude $\|\boldsymbol{\delta}\|_\infty \leq 0.1$ for IN-100. The rest of the hyper-parameters were set similarly to CIFAR-100 and SVHN.
- For EMN [30], TAP [17], and REMN [18], we use the online repository of REMN[6] which contains an implementation of EMN and TAP as well. We use the default CIFAR-10 configurations of this repository for CIFAR-10, CIFAR-100, and SVHN. For IN-10, we used the default MiniIN configurations of the REMN code-base.
- Moreover, we use the SHR GitHub repository[4] to generate shortcut attacks. For CIFAR-10, CIFAR-100, and SVHN, we use the default settings. For IN-100, we use `patchsize` of 32 as advised by the authors.
- Finally, we use the official data released on the AR GitHub repository for this attack.[7]

A few samples for each availability attack are shown in Figure 7.

B. Exploiting Protected Data

Table V shows our results for breaking availability attacks against for four different datasets. As can be seen, AVATAR can significantly improve the performance of neural network training in almost all cases. Moreover, although the training data was produced using diffusion models, the trained neural networks can generalize to unseen test data easily. This trend is more evident in the CIFAR-10 and SVHN datasets where the pre-trained diffusion model can better represent the image data density, as indicated by their low FID scores.

C. Comparison with Data Augmentation Techniques

AVATAR can be regarded as a type of data pre-processing where the inner mechanics of the learning algorithms are not modified. As such, here we compare our approach with various SOTA data augmentation techniques that can be utilized during model training. To this end, we follow the settings of [30], and adopt four widely used data augmentation techniques. In addition, we employ the JPEG and grayscale pre-processing [38] as well as two blurring techniques in Table VI. Finally, we also test the quantization and total variation minimization (TVM) approaches that have shown to be effective against adversarial attacks [24]. Table VI shows the performance of these methods compared to AVATAR. As shown, our approach outperforms various types of pre-processing/data augmentation methods.

D. The Effect of Early Stopping

It has been previously shown that early stopping can also be beneficial against availability attacks [30]. As such, here we run the same set of experiments over availability attacks for the CIFAR-10 dataset, but this time we record the highest accuracy attainable during training. Table VII shows our results. As seen, using our approach one achieves stable training, where the variance between the final model accuracy and the highest attainable accuracy is very low. Notably, while these results indicate that existing availability attacks are less powerful than what is thought, early stopping is not sufficient to recover the best model performance. In contrast, AVATAR can significantly cancel the effects of availability attacks.

[2] https://github.com/NVlabs/DiffPure

[3] For CIFAR-10, we used the checkpoint for the `vp/cifar10_ddpmpp_deep_continuous` setting on score SDE repository: https://github.com/yang-song/score_sde_pytorch. Moreover, we used the unconditional 256 × 256 model available on the guided DDPM code-base for IN-100 expriments: https://github.com/openai/guided-diffusion. Finally, we use the pre-trained DDPM-IP [43] models available on https://github.com/forever208/DDPM-IP

[4] https://github.com/dayu11/Availability-Attacks-Create-Shortcuts

[5] https://github.com/lionelmessi6410/ntga

[6] https://github.com/fshp971/robust-unlearnable-examples

[7] https://github.com/psandovalsegura/autoregressive-poisoning

TABLE II: Training hyper-parameters used in our experiments.

Hyper-parameter	Setting #1	Setting #2
Optimizer	SGD	SGD
Scheduler	Multi-step	Multi-step
Initial lr.	0.1	0.1
lr. decay	0.1 (@epoch: 80 & 100)	0.1 (@iter: 16k & 32k)
Batch Size	128	128
Training Steps	120 (epochs)	40k (iters)
Weight Decay	0.0005	0.0005
PGD Steps (for AT only)	-	10
PGD Step Size (for AT only)	-	0.8

TABLE III: Setting number used for each experiment.

Experiment	Setting #1	Setting #2
Table V (CIFAR-10)	✓	-
Table V (CIFAR-100)	✓	-
Table V (SVHN)	✓	-
Table V (IN-100)	-	✓
Table VI	✓	-
Table VII	✓	-
Table VIII	✓	-
Figure 4	✓	-
Figure 3	-	✓

TABLE IV: The FID of the diffusion models used for denoising. * denotes that the FID has been computed using 10k generated samples only. † indicates that the scores have been adapted from relative literature.

Dataset	FID	Dataset	FID
CIFAR-10†	2.41	SVHN*	2.59
CIFAR-10 (TAP)*	4.11	CIFAR-100*	4.85
IN-10*	17.32	IN†	4.59

E. Comparison with Adversarial Training

As mentioned in Section II, adversarial training (AT) [39] is the most successful defense technique against availability attacks [61]. For the next set of experiments, we follow the settings of Fu *et al.* [18] and compare our approach with AT. To this end, we run two different scenarios. First, we perform AT over the protected data. Then, we run AT over the data that is defused (i.e., counteracted) by AVATAR. In both cases, we vary the perturbation bound ε from 0 to 4, where 0 is the vanilla training. Figure 3 shows our results. Apart from what we discussed in Section III-D, two additional insights are worth mentioning here:

(1) In most cases, AVATAR without AT (i.e., $\varepsilon = 0$) performs on-par or better than AT with $\varepsilon > 0$. Thus, AVATAR delivers the SOTA against availability attacks.

(2) As seen in Figure 3, AT yields the worst performance against REMN [18]. However, our approach can combat REMN [18] successfully, and it is the first approach that does so.

F. Setting Diffusion Step t^*

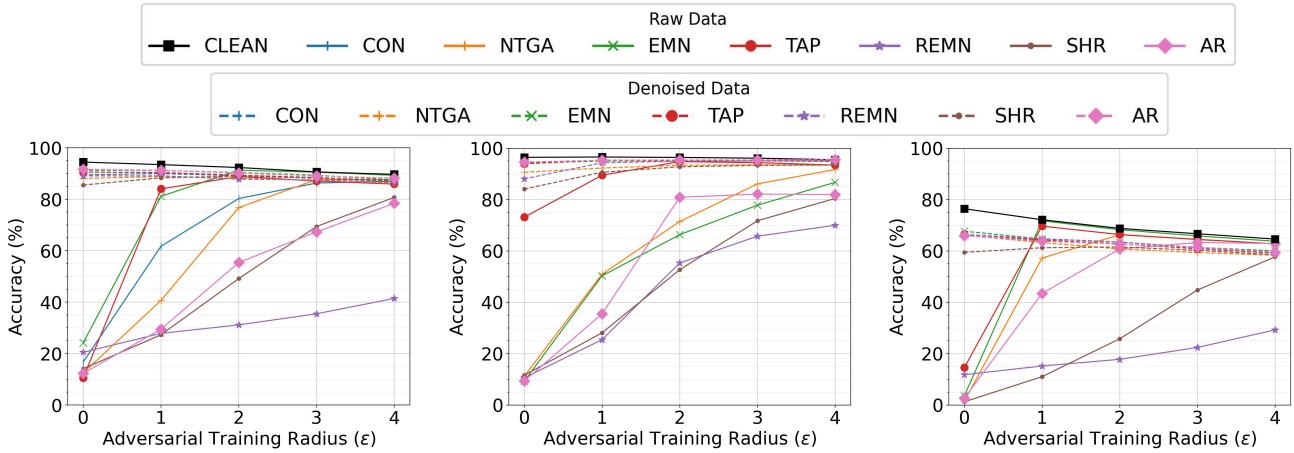
As discussed in Section III-C, setting the diffusion timestep should be performed carefully. Otherwise, either the data-protecting noise is not eliminated, or the semantic information of the image is lost. Here, we run an ablation study over the diffusion timestep. In particular, for our CIFAR-10 experiments, we run AVATAR with five different timesteps from $\{0, 100, 200, 300, 400\}$. Then, we measure the test accuracy of the trained neural networks over the clean test set. As shown in Figure 4, setting t^* too small means that the data-protecting perturbations are not removed. In contrast, setting t^* to a large value might remove the semantic information which in turn damages the generalizability of the trained model. For a more thorough discussion on selecting t^*, please see Appendix B-C.

G. The Effect of Diffusion Models' Training Data

It is well-known from the literature that diffusion models are not a mere memorization of their training data [58] and can further enhance the accuracy of down-stream tasks [1, 66]. To empirically eradicate the influence of training data overlap on our results, we perform the following experiment. Apart from our results in Table V, we run a second set of experiments where we create disjoint subsets of training data for training diffusion models and those used as unlearnable examples. Then, we train our in-house diffusion model and perform a similar experiment to that of Table V, but this time with this new, non-overlapping set of data. Finally, we measure the performance over the unseen test data. We report the relative error rate with respect to the clean data performance in Figure 5. As seen, the overlap in diffusion models' training data has no impact on AVATAR's final performance. We further validate this through our real-world experiments in Section IV-I.

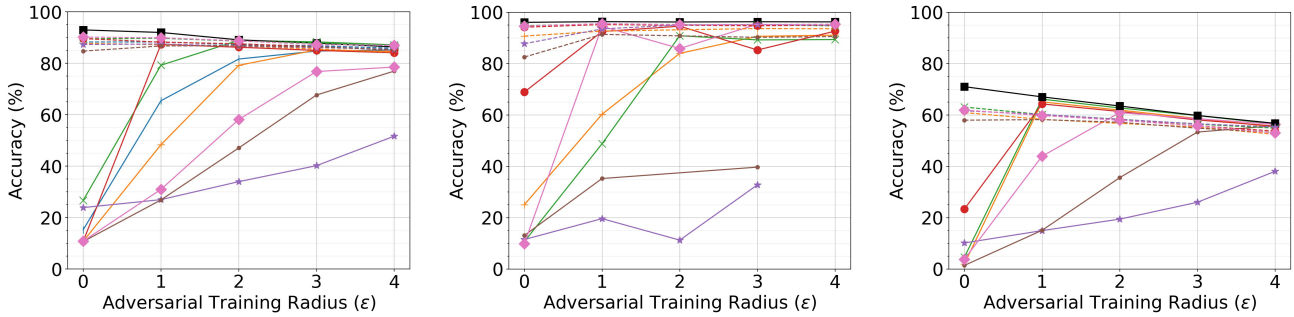
H. Distribution Mismatch

To go even further, we show that AVATAR is even resilient to a distribution mismatch between the diffusion model and the training data. In particular, we train three diffusion models over the protected CIFAR-10 dataset with TAP [17], IN-10 which contains 10 classes of ImageNet that are most similar to CIFAR-10 dataset [30] (see Table I for more details), and



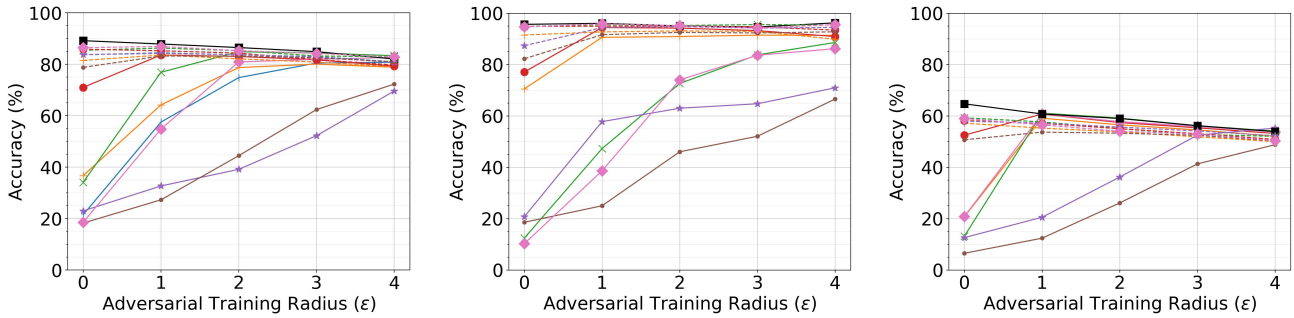
CIFAR-10 SVHN CIFAR-100

(a) RN-18



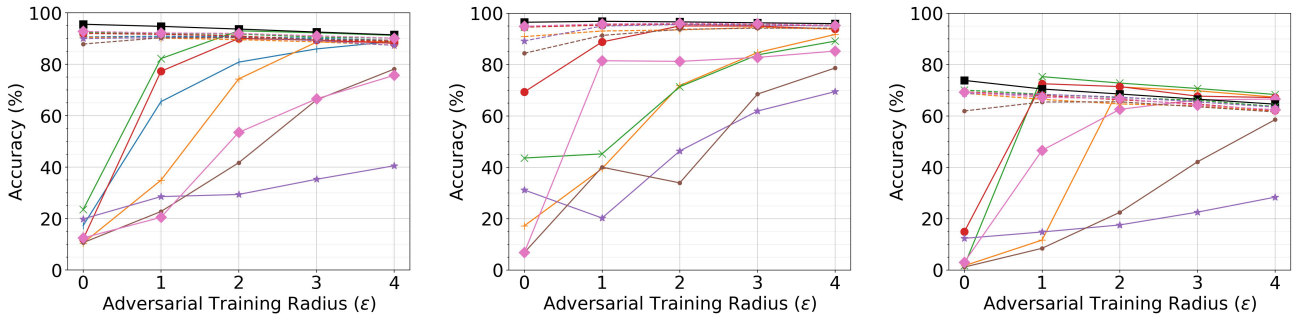
CIFAR-10 SVHN CIFAR-100

(b) VGG-16



CIFAR-10 SVHN CIFAR-100

(c) DN-121



CIFAR-10 SVHN CIFAR-100

(d) WRN-34

Fig. 3: Test accuracy of CIFAR-10, SVHN, and CIFAR-100 classifiers against availability attacks using adversarial training with different perturbation radii.

TABLE V: Test accuracy (%) of RN-18 architectures trained over data availability attacks on CIFAR-10, CIFAR-100, and SVHN, and ImageNet-100 datasets without and with our denoising approach. The mean and standard deviation are computed over 5 seeds. For our results over other architectures, please see Table X.

Data	Method	Clean	Data Availability Attacks					
			NTGA	EMN	TAP	REMN	SHR	AR
CIFAR-10	Vanilla	94.50 ± 0.09	11.49 ± 0.69	24.85 ± 0.71	7.86 ± 0.90	20.50 ± 1.16	10.82 ± 0.22	12.09 ± 1.12
	AVATAR		87.95 ± 0.28	90.95 ± 0.10	90.71 ± 0.19	88.49 ± 0.24	85.69 ± 0.27	91.57 ± 0.18
SVHN	Vanilla	96.29 ± 0.12	9.65 ± 0.70	9.13 ± 2.00	65.97 ± 1.99	11.55 ± 0.19	10.59 ± 3.98	6.76 ± 0.07
	AVATAR		89.84 ± 0.32	93.84 ± 0.12	93.35 ± 0.10	88.51 ± 0.23	83.82 ± 0.39	94.13 ± 0.17
CIFAR-100	Vanilla	75.01 ± 0.41	1.32 ± 0.31	2.05 ± 0.18	14.10 ± 0.19	10.88 ± 0.33	1.39 ± 0.10	2.15 ± 0.46
	AVATAR		63.98 ± 0.55	65.73 ± 0.36	64.99 ± 0.10	64.88 ± 0.08	58.52 ± 0.46	64.54 ± 0.23
IN-100	Vanilla	80.05 ± 0.13	74.74 ± 0.52	1.78 ± 0.17	9.14 ± 0.40	13.28 ± 0.51	43.48 ± 1.56	
	AVATAR		71.08 ± 0.48	72.84 ± 0.90	76.52 ± 0.46	39.79 ± 0.98	59.85 ± 1.01	

TABLE VI: Test accuracy (%) of RN-18 models trained over data availability attacks on CIFAR-10 dataset using different data augmentation/pre-processing techniques. The results are averaged over 5 runs. The best results are highlighted in bold.

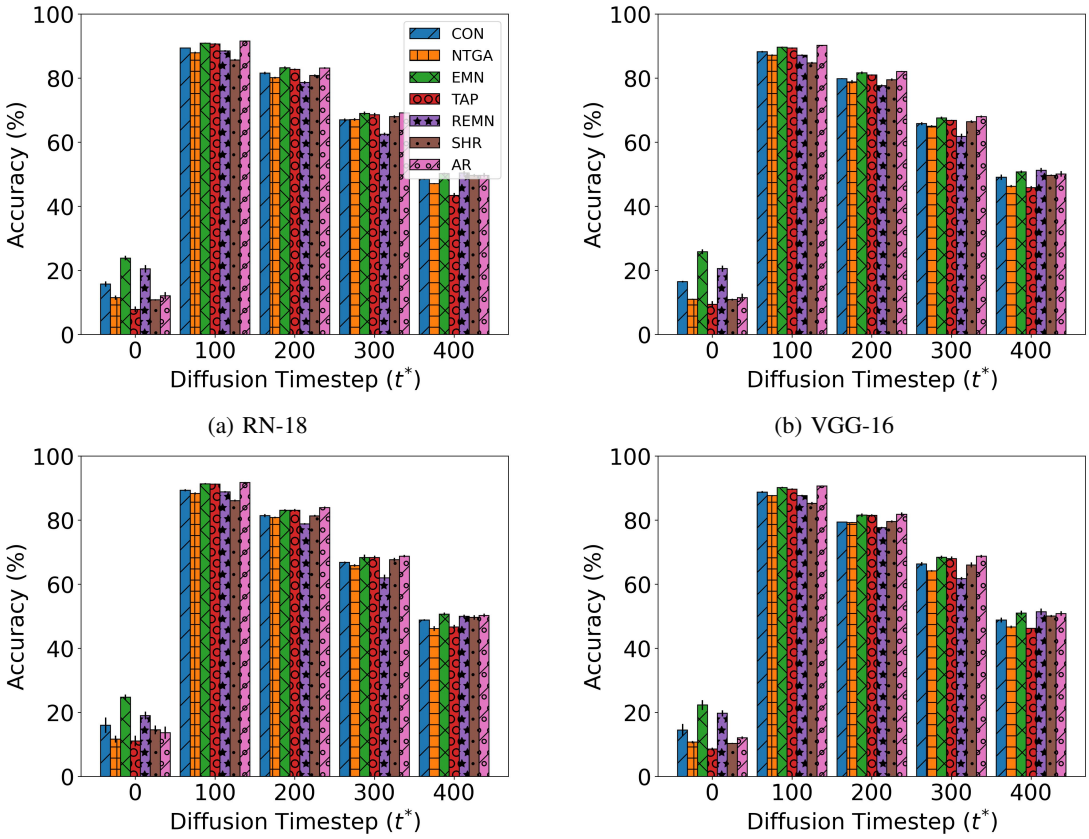
Method	Clean	Data Availability Attacks						
		CON	NTGA	EMN	TAP	REMN	SHR	AR
Vanilla	94.50 ± 0.09	15.75 ± 0.82	11.49 ± 0.69	24.85 ± 0.71	7.86 ± 0.90	20.50 ± 1.16	10.82 ± 0.22	12.09 ± 1.12
Cutout	94.39 ± 0.12	13.53 ± 0.34	13.43 ± 1.15	23.79 ± 1.28	9.73 ± 1.06	20.48 ± 1.09	11.78 ± 0.81	11.21 ± 1.01
MixUp	94.87 ± 0.05	28.58 ± 2.88	13.54 ± 0.36	51.48 ± 0.97	30.09 ± 1.93	26.61 ± 1.65	19.69 ± 0.71	12.67 ± 1.02
CutMix	$\mathbf{95.16 \pm 0.03}$	19.04 ± 2.74	14.16 ± 1.64	25.30 ± 1.18	7.45 ± 1.21	26.83 ± 1.99	10.89 ± 0.34	11.36 ± 0.50
FAutoAug.	95.11 ± 0.14	51.62 ± 1.28	27.56 ± 2.45	56.31 ± 1.13	20.39 ± 0.81	26.65 ± 0.89	25.88 ± 0.62	13.53 ± 0.79
Median Blur	85.83 ± 0.71	15.14 ± 0.38	28.43 ± 1.41	26.97 ± 0.39	57.16 ± 0.75	23.32 ± 0.69	17.50 ± 0.38	14.97 ± 0.40
Gaus. Blur	94.33 ± 0.08	15.36 ± 0.69	11.86 ± 0.81	24.08 ± 0.40	8.87 ± 0.65	20.89 ± 1.56	11.39 ± 1.91	13.39 ± 1.08
Quantization	94.57 ± 0.14	15.17 ± 0.79	16.29 ± 1.03	25.38 ± 0.68	7.38 ± 1.59	22.33 ± 1.21	11.12 ± 0.24	12.87 ± 0.69
TVM	73.20 ± 1.32	42.82 ± 2.00	47.41 ± 1.37	54.86 ± 2.17	70.66 ± 0.58	19.28 ± 1.12	25.35 ± 2.54	34.09 ± 2.07
Grayscale	92.89 ± 0.20	83.30 ± 0.40	63.21 ± 0.85	$\mathbf{92.09 \pm 0.22}$	9.57 ± 0.59	75.84 ± 1.36	57.13 ± 0.87	35.88 ± 0.99
JPEG	84.99 ± 0.28	82.87 ± 0.23	79.26 ± 0.12	84.65 ± 0.16	83.44 ± 0.36	83.66 ± 0.30	69.03 ± 0.62	85.03 ± 0.23
AVATAR (Ours)	90.16 ± 0.21	$\mathbf{89.43 \pm 0.09}$	$\mathbf{87.95 \pm 0.28}$	90.95 ± 0.10	$\mathbf{90.71 \pm 0.19}$	$\mathbf{88.49 \pm 0.24}$	$\mathbf{85.69 \pm 0.27}$	$\mathbf{91.57 \pm 0.18}$

CIFAR-100. Then, we use these surrogate distributions to sanitize protected CIFAR-10 data and train a neural network over the denoised data. We report our results in Table VIII. Surprisingly, our approach can tolerate the distribution mismatch to some extent. As the diffusion model density gets closer to the true training data, the performance gap is gradually closed. Interestingly, even using a diffusion model that is trained over protected data can be beneficial in removing the effects of availability attacks. Note that according to our threat model discussed in Figure 1, this case is too extreme, meaning that the data protector needs to add a perturbation to all the data on the web which is almost impossible. Interestingly, our method using the sub-optimal CIFAR-100 distribution is still performing better than grayscale and JPEG compression techniques of Liu *et al.* [38].

These results motivates us to run AVATAR in a real-world case. In particular, we employ the off-the-shelf diffusion model, DDPM-IP [43], that is trained over the 32×32 version of the ImageNet dataset in AVATAR. Then, we re-run our experiments of Table V on CIFAR-10, CIFAR-100, and SVHN

TABLE VII: Test accuracy (%) of RN-18 models trained over data availability attacks on CIFAR-10 dataset. The early stopping rows contain the highest achievable accuracy over the course of training. The results are averaged over 5 runs.

Method	**Data Availability Attacks**						
	CON	NTGA	EMN	TAP	REMN	SHR	AR
Vanilla	15.75 ± 0.82	11.49 ± 0.69	24.85 ± 0.71	7.86 ± 0.90	20.50 ± 1.16	10.82 ± 0.22	12.09 ± 1.12
+ Early Stopping	23.99 ± 6.22	31.71 ± 3.97	27.23 ± 1.83	67.13 ± 2.03	21.90 ± 0.57	22.72 ± 0.83	38.78 ± 8.65
AVATAR (Ours)	89.43 ± 0.09	87.95 ± 0.28	90.95 ± 0.10	90.71 ± 0.19	88.49 ± 0.24	85.69 ± 0.27	91.57 ± 0.18
+ Early Stopping	89.55 ± 0.15	88.07 ± 0.22	91.07 ± 0.11	91.00 ± 0.11	88.59 ± 0.26	85.76 ± 0.25	91.63 ± 0.17



(a) RN-18 (b) VGG-16 (c) DN-121 (d) WRN-34

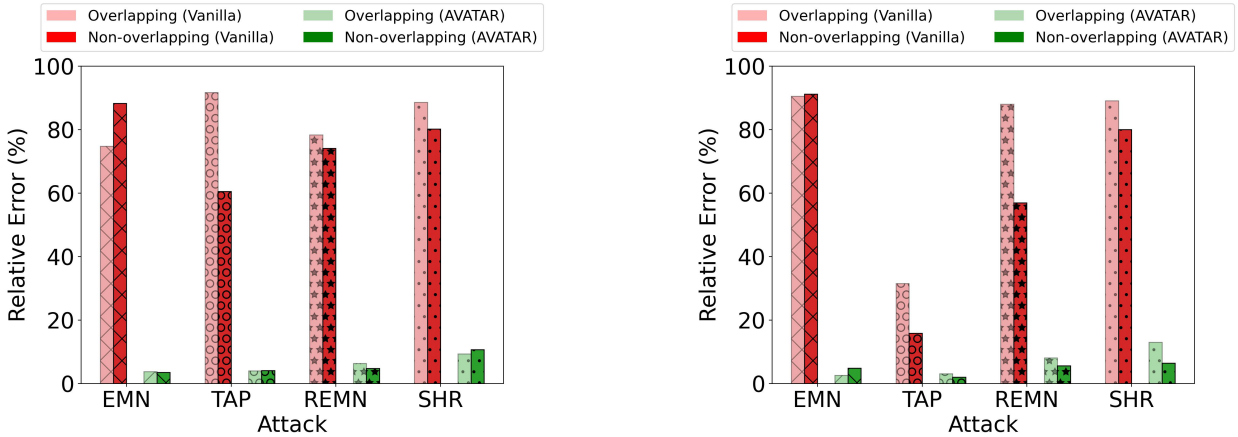
Fig. 4: Effect of changing the forward process diffusion timestep in AVATAR on the final test accuracy in CIFAR-10 classifiers.

using this diffusion model. As this DDPM-IP [43] uses a cosine schedule [11], we need to adjust the value of t^* to reflect this change. As we discuss in Appendix B-C, we set $t^* = 200$ to have an equivalent performance to the linear schedule that was used in our earlier experiments.

Our results are shown in Table IX. As seen, AVATAR is resilient to the choice of the diffusion model. Even though there is a distribution mismatch between our test datasets and ImageNet-32×32, our results are on par with the use of the matching data distribution. These results indicate the real-world value of AVATAR which can serve as a strong baseline against availability attacks.

I. Real-world Example I: Face Recognition

In Section I, we discussed in detail that the threat model of existing availability attacks is fragile and a malicious adversary might still exploit the personal data. This means that possibly no *imperceptible* adversary can protect the image data from being maliciously used. To show this, we discussed a real-world example in Section IV following a similar experiment from Huang *et al.* [30]. In particular, we create a set of clean and protected identities in the WebFace [69] dataset by randomly selecting 50 identities from this dataset. As a result, the remaining 10522 identities constitute our clean data. For all of the identities, we randomly split the data so that 80% of that



(a) CIFAR-10

(b) SVHN

Fig. 5: Relative error rate of RN-18 models trained against availability attacks on CIFAR-10 and SVHN averaged over 5 runs. **Overlapping** indicates that the diffusion model and availability attacks use the same subset as training data. **Non-overlapping** means that the diffusion model and availability attacks are trained on **disjoint** subsets of data.

TABLE VIII: Test accuracy (%) of RN-18 models trained over data availability attacks on the CIFAR-10 dataset. For AVATAR, we use different pre-trained distributions over CIFAR-10, poisoned CIFAR-10 (TAP), ImageNet-10 (IN-10) [30], and CIFAR-100. The results are averaged over 5 runs.

Distribution	**Data Availability Attacks**						
	CON	NTGA	EMN	TAP	REMN	SHR	AR
Vanilla	15.75 ± 0.82	11.49 ± 0.69	24.85 ± 0.71	7.86 ± 0.90	20.50 ± 1.16	10.82 ± 0.22	12.09 ± 1.12
CIFAR-10 (TAP)	61.70 ± 2.02	75.62 ± 3.75	64.03 ± 0.98	35.09 ± 2.28	60.16 ± 1.44	74.96 ± 2.82	60.36 ± 2.29
IN-10	80.98 ± 0.06	79.42 ± 0.25	83.78 ± 0.39	82.71 ± 0.24	82.83 ± 0.28	75.91 ± 0.06	84.88 ± 0.19
CIFAR-100	84.85 ± 0.49	83.07 ± 0.33	87.81 ± 0.14	86.55 ± 0.26	85.84 ± 0.19	79.52 ± 0.22	88.59 ± 0.15
CIFAR-10	89.43 ± 0.09	87.95 ± 0.28	90.95 ± 0.10	90.71 ± 0.19	88.49 ± 0.24	85.69 ± 0.27	91.57 ± 0.18

TABLE IX: Test accuracy (%) of RN-18 models trained over data availability attacks on CIFAR-10, CIFAR-100, SVHN with our denoising approach using the matching distribution and ImageNet-32×32. The mean and standard deviation are computed over 5 seeds.

Data	**Distribution**	**Clean**	**Data Availability Attacks**					
			NTGA	EMN	TAP	REMN	SHR	AR
CIFAR-10	CIFAR-10	94.50 ± 0.09	87.95 ± 0.28	90.95 ± 0.10	90.71 ± 0.19	88.49 ± 0.24	85.69 ± 0.27	91.57 ± 0.18
	ImageNet-32×32		86.41 ± 0.21	90.17 ± 0.15	89.02 ± 0.15	88.26 ± 0.24	82.97 ± 0.24	90.61 ± 0.18
SVHN	SVHN	96.29 ± 0.12	89.84 ± 0.32	93.84 ± 0.12	93.35 ± 0.10	88.51 ± 0.23	83.82 ± 0.39	94.13 ± 0.17
	ImageNet-32×32		91.32 ± 0.17	94.82 ± 0.10	95.01 ± 0.21	91.00 ± 0.27	83.12 ± 0.30	94.29 ± 0.22
CIFAR-100	CIFAR-100	75.01 ± 0.41	63.98 ± 0.55	65.73 ± 0.36	64.99 ± 0.10	64.88 ± 0.08	58.52 ± 0.46	64.54 ± 0.23
	ImageNet-32×32		65.22 ± 0.55	67.09 ± 0.18	66.52 ± 0.25	66.52 ± 0.15	58.32 ± 0.56	66.44 ± 0.17

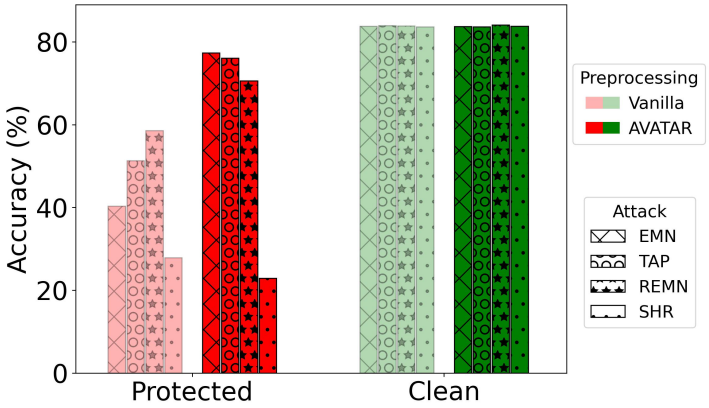


Fig. 6: Test accuracy for protected vs. clean identities in WebFace [69] facial recognition. The protected users protect their images using data-protecting perturbations. Our approach uses a diffusion model trained over the CelebA [37] dataset. For all the stealthy data-protecting perturbations our approach manages to recover the performance over protected data. The only exception is SHR, which according to Figure 8, leaves a noticeable trace over the image, rendering them not useful anymore.

data is allocated to a training set and the rest is the test set. We assume that the protected identities would add data-protecting perturbations to their images before sharing them. To this end, we use class-wise EMN [30], TAP [17], REMN [18], and SHR [71] with a perturbation radius of $\|\boldsymbol{\delta}\|_\infty \leq 16/255$. For perturbation generation using the first three attacks, we follow the settings of Huang *et al.* [30]. Specifically, we select 100 random identities from the CelebA [37] dataset and create an auxiliary dataset consisting of these 100 identities and the 50 protected WebFace [69] identities. Then, using these 150 identities we generate data protecting perturbations against a neural network with 150 classes. For SHR, however, we generate the data for all the 10572 WebFace identities and select the relevant data for protecting our above-mentioned 50 identities. Once we have the protected data, we train an InceptionResNet [59] facial recognition over the training set with or without our approach and evaluate the models over the test set. In our case, we assume that the malicious entity has access to a pre-trained diffusion model over CelebA [37] faces[8], and can run AVATAR over the protected data that it acquires from crawling the web. Since the WebFace photos are of size 112×112 but the diffusion model generates 256×256 images, we use bi-linear up- and down-sampling to connect the two. Like the CIFAR-10 experiments, here we also denoise the data with timestep set to 100. Samples of the WebFace dataset along with the protected data are shown in Figure 8. To evaluate the performance of our method, we test the models over the clean test set and record the recognition accuracy for both the protected and clean identities.[9]

As shown in Figure 6, AVATAR can recover the recognition accuracy over protected identities in all cases except the SHR [71] perturbations. The reason behind this might be two-fold. First, we are using a sub-optimal diffusion model as both the domain and, more importantly, size of the images have a mismatch. Second, looking at Figure 8, we see that while the SHR perturbations can protect the data, they trade the stealthiness of the original data due to their large patches. As such, the images would lose their utility. Now, the question is:

Can we protect the data using stealthy patterns without losing the data utility?

Interestingly, our theoretical result in Theorem 1 says that this might not be possible. According to Theorem 1, if the data curator wants to makes the denoising process harder, they need to increase the data-protecting perturbation. This increase is naturally at odds with the data utility, since by adding more powerful perturbations we lose the data utility.

V. Conclusion

In this paper, we introduced a countermeasure against data protection algorithms that use availability attacks. In particular, we show that by adding a controlled amount of Gaussian noise to the images and subsequently denoising them one can eliminate data-protecting perturbations. To this end, we use the forward and reverse diffusion processes of pre-trained models. We theoretically analyze our approach and show that the amount of Gaussian noise required to defuse the data-protecting perturbations is directly related to their norm. We conduct extensive experiments over various availability attacks. Our experiments demonstrate the superiority of our approach compared to adversarial training, setting a new SOTA defense against availability attacks. AVATAR demonstrates brittleness of availability attacks and calls for more research to protect personal data. Future work involves investigating the applicability of AVATAR to other models such as text-to-image generative models [52] and its relationship with techniques such as randomized smoothing [7].

[8]For this experiment, we use a pre-trained DDPM model over CelebA-HQ: https://github.com/ermongroup/SDEdit.

[9]Running the identity overlap removal of Wang *et al.* [65], we found that only 8 out of 50 protected identities had overlap between CelebA-HQ and WebFace. After removing these identities, we saw no major drop in the final performance of AVATAR.

Acknowledgments

This research was undertaken using the LIEF HPC-GPGPU Facility hosted at the University of Melbourne. This Facility was established with the assistance of LIEF Grant LE170100200. Sarah Erfani is in part supported by Australian Research Council (ARC) Discovery Early Career Researcher Award (DECRA) DE220100680. Moreover, this research was partially supported by the ARC Centre of Excellence for Automated Decision-Making and Society (CE200100005), and funded partially by the Australian Government through the Australian Research Council.

References

[1] Shekoofeh Azizi, Simon Kornblith, Chitwan Saharia, Mohammad Norouzi, and David J. Fleet. Synthetic data from diffusion models improves imagenet classification. *CoRR*, abs/2304.08466, 2023.

[2] Mauro Barni, Kassem Kallas, and Benedetta Tondi. A new backdoor attack in CNNs by training set corruption without label poisoning. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 101–105, 2019.

[3] Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, pages 1467–1474, 2012.

[4] Battista Biggio and Fabio Roli. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84:317–331, 2018.

[5] Abeba Birhane and Vinay Uday Prabhu. Large image datasets: A pyrrhic win for computer vision? In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1536–1546, 2021.

[6] Hyungjin Chung, Byeongsu Sim, and Jong Chul Ye. Come-closer-diffuse-faster: Accelerating conditional diffusion models for inverse problems through stochastic contraction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12403–12412, 2022.

[7] Jeremy Cohen, Elan Rosenfeld, and J. Zico Kolter. Certified adversarial robustness via randomized smoothing. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1310–1320, 2019.

[8] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *CoRR*, abs/2209.04747, 2022.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.

[10] Terrance Devries and Graham W. Taylor. Improved regularization of convolutional neural networks with cutout. *CoRR*, abs/1708.04552, 2017.

[11] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 8780–8794, 2021.

[12] Hadi M. Dolatabadi, Sarah M. Erfani, and Christopher Leckie. COLLIDER: A robust training framework for backdoor data. In *Proceedings of the 16th Asian Conference on Computer Vision (ACCV)*, pages 681–698, 2022.

[13] Hadi M. Dolatabadi, Sarah M. Erfani, and Christopher Leckie. ℓ_∞-robustness and beyond: Unleashing efficient adversarial training. In *Proceedings of the 17th European Conference on Computer Vision (ECCV)*, pages 467–483, 2022.

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, 2021.

[15] Ji Feng, Qi-Zhi Cai, and Zhi-Hua Zhou. Learning to confuse: Generating training time adversarial data with auto-encoder. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 11971–11981, 2019.

[16] Liam Fowl, Ping-yeh Chiang, Micah Goldblum, Jonas Geiping, Arpit Bansal, Wojtek Czaja, and Tom Goldstein. Preventing unauthorized use of proprietary data: Poisoning for secure dataset release. *CoRR*, abs/2103.02683, 2021.

[17] Liam Fowl, Micah Goldblum, Ping-yeh Chiang, Jonas Geiping, Wojciech Czaja, and Tom Goldstein. Adversarial examples make strong poisons. In *Proceedings of the Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 30339–30351, 2021.

[18] Shaopeng Fu, Fengxiang He, Yang Liu, Li Shen, and Dacheng Tao. Robust unlearnable examples: Protecting data privacy against adversarial learning. In *Proceedings of the 10th International Conference on Learning Representations (ICLR)*, 2022.

[19] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard S. Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

[20] Micah Goldblum, Dimitris Tsipras, Chulin Xie, Xinyun Chen, Avi Schwarzschild, Dawn Song, Aleksander Madry, Bo Li, and Tom Goldstein. Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 45(2):1563–1580, 2023.

[21] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial

nets. In *Proceedings of the Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 2672–2680, 2014.

[22] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015.

[23] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. BadNets: Identifying vulnerabilities in the machine learning model supply chain. *CoRR*, abs/1708.06733, 2017.

[24] Chuan Guo, Mayank Rana, Moustapha Cissé, and Laurens van der Maaten. Countering adversarial images using input transformations. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, 2018.

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[26] Kashmir Hill. The secretive company that might end privacy as we know it. *The New York Times*, 2020.

[27] Kashmir Hill and Aaron Krolik. How photos of your kids are powering surveillance technology. *The New York Times*, 2019.

[28] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

[29] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017.

[30] Hanxun Huang, Xingjun Ma, Sarah Monazam Erfani, James Bailey, and Yisen Wang. Unlearnable examples: Making personal data unexploitable. In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, 2021.

[31] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8107–8116, 2020.

[32] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 1885–1894, 2017.

[33] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Master's thesis, Department of Computer Science, University of Toronto, 2009.

[34] Sungbin Lim, Ildoo Kim, Taesup Kim, Chiheon Kim, and Sungwoong Kim. Fast autoaugment. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 6662–6672, 2019.

[35] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *Proceedings of the 13th European Conference on Computer Vision (ECCV)*, pages 740–755, 2014.

[36] Huayao Liu, Jiaming Zhang, Kailun Yang, Xinxin Hu, and Rainer Stiefelhagen. CMX: cross-modal fusion for RGB-X semantic segmentation with transformers. *CoRR*, abs/2203.04838, 2022.

[37] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3730–3738, 2015.

[38] Zhuoran Liu, Zhengyu Zhao, and Martha Larson. Image shortcut squeezing: Countering perturbative availability poisons with compression. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023.

[39] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, 2018.

[40] Luis Muñoz-González, Battista Biggio, Ambra Demontis, Andrea Paudice, Vasin Wongrassamee, Emil C. Lupu, and Fabio Roli. Towards poisoning of deep learning algorithms with back-gradient optimization. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security (AISec@CCS)*, pages 27–38, 2017.

[41] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NeurIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[42] Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Animashree Anandkumar. Diffusion models for adversarial purification. In *International Conference on Machine Learning (ICML)*, pages 16805–16827, 2022.

[43] Mang Ning, Enver Sangineto, Angelo Porrello, Simone Calderara, and Rita Cucchiara. Input perturbation reduces exposure bias in diffusion models. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 26245–26265, 2023.

[44] Quang-Cuong Pham. Analysis of discrete and hybrid stochastic systems by nonlinear contraction theory. In *Proceedings of the 10th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 1054–1059, 2008.

[45] Quang-Cuong Pham, Nicolas Tabareau, and Jean-Jacques E. Slotine. A contraction theory approach to stochastic incremental stability. *IEEE Transactions on Automatic Control*, 54(4):816–820, 2009.

[46] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. DreamFusion: Text-to-3D using 2D diffusion. In *Proceedings of the 11th International Conference*

on Learning Representations (ICLR), 2023.

[47] Evani Radiya-Dixit, Sanghyun Hong, Nicholas Carlini, and Florian Tramèr. Data poisoning won't save you from facial recognition. In *Proceedings of the 10th International Conference on Learning Representations (ICLR)*, 2022.

[48] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10674–10685, 2022.

[49] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[50] Pedro Sandoval-Segura, Vasu Singla, Jonas Geiping, Micah Goldblum, Tom Goldstein, and David W. Jacobs. Autoregressive perturbations for data poisoning. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[51] Avi Schwarzschild, Micah Goldblum, Arjun Gupta, John P. Dickerson, and Tom Goldstein. Just how toxic is data poisoning? A unified benchmark for backdoor and data poisoning attacks. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pages 9389–9398, 2021.

[52] Shawn Shan, Jenna Cryan, Emily Wenger, Haitao Zheng, Rana Hanocka, and Ben Y. Zhao. Glaze: Protecting artists from style mimicry by text-to-image models. In *Proceedings of the USENIX Security Symposium*, pages 2187–2204, 2023.

[53] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015.

[54] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-a-video: Text-to-video generation without text-video data. In *Proceedings of the 11th International Conference on Learning Representations (ICLR)*, 2023.

[55] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pages 2256–2265, 2015.

[56] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 11895–11907, 2019.

[57] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 12438–12448, 2020.

[58] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *Procceding of the 9th International Conference on Learning Representations (ICLR)*, 2021.

[59] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, pages 4278–4284, 2017.

[60] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*, 2014.

[61] Lue Tao, Lei Feng, Jinfeng Yi, Sheng-Jun Huang, and Songcan Chen. Better safe than sorry: Preventing delusive adversaries with adversarial training. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 16209–16225, 2021.

[62] Qi Tian, Kun Kuang, Kelu Jiang, Furui Liu, Zhihua Wang, and Fei Wu. ConfounderGAN: Protecting image data privacy with causal confounder. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[63] Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. In *Proceedings of the Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 8011–8021, 2018.

[64] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, 2019.

[65] Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 25(7):926–930, 2018.

[66] Zekai Wang, Tianyu Pang, Chao Du, Min Lin, Weiwei Liu, and Shuicheng Yan. Better diffusion models further improve adversarial training. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023.

[67] Yilun Xu, Shangyuan Tong, and Tommi S. Jaakkola. Stable target field for reduced variance score estimation. In *Proceedings of the 11th International Conference on Learning Representations (ICLR)*, 2023.

[68] Chaofei Yang, Qing Wu, Hai Li, and Yiran Chen. Generative poisoning attack method against neural networks. *CoRR*, abs/1703.01340, 2017.

[69] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z. Li.

Learning face representation from scratch. *CoRR*, abs/1411.7923, 2014.

[70] Jongmin Yoon, Sung Ju Hwang, and Juho Lee. Adversarial purification with score-based generative models. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pages 12062–12072, 2021.

[71] Da Yu, Huishuai Zhang, Wei Chen, Jian Yin, and Tie-Yan Liu. Availability attacks create shortcuts. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2367–2376, 2022.

[72] Chia-Hung Yuan and Shan-Hung Wu. Neural tangent generalization attacks. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pages 12230–12240, 2021.

[73] Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Seong Joon Oh, Youngjoon Yoo, and Junsuk Choe. CutMix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6022–6031, 2019.

[74] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2016.

[75] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, 2018.

[76] Hengtong Zhang, Jing Gao, and Lu Su. Data poisoning attacks against outcome interpretations of predictive models. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2165–2173, 2021.

[77] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pages 7472–7482, 2019.

[78] Wujie Zhou, Jinfu Liu, Jingsheng Lei, Lu Yu, and Jenq-Neng Hwang. GMNet: graded-feature multilabel-learning network for rgb-thermal urban scene semantic segmentation. *IEEE Transactions on Image Processing*, 30:7790–7802, 2021.

The Devil's Advocate: Shattering the Illusion of Unexploitable Data using Diffusion Models

Appendix A
Proofs

Here we provide our proof for Theorem 1. First, we provide the theoretical results that would be used in our proof. Then, we re-state Theorem 1 and provide its detailed proof. Our proofs heavily borrow from the contraction properties of stochastic difference equations [44, 45, 6].

Theorem 2 (Discrete stochastic contraction [44, 6])**.** *Let*

$$\boldsymbol{x}_{t-1} = \mathbf{h}(\boldsymbol{x}_t, t) + \sigma(\boldsymbol{x}_t, t)\boldsymbol{\epsilon}_t, \tag{8}$$

denote a stochastic difference equation where:

(a) $\mathbf{h} : \mathbb{R}^d \times \mathbb{N} \to \mathbb{R}^d$ *is a contraction mapping, i.e., for every* $t \in \mathbb{N}$ *there exists a* $\lambda_t \in [0, 1)$ *such that*

$$\|\mathbf{h}(\boldsymbol{x}, t) - \mathbf{h}(\boldsymbol{y}, t)\| \leq \lambda_t \|\boldsymbol{x} - \boldsymbol{y}\| \quad \forall\, \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d, \tag{9}$$

(b) $\sigma : \mathbb{R}^d \times \mathbb{N} \to \mathbb{R}$ *is a function such that for every* $t \in \mathbb{N}$ *and* $\boldsymbol{x} \in \mathbb{R}^d$

$$\mathrm{Tr}\big(\sigma(\boldsymbol{x}, t)\mathbf{I}\sigma(\boldsymbol{x}, t)\big) \leq C_t, \tag{10}$$

(c) and $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

Then, for two sample trajectories $\boldsymbol{x}_{t-1}$ *and* $\bar{\boldsymbol{x}}_{t-1}$ *that satisfy Equation (8) we have:*

$$\mathbb{E}\left[\|\boldsymbol{x}_{t-1} - \bar{\boldsymbol{x}}_{t-1}\|^2\right] \leq \lambda_t^2\, \mathbb{E}\left[\|\boldsymbol{x}_t - \bar{\boldsymbol{x}}_t\|^2\right] + 2C_t. \tag{11}$$

Using Theorem 2 and Equation (5) we can get the following result [6].

Corollary 2.1. *The reverse diffusion process of DDPMs are contracting stochastic difference equations.*

Proof. Our proof closely follows that of Chung *et al.* [6]. Specifically, we need to show that for the reverse diffusion process given in Equation (5), the conditions of Equations (9) and (10) hold. To show this, note that if we set:

$$\mathbf{h}(\boldsymbol{x}_t, t) = \frac{1}{\sqrt{1-\beta_t}}\left(\boldsymbol{x}_t + \beta_t \mathbf{s}_{\boldsymbol{\phi}}(\boldsymbol{x}_t, t)\right)$$

and

$$\sigma(\boldsymbol{x}_t, t) = \sqrt{\beta_t}$$

then Equations (5) and (8) coincide. Using Lemma A.1. from Chung *et al.* [6], one can show that for

$$\lambda_t = \sqrt{1-\beta_t}\frac{1-\alpha_{t-1}}{1-\alpha_t} \tag{12}$$

and

$$C_t = d\beta_t \tag{13}$$

the conditions of Equations (9) and (10) are satisfied. As such, for two reverse sample trajectories $\boldsymbol{x}_{t-1}$ and $\bar{\boldsymbol{x}}_{t-1}$ that satisfy the reverse diffusion process of Equation (5), Equation (11) holds. □

Next, we present two lemmas that are going to be used in our proof of Theorem 1.

Lemma 1 ([6])**.** *For* λ_t*'s given in Equation (12) the following holds:*

$$\prod_{s=1}^{t^*} \lambda_s^2 \leq \exp(-\frac{t^*\beta_{t^*}}{2}). \tag{14}$$

Proof. See Lemma C.1. in [6]. □

Lemma 2. *For two random vectors* $\mathbf{x}$ *and* $\mathbf{y}$ *we have:*

$$\mathbb{E}\left[\|\mathbf{x}+\mathbf{y}\|^2\right] \leq 2\,\mathbb{E}\left[\|\mathbf{x}\|^2\right] + 2\,\mathbb{E}\left[\|\mathbf{y}\|^2\right]. \tag{15}$$

Proof. We know that:

$$\begin{aligned}\mathbb{E}\left[\|\mathbf{x}+\mathbf{y}\|^2\right] &= \mathbb{E}\left[\|\mathbf{x}\|^2\right] + \mathbb{E}\left[\|\mathbf{y}\|^2\right] + 2\ \mathbb{E}\left[\mathbf{x}^\top\mathbf{y}\right] \\ &\leq 2\ \mathbb{E}\left[\|\mathbf{x}\|^2\right] + 2\ \mathbb{E}\left[\|\mathbf{y}\|^2\right],\end{aligned}$$

where the last inequality follows from the fact that $\mathbb{E}\left[\|\mathbf{x}-\mathbf{y}\|^2\right] \geq 0$. □

We are now ready to prove our theoretical result.

Theorem 1 (restated)**.** *Let $\boldsymbol{x} \in \mathbb{R}^d$ denote a clean image and $\tilde{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{\delta}$ its protected version, where $\boldsymbol{\delta}$ denotes any arbitrary data protection perturbation. Also, let $\bar{\boldsymbol{x}}_0$ be the sanitized image using the* AVATAR *denoising process given in Equations (6) and (7). If we set t^* such that*

$$2\log\left(\frac{2\|\boldsymbol{\delta}\|^2 + 4d}{\mu\Delta}\right) \leq t^*\beta_{t^*} \leq \frac{\mu\Delta}{4d},$$

then the estimation error between the sanitized $\bar{\boldsymbol{x}}_0$ and clean image $\boldsymbol{x}$ can be bounded as:

$$\mathbb{E}\left[\|\bar{\boldsymbol{x}}_0 - \boldsymbol{x}\|^2\right] \leq 2(\mu+1)\Delta,$$

where $\Delta = \mathbb{E}[\|\boldsymbol{x}_0 - \boldsymbol{x}\|^2]$ and $\mu > 0$ is a constant.

Proof. We are looking to find an upper-bound for the estimation error between the sanitized image and its clean version. Using Lemma 2 we can write:

$$\begin{aligned}\mathbb{E}\left[\|\bar{\boldsymbol{x}}_0 - \boldsymbol{x}\|^2\right] &= \mathbb{E}\left[\|(\bar{\boldsymbol{x}}_0 - \boldsymbol{x}_0) + (\boldsymbol{x}_0 - \boldsymbol{x})\|^2\right] \\ &\leq 2\ \mathbb{E}\left[\|\bar{\boldsymbol{x}}_0 - \boldsymbol{x}_0\|^2\right] + 2\ \mathbb{E}\left[\|\boldsymbol{x}_0 - \boldsymbol{x}\|^2\right] \\ &\leq 2\ \mathbb{E}\left[\|\bar{\boldsymbol{x}}_0 - \boldsymbol{x}_0\|^2\right] + 2\ \Delta. \end{aligned} \tag{16}$$

Now, we need to find an upper-bound for the first term. To this end, we are going to use the contraction property of the DDPMs (Corollary 2.1). In particular, given the noisy versions of the clean $\boldsymbol{x}$ and the protected image $\tilde{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{\delta}$, in other words:

$$\begin{aligned}\boldsymbol{x}_{t^*} &= \sqrt{\alpha_{t^*}}\boldsymbol{x} + \sqrt{1-\alpha_{t^*}}\boldsymbol{\epsilon}_0 \\ \bar{\boldsymbol{x}}_{t^*} &= \sqrt{\alpha_{t^*}}\tilde{\boldsymbol{x}} + \sqrt{1-\alpha_{t^*}}\boldsymbol{\epsilon}'_0,\end{aligned} \tag{17}$$

we know that both $\boldsymbol{x}_0$ and $\bar{\boldsymbol{x}}_0$ satisfy the reverse diffusion process, or:

$$\begin{aligned}\boldsymbol{x}_{t-1} &= \frac{1}{\sqrt{1-\beta_t}}\left(\boldsymbol{x}_t + \beta_t \mathbf{s}_{\boldsymbol{\phi}}(\boldsymbol{x}_t, t)\right) + \sqrt{\beta_t}\boldsymbol{\epsilon}_t \\ \bar{\boldsymbol{x}}_{t-1} &= \frac{1}{\sqrt{1-\beta_t}}\left(\bar{\boldsymbol{x}}_t + \beta_t \mathbf{s}_{\boldsymbol{\phi}}(\bar{\boldsymbol{x}}_t, t)\right) + \sqrt{\beta_t}\boldsymbol{\epsilon}'_t, \quad \forall\ t \in \{1, 2, \ldots, t^*\},\end{aligned} \tag{18}$$

where $\boldsymbol{\epsilon}_t, \boldsymbol{\epsilon}'_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. As such, we can treat $\boldsymbol{x}_0$ and $\bar{\boldsymbol{x}}_0$ as two sample trajectories of the same stochastic difference equation. Thus, by recursively applying Equation (11) we would get:

$$\mathbb{E}\left[\|\bar{\boldsymbol{x}}_0 - \boldsymbol{x}_0\|^2\right] \leq \mathbb{E}\left[\|\bar{\boldsymbol{x}}_{t^*} - \boldsymbol{x}_{t^*}\|^2\right]\prod_{s=1}^{t^*}\lambda_s^2 + 2\sum_{s=1}^{t^*} C_s \prod_{r=1}^{s-1}\lambda_r^2. \tag{19}$$

Now, let us consider each term on the RHS of Equation (19) separately. For the red term, we can write:

$$\begin{aligned}\mathbb{E}\left[\|\bar{\boldsymbol{x}}_{t^*} - \boldsymbol{x}_{t^*}\|^2\right] &\overset{(1)}{=} \mathbb{E}\left[\left\|\sqrt{\alpha_{t^*}}(\tilde{\boldsymbol{x}} - \boldsymbol{x}) + \sqrt{1-\alpha_{t^*}}(\boldsymbol{\epsilon}'_0 - \boldsymbol{\epsilon}_0)\right\|^2\right] \\ &\overset{(2)}{=} \mathbb{E}\left[\left\|\sqrt{\alpha_{t^*}}\boldsymbol{\delta} + \sqrt{1-\alpha_{t^*}}(\boldsymbol{\epsilon}'_0 - \boldsymbol{\epsilon}_0)\right\|^2\right] \\ &= \|\sqrt{\alpha_{t^*}}\boldsymbol{\delta}\|^2 + \mathbb{E}\left[\left\|\sqrt{1-\alpha_{t^*}}(\boldsymbol{\epsilon}'_0 - \boldsymbol{\epsilon}_0)\right\|^2\right] + 2\sqrt{\alpha_{t^*}}\sqrt{1-\alpha_{t^*}}\boldsymbol{\delta}^\top\mathbb{E}\left[\boldsymbol{\epsilon}'_0 - \boldsymbol{\epsilon}_0\right] \\ &\overset{(3)}{=} \alpha_{t^*}\|\boldsymbol{\delta}\|^2 + (1-\alpha_{t^*})\mathbb{E}\left[\|(\boldsymbol{\epsilon}'_0 - \boldsymbol{\epsilon}_0)\|^2\right].\end{aligned} \tag{20}$$

where (1) is derived from Equation (17), (2) holds since $\tilde{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{\delta}$, and (3) is valid as $\boldsymbol{\epsilon}_0, \boldsymbol{\epsilon}'_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Given that:

$$\boldsymbol{\epsilon}'_0 - \boldsymbol{\epsilon}_0 \sim \mathcal{N}(\mathbf{0}, 2\mathbf{I}),$$

we can simplify Equation (20) as:

$$\mathbb{E}\left[\|\bar{\boldsymbol{x}}_{t^*} - \boldsymbol{x}_{t^*}\|^2\right] = \alpha_{t^*} \|\boldsymbol{\delta}\|^2 + 2(1-\alpha_{t^*})\mathbb{E}\left[\chi\right],$$

where χ follows the chi-squared distribution with d degrees of freedom. Using the fact that $0 < \alpha_{t^*} < 1$, we can finally write:

$$\begin{aligned}\mathbb{E}\left[\|\bar{\boldsymbol{x}}_{t^*} - \boldsymbol{x}_{t^*}\|^2\right] &= \alpha_{t^*} \|\boldsymbol{\delta}\|^2 + 2(1-\alpha_{t^*})d \\ &\leq \|\boldsymbol{\delta}\|^2 + 2d. \end{aligned} \tag{21}$$

Using Lemma 1, for the blue term in Equation (19) we can write:

$$\prod_{s=1}^{t^*} \lambda_s^2 \leq \exp(-\frac{t^*\beta_{t^*}}{2}). \tag{22}$$

Finally, for the green term we have:

$$\begin{aligned}2\sum_{s=1}^{t^*} C_s \prod_{r=1}^{s-1} \lambda_r^2 &\overset{(1)}{=} 2\sum_{s=1}^{t^*} d\beta_s \prod_{r=1}^{s-1} \lambda_r^2 \\ &\overset{(2)}{\leq} 2\sum_{s=1}^{t^*} d\beta_s \\ &\overset{(3)}{\leq} 2dt^*\beta_{t^*}. \end{aligned} \tag{23}$$

Here, (1) is the result of Equation (13), (2) holds since $0 < \lambda_r < 1$ (see Equation (12)), and (3) is derived from $0 < \beta_1 < \cdots < \beta_t < 1$.

Putting Equations (21) to (23) together, we have:

$$\mathbb{E}\left[\|\bar{\boldsymbol{x}}_0 - \boldsymbol{x}_0\|^2\right] \leq \left(\|\boldsymbol{\delta}\|^2 + 2d\right) \exp(-\frac{t^*\beta_{t^*}}{2}) + 2dt^*\beta_{t^*}. \tag{24}$$

Given that:

$$2\log\left(\frac{2\|\boldsymbol{\delta}\|^2 + 4d}{\mu\Delta}\right) \leq t^*\beta_{t^*} \leq \frac{\mu\Delta}{4d},$$

we can simplify Equation (24) as:

$$\begin{aligned}\mathbb{E}\left[\|\bar{\boldsymbol{x}}_0 - \boldsymbol{x}_0\|^2\right] &\leq \left(\|\boldsymbol{\delta}\|^2 + 2d\right) \exp(-\frac{t^*\beta_{t^*}}{2}) + 2dt^*\beta_{t^*} \\ &\leq \left(\|\boldsymbol{\delta}\|^2 + 2d\right) \frac{\mu\Delta}{2\|\boldsymbol{\delta}\|^2 + 4d} + 2d\frac{\mu\Delta}{4d} \\ &\leq \mu\Delta. \end{aligned} \tag{25}$$

Replacing Equation (25) into Equation (16), the proof can be completed. □

APPENDIX B
ADDITIONAL EXPERIMENTAL RESULTS

In this section, we provide additional experiments and insights that were omitted from the main paper due to space limitations.

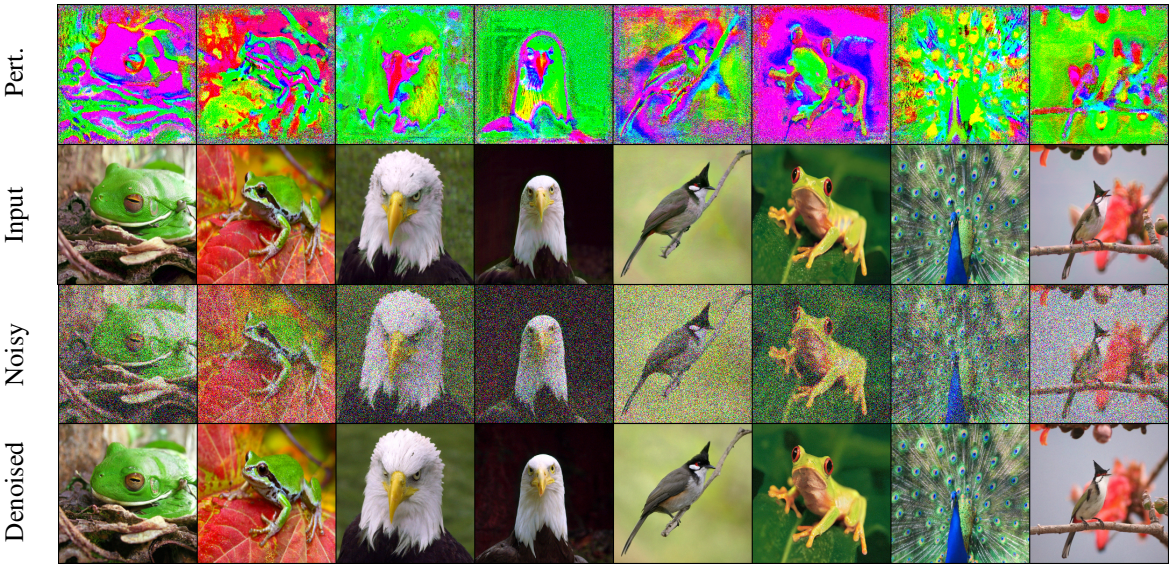
A. Denoising Samples

Figures 7 and 8 include samples from the protected IN-100 and WebFace datasets alongside their denoised ones. As seen, AVATAR can successfully recover the benign data except cases where the perturbations are sever enough to remain visible. In these cases, however, the protected data has lost its normal utility due to the visibility of the protecting perturbation.

(a) NTGA [72]

(b) EMN [30]

(c) TAP [17]

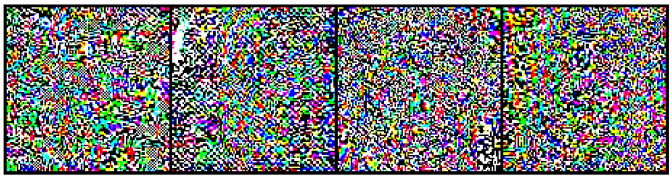


(d) REMN [18]

(e) SHR [71]

Fig. 7: Samples from IN-100 dataset. For each attack, we show the perturbation, the protected image, the noisy version of the image, and the denoised one using AVATAR.

Clean Image



Perturbation

Protected Image

Denoised Image

(a) EMN [30]

Perturbation

Protected Image

Denoised Image

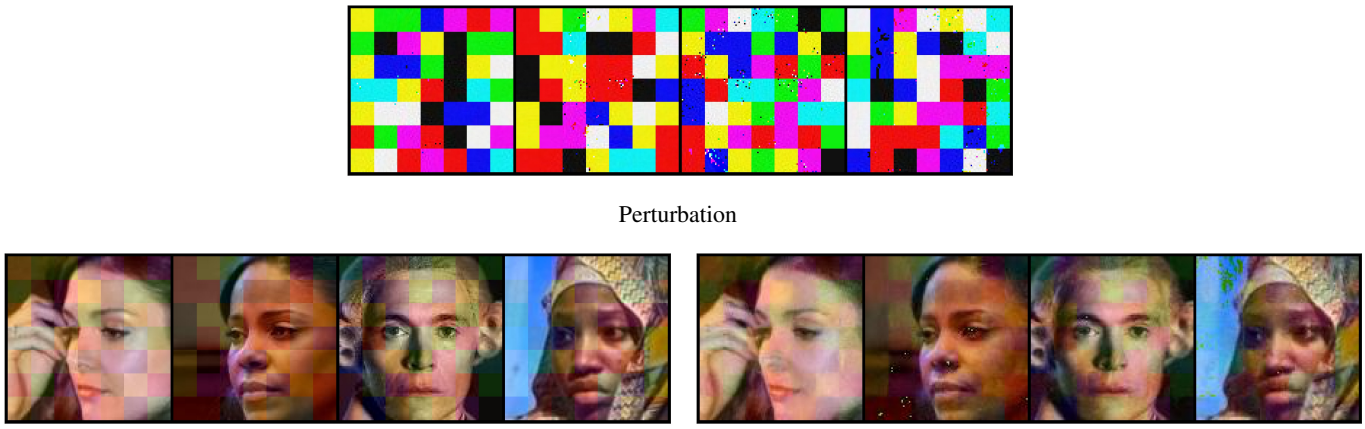
(b) TAP [17]

Perturbation

Protected Image

Denoised Image

(c) REMN [30]



(d) SHR [71]

Fig. 8: Samples from the WebFace [69] dataset. In each case, we generate the data-protecting perturbations with a maximum magnitude of $16/255$. For denoising, we use AVATAR based on a diffusion model pre-trained on the CelebA [37] dataset. As seen, the SHR [71] perturbations leave a noticeable pattern over the protected data, which put their utility (e.g., posting over the social media) under question. Nevertheless, since they have a more global structure, they remain persistent even after denoising.

B. Extended Experimental Results over Different Architectures

In Table V, we presented our results on training RN-18 models over protected data. To show the applicability of our approach across various architectures, we also report our results for three additional architectures, namely DN-121, VGG-16, and WRN-34, in Table X. Similar to our RN-18 experiments, AVATAR delivers the best performance against protected data.

C. On Selecting Diffusion Step t^*

In Figure 4, we demonstrated that setting $t^* = 100$ delivers a consistent performance across different architectures. However, chances are that practitioners may want to replace the diffusion model used in AVATAR with one of their own. In such cases, the diffusion model might have different characteristics compared to the ones used in this paper. In this part, we present two methods for setting t^*.

1) Using the α_t Curves: A naïve approach in selecting a suitable diffusion timestep t^* is using the α_t curves between the new diffusion model and a reference model. Specifically, since the value of α_t in Equation (3) controls the amount of disruptive noise, we can use the value of α_t to guide our hyper-parameter selection. To this end, we can find an equivalent t^* such that the value of α_{t^*} is set to an acceptable value. This is because if too much disruptive noise is required to be added to the data to counteract the protecting perturbation, it means that the data has already been corrupted so much that it has lost its utility in the first place.

We demonstrate this approach for selecting the timestep t^* for our IN-1k-32×32 experiments in Table IX. As discussed in Section IV-H, for this new experiment we want to use a guided diffusion model (DDPM-IP [43]) which uses a cosine schedule for sampling. As per our prior experience, we know that an acceptable value for t^* using a linear scheduler is 100. As such, we can draw the α_t curve for both cases, and find an equivalent t^* for the cosine scheduler in DDPM-IP. As shown in Figure 9, we can see that in this new case we should set $t^* = 200$ to get an equivalent α_t as the one which we previously used for the CIFAR-10 experiments.

2) Using Reconstruction Quality: Another approach to set a viable value for the diffusion timestep t^* is through controlling a desirable reconstruction quality. Recall that the goal of availability attacks is to preserve the normal utility of the data. As such, they usually aim to add imperceptible perturbations to the data. This assumption can help us in selecting a good value for t^*. In particular, having a small portion of clean data, we can run the denoising process of AVATAR on these benign data and record a reconstruction Peak-to-Signal-Noise-Ratio (PSNR) for different values of t^*. In general, as we move towards larger t^*, the PSNR drops. We can set an acceptable level of PSNR value, for example 22dB, to select t^*. Beyond that, the PSNR drops so significantly that both the clean and protected data become unreasonably noisy, losing their utility.

To demonstrate this point through our IN-1k-32×32 experiments in Table IX, we have reported AVATAR's reconstruction PSNR for different values of t in Table XI. As seen, while $t^* = 100$ reaches a PSNR value of 22.52dB when we use a linear

scheduler for sampling, we can still get a reasonable PSNR of 23.71dB for $t^* = 200$ in DDPM-IP. Therefore, we can pick $t^* = 200$ for denoising using the IN-1k-32×32 model.

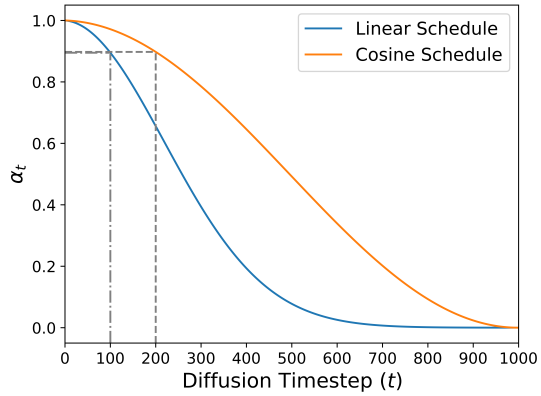


Fig. 9: The α_t curve for linear vs. cosine sampling schedulers.

TABLE XI: Reconstruction PSNR for clean CIFAR-10 dataset. The mean and standard deviation are computed over 1000 random samples as the validation set.

Scheduler	**Diffusion Timestep (t)**			
	50	100	150	200
Linear	25.19 ± 4.16	22.52 ± 3.32	20.83 ± 2.92	19.54 ± 2.61
Cosine	31.20 ± 1.50	27.49 ± 1.51	25.27 ± 1.50	23.71 ± 1.50

D. Additional Experimental Results over Different Combination of Availability Attacks

A scenario that might happen in the real-world is that different classes use a different type of protection. To simulate this scenario, we choose five of the best performing availability attacks, namely CON (C), NTGA (N), TAP (T), REMN (R), and SHR (S), based on our results in Table V to protect four classes of the CIFAR-10 dataset. We create different combinations of these five attacks to protect the four classes, resulting in five distinct combinations which we name CNTR, NTRS, RSCN, SCNT, and TRSC. We use AVATAR to defuse the entire dataset, which includes both protected and unprotected classes. To this end, we use our setting from Section IV-H and use DDPM-IP models pre-trained over the IN-1k-32×32 dataset. We then train RN-18 models over protected and defused data. Our results have been reported as confusion matrices in Figure 11. As seen, our model is attack-agnostic and can revive the normal data.

E. Additional Experimental Results over Different Perturbation Norms

Another interesting use-case might happen when different classes use a different perturbation norm to protect their data. We designed an experiment on CIFAR-10 to test this case. For these experiments, we first choose four classes of the CIFAR-10 randomly and aim to protect them with availability attacks. We then use four distinct levels of protection, from $\varepsilon = 4$ to $\varepsilon = 32$, to protect these selected classes. Figure 10 shows a few samples for each of the availability attacks used in this scenario. Like the previous experiment, again we use our settings from Section IV-H to run these experiments. As seen in the confusion matrices of Figure 12, AVATAR performance decreases as we increase the perturbation norm. This is in line with our theoretical insights: to protect the data against AVATAR, we need larger perturbations. However, a larger perturbation means losing the regular utility of the data.

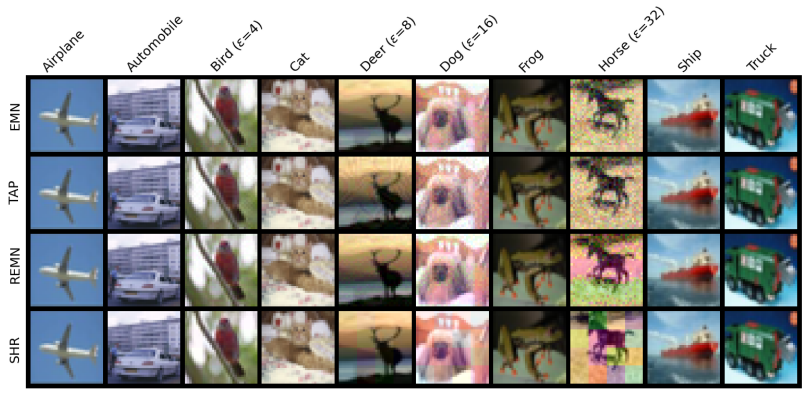
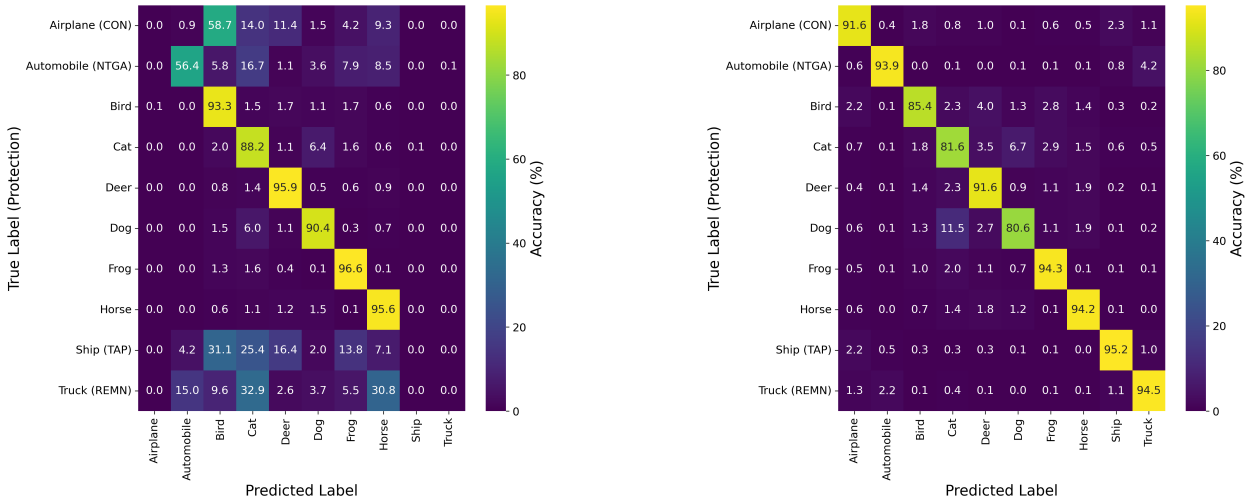


Fig. 10: Samples from protected CIFAR-10 datasets with four different availability attacks. We have selected four classes to protect, where in each case we use a different perturbation norm to protect the data. The protecting perturbations become extremely visible as we increase their norm.

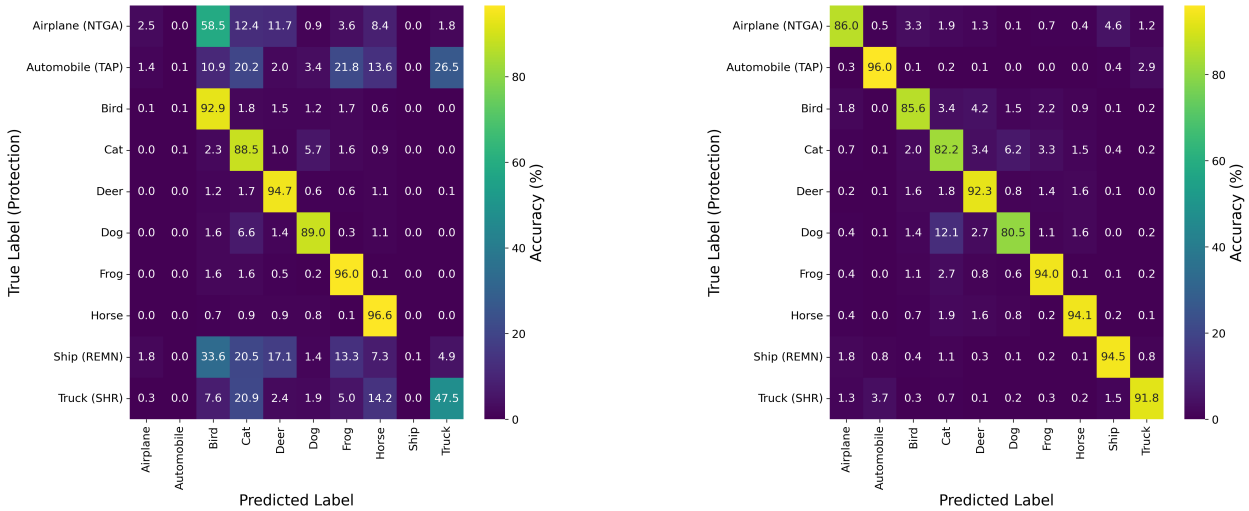
TABLE X: Test accuracy (%) of various neural network architectures trained over data availability attacks on CIFAR-10, CIFAR-100, SVHN, and ImageNet-100 datasets without and with our denoising approach. The mean and standard deviation are computed over 5 seeds.

Data	Model	Method	Clean	Data Availability Attacks					
				NTGA	EMN	TAP	REMN	SHR	AR
CIFAR-10	RN-18	Vanilla	94.50 ± 0.09	11.49 ± 0.69	24.85 ± 0.71	7.86 ± 0.90	20.50 ± 1.16	10.82 ± 0.22	12.09 ± 1.12
		AVATAR		87.95 ± 0.28	90.95 ± 0.10	90.71 ± 0.19	88.49 ± 0.24	85.69 ± 0.27	91.57 ± 0.18
	VGG-16	Vanilla	93.22 ± 0.05	11.02 ± 0.15	25.82 ± 0.74	9.42 ± 1.03	20.58 ± 0.96	10.88 ± 0.28	11.49 ± 1.23
		AVATAR		87.13 ± 0.27	89.71 ± 0.16	89.38 ± 0.17	87.17 ± 0.22	84.74 ± 0.30	90.21 ± 0.15
	DN-121	Vanilla	94.62 ± 0.14	11.63 ± 1.08	24.73 ± 0.81	11.09 ± 1.69	19.02 ± 1.21	14.57 ± 1.36	13.67 ± 1.91
		AVATAR		88.33 ± 0.35	91.33 ± 0.21	91.24 ± 0.16	88.83 ± 0.24	86.12 ± 0.29	91.77 ± 0.15
	WRN-34	Vanilla	93.87 ± 0.07	10.72 ± 0.37	22.31 ± 1.55	8.59 ± 0.43	19.71 ± 0.95	10.33 ± 0.14	12.09 ± 0.43
		AVATAR		87.64 ± 0.19	90.19 ± 0.21	89.64 ± 0.24	87.64 ± 0.28	85.22 ± 0.42	90.71 ± 0.14
SVHN	RN-18	Vanilla	96.29 ± 0.12	9.65 ± 0.70	9.13 ± 2.00	65.97 ± 1.99	11.55 ± 0.19	10.59 ± 3.98	6.76 ± 0.07
		AVATAR		89.84 ± 0.32	93.84 ± 0.12	93.35 ± 0.10	88.51 ± 0.23	83.82 ± 0.39	94.13 ± 0.17
	VGG-16	Vanilla	95.94 ± 0.12	23.24 ± 8.42	9.13 ± 2.00	63.81 ± 2.77	10.87 ± 0.43	9.45 ± 3.69	10.87 ± 5.09
		AVATAR		89.75 ± 0.19	93.61 ± 0.14	93.03 ± 0.26	87.01 ± 0.41	82.03 ± 0.48	93.73 ± 0.14
	DN-121	Vanilla	96.47 ± 0.12	19.06 ± 5.84	30.49 ± 5.53	69.04 ± 1.80	11.48 ± 2.09	10.54 ± 3.45	10.23 ± 3.64
		AVATAR		90.52 ± 0.13	94.39 ± 0.24	94.05 ± 0.18	88.76 ± 0.53	84.35 ± 0.71	94.61 ± 0.10
	WRN-34	Vanilla	96.35 ± 0.07	23.24 ± 8.42	18.57 ± 7.00	70.72 ± 1.23	18.10 ± 10.17	6.84 ± 0.15	9.04 ± 2.47
		AVATAR		90.26 ± 0.16	94.29 ± 0.12	94.01 ± 0.28	89.34 ± 0.35	83.50 ± 0.47	94.60 ± 0.12
CIFAR-100	RN-18	Vanilla	75.01 ± 0.41	1.32 ± 0.31	2.05 ± 0.18	14.10 ± 0.19	10.88 ± 0.33	1.39 ± 0.10	2.15 ± 0.46
		AVATAR		63.98 ± 0.55	65.73 ± 0.36	64.99 ± 0.10	64.88 ± 0.08	58.52 ± 0.46	64.54 ± 0.23
	VGG-16	Vanilla	72.03 ± 0.06	1.16 ± 0.03	1.94 ± 0.19	15.25 ± 0.57	9.11 ± 0.54	1.57 ± 0.33	2.42 ± 0.38
		AVATAR		62.00 ± 0.34	63.60 ± 0.26	62.69 ± 0.21	62.38 ± 0.30	56.38 ± 0.41	62.27 ± 0.34
	DN-121	Vanilla	77.47 ± 0.33	1.87 ± 0.34	2.41 ± 0.40	15.94 ± 0.25	8.94 ± 0.49	1.81 ± 0.23	2.34 ± 0.48
		AVATAR		65.84 ± 0.41	67.86 ± 0.22	67.27 ± 0.18	66.85 ± 0.20	60.16 ± 0.24	66.78 ± 0.43
	WRN-34	Vanilla	73.39 ± 0.43	1.51 ± 0.18	1.94 ± 0.24	12.00 ± 0.45	9.16 ± 0.61	1.37 ± 0.19	1.90 ± 0.24
		AVATAR		61.62 ± 0.36	63.49 ± 0.32	62.68 ± 0.34	62.52 ± 0.32	56.64 ± 0.60	62.57 ± 0.26
ImageNet-100	RN-18	Vanilla	80.05 ± 0.13	74.74 ± 0.52	1.78 ± 0.17	9.14 ± 0.40	13.28 ± 0.51	43.48 ± 1.56	
		AVATAR		71.08 ± 0.48	72.84 ± 0.90	76.52 ± 0.46	39.79 ± 0.98	59.85 ± 1.01	
	VGG-16	Vanilla	79.16 ± 0.30	73.98 ± 0.45	1.54 ± 0.24	9.30 ± 0.39	12.04 ± 0.15	72.10 ± 0.68	
		AVATAR		70.78 ± 0.56	73.34 ± 0.50	76.25 ± 0.19	39.74 ± 0.44	67.22 ± 0.54	
	DN-121	Vanilla	79.66 ± 0.25	75.16 ± 0.22	4.58 ± 0.31	19.28 ± 0.84	15.46 ± 0.77	31.47 ± 1.01	
		AVATAR		73.41 ± 0.23	75.96 ± 0.48	77.96 ± 0.48	48.20 ± 0.32	54.24 ± 0.65	
	WRN-34	Vanilla	74.46 ± 0.52	68.38 ± 1.17	1.22 ± 0.15	8.20 ± 0.28	10.45 ± 0.28	52.95 ± 4.13	
		AVATAR		64.20 ± 0.76	66.33 ± 1.01	70.34 ± 0.85	29.89 ± 0.66	58.24 ± 1.47	



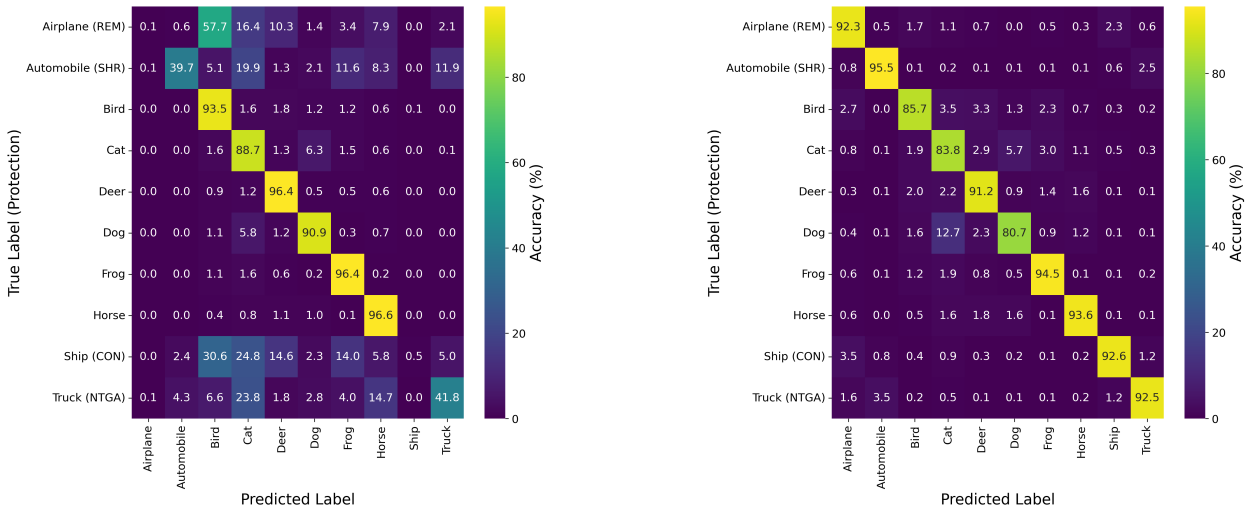
Vanilla Training AVATAR

(a) CNTR



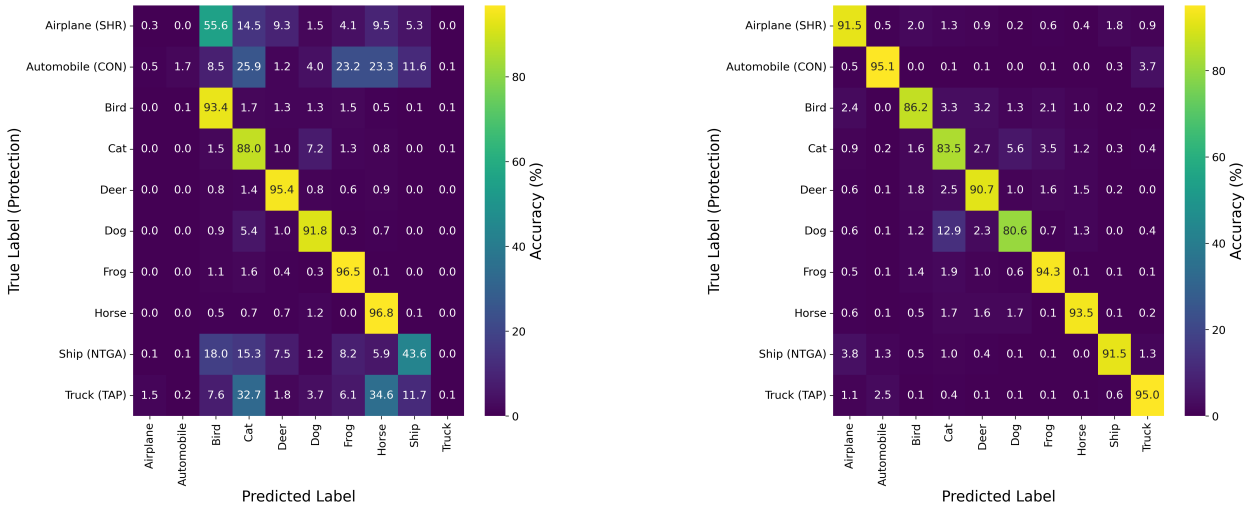
Vanilla Training AVATAR

(b) NTRS



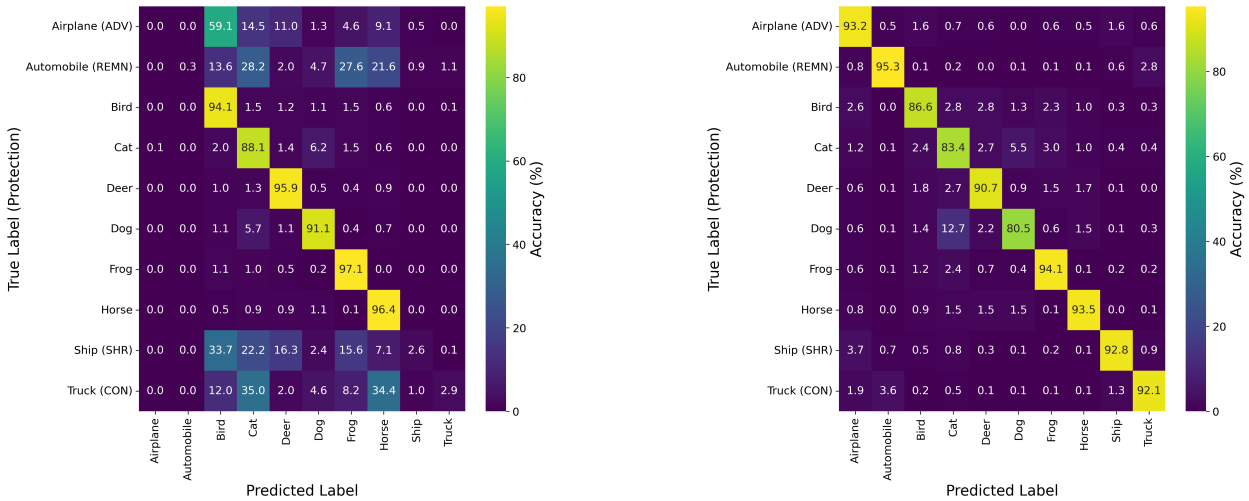
Vanilla Training AVATAR

(c) RSCN



Vanilla Training AVATAR

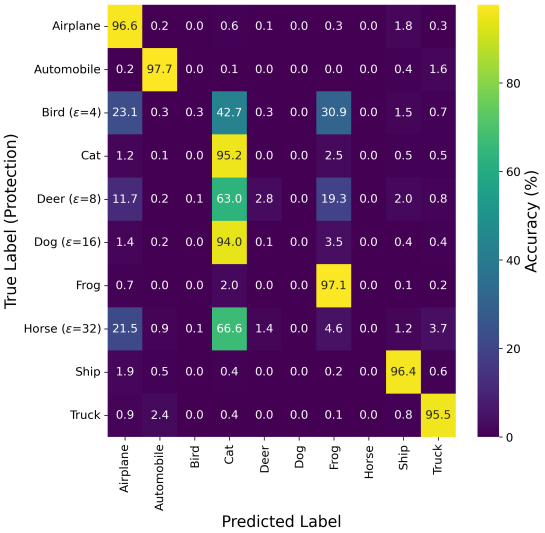
(d) SCNT



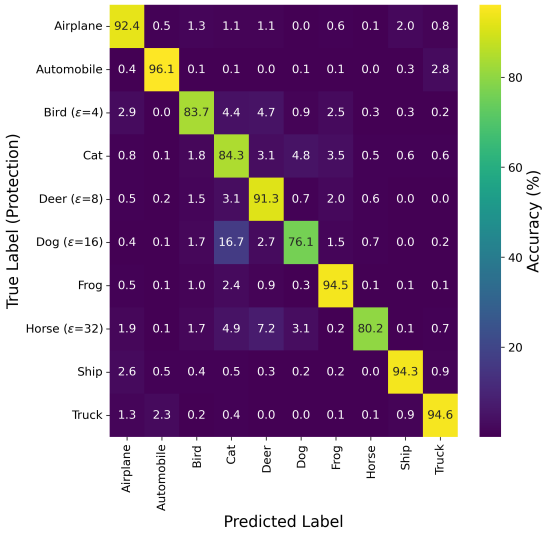
Vanilla Training AVATAR

(e) TRSC

Fig. 11: The confusion matrices of RN-18 classifiers trained over CIFAR-10 dataset. In each case, we use a different combination of availability attacks to protect some of the classes. For AVATAR, we follow our settings for the experiments in Table IX and use a DDPM-IP pre-trained over the IN-1k-32×32 dataset.

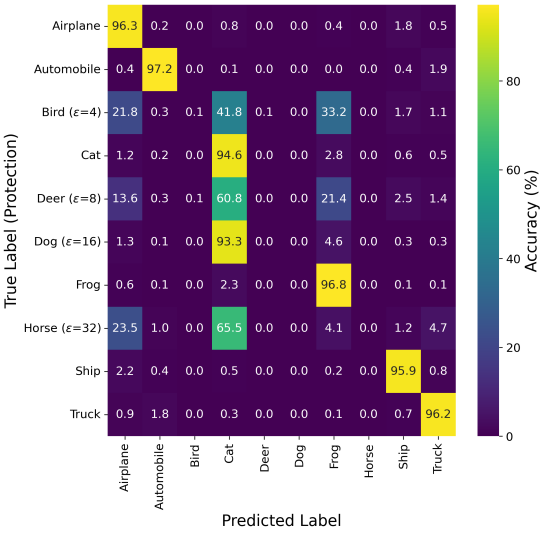


Vanilla Training

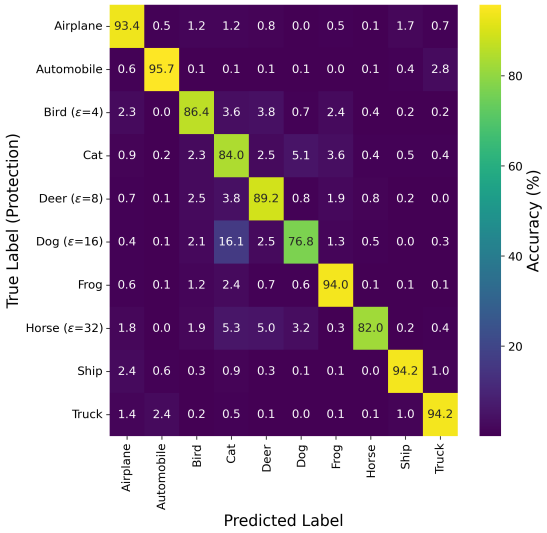


AVATAR

(a) EMN [30]

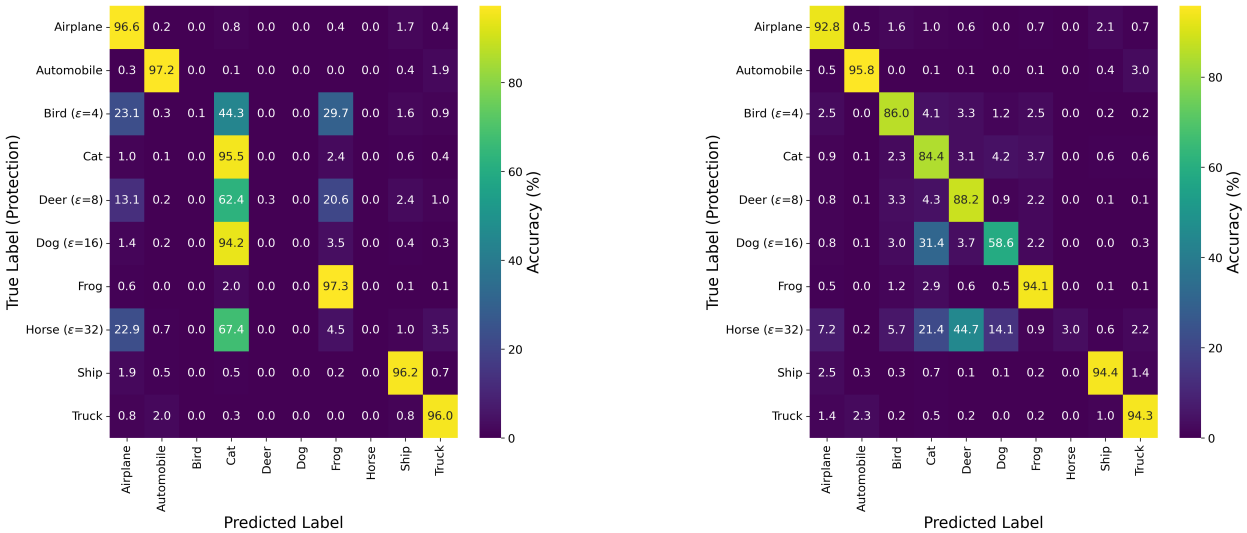


Vanilla Training



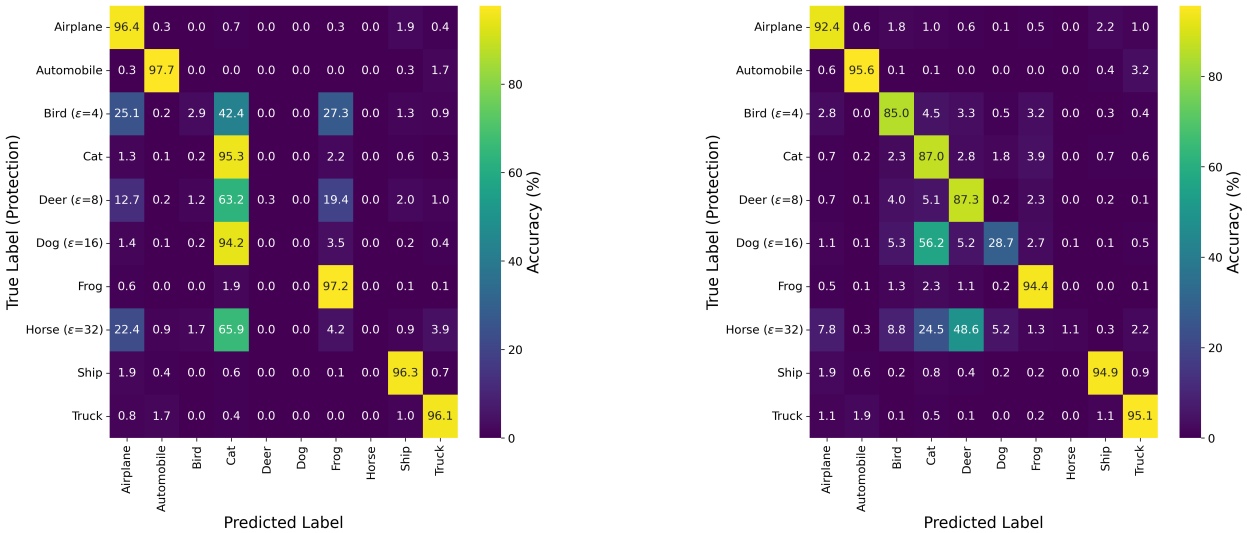
AVATAR

(b) TAP [17]



Vanilla Training AVATAR

(c) REMN [18]



Vanilla Training AVATAR

(d) SHR [71]

Fig. 12: The confusion matrices of RN-18 classifiers trained over CIFAR-10 dataset. For each availability attack, we use four different perturbation norms to protect a randomly selected set of classes. For AVATAR, we follow our settings for the experiments in Table IX and use a DDPM-IP pre-trained over the IN-1k-32×32 dataset.