Pre-trained Multiple Latent Variable Generative Models are good defenders against Adversarial Attacks

Dario Serez[1,3] Marco Cristani[2] Alessio Del Bue[1] Vittorio Murino[1,2,3] Pietro Morerio[1]
[1]Istituto Italiano di Tecnologia, Italy [2]University of Verona, Italy [3]University of Genoa, Italy
{dario.serez, alessio.delbue, vittorio.murino, pietro.morerio}@iit.it
marco.cristani@univr.it

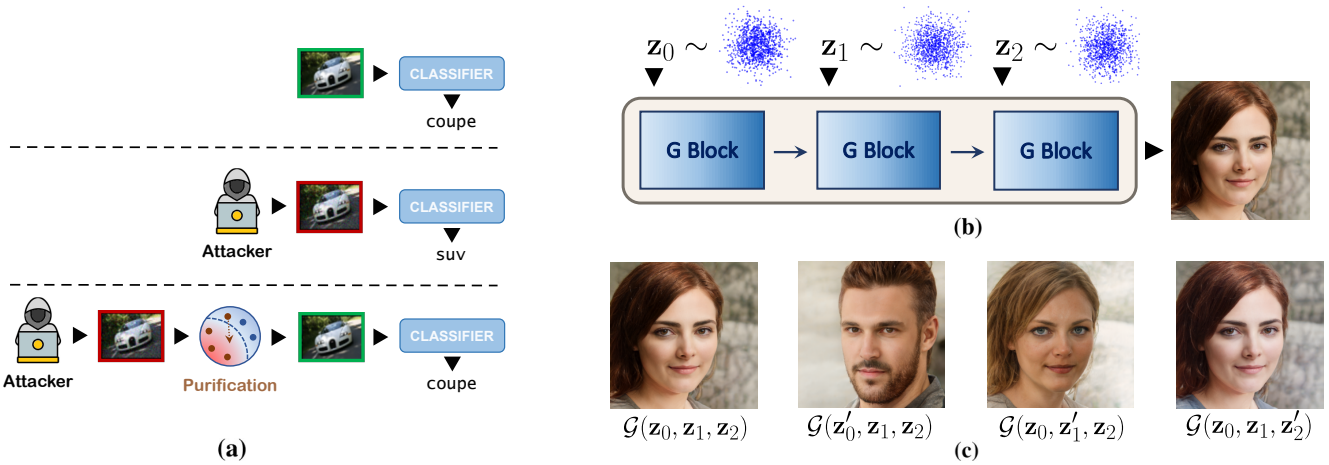


Figure 1. **(a)**: Overview of the adversarial attack and purification mechanism. The attacker subtly perturbs a source image to alter its prediction label (top to center). Purification (bottom) seeks to correct adversarial examples to the right class without affecting clean samples. **(b)**: A Multiple Latent Variable Generative Model (MLVGM) maps latent variables (or codes, here $\mathbf{z}_0, \mathbf{z}_1, \mathbf{z}_2$) sampled from a known prior distribution, to high-quality images. **(c)**: Each code impacts the image differently, from global to local features. From left to right: original image and those generated by replacing variables $\mathbf{z}_0$ with $\mathbf{z}'_0$, $\mathbf{z}_1$ with $\mathbf{z}'_1$, and $\mathbf{z}_2$ with $\mathbf{z}'_2$, respectively.

Abstract

Attackers can deliberately perturb classifiers' input with subtle noise, altering final predictions. Among proposed countermeasures, adversarial purification employs generative networks to preprocess input images, filtering out adversarial noise. In this study, we propose specific generators, defined Multiple Latent Variable Generative Models (MLVGMs), for adversarial purification. These models possess multiple latent variables that naturally disentangle coarse from fine features. Taking advantage of these properties, we autoencode images to maintain class-relevant information, while discarding and re-sampling any detail, including adversarial noise. The procedure is completely training-free*, exploring the generalization abilities of pre-trained MLVGMs on the adversarial purification downstream task. Despite the lack of large models, trained on billions of samples, we show that smaller MLVGMs are already competitive with traditional methods, and can be used as foundation models. Official code released at https://github.com/SerezD/gen_adversarial.*

1. Introduction

It is well known that subtle alterations of input data can significantly impact the predictions of properly trained target models, such as image classifiers. This was studied in the pioneering work of Szegedy et al. [63] and in follow-up works like [9, 20, 50]. An attacker can thus intentionally design an adversarial example with this objective (Figure 1 (a), top to center), threatening the deployment of deep learning in real-world applications, like smartphone's captured images classification [17] or street sign recognition [49], to cite a few. These vulnerabilities led to an ongoing race between increasingly sophisticated attacks and possible defenses [1, 13, 42, 45]. In fact, researchers proposed different solutions to overcome these shortcomings, like

increasing classifier's regularization by training with adversarial images (adversarial training) [57, 63, 66, 75], or applying adversarial purification algorithms [27, 44, 55, 61]. Purification methods (Figure 1 (a), bottom) act as filters between input images and classifiers, removing adversarial noise. Early approaches (*e.g.* [61]) demonstrated that adversarial noise moves the target image to low-confidence regions of the classifier's learned manifold. Therefore, they employed a pre-trained generative model [68] as a noise-removal, shifting adversaries back to high-confidence areas. Recently, pre-trained diffusion models [23] have been proposed for the same task [48, 76], removing noise directly in the pixel space. However, it was shown that specific counter-attacks can prevent their effectiveness [29].
Alternatively, purification can occur by projecting images onto some intermediate latent space [28, 72]. These approaches leverage the regularization properties of VAEs to ensure that close points on the low-dimensional manifold produce similar high-quality images, preserving the coarse semantic content (which determines the category label), while altering local details, including adversarial noise. However, these autoencoder-purification methods are usually specifically trained for the task, significantly increasing the computational overhead.

In this study, we propose a novel autoencoder-purification method, that effectively exploits the power of latent regularizations, but *does not require specific training*. Instead, we define and leverage an emerging class of pre-trained generators, namely, Multiple Latent Variable Generative Models (MLVGMs). Differently from standard latent variable generators like VAEs [35, 54] and GANs [19], these models use multiple latent variables (also called *latents* or *codes*) to inject progressive noise during decoding, generating richer and more detailed samples (Figure 1 (b)). Notable examples include StyleGANs [31–33, 56], GigaGAN [30], and NVAE [67]. As observed in these works, the multiple variables enhance control over the generative process, naturally disentangling global/coarse and local/fine features (see Figure 1 (c)). So far, these natural disentangling properties have been exploited in various image editing applications, particularly by coupling StyleGAN networks with appropriate encoders [25, 34, 52, 65, 69, 79]. In this work instead, we take advantage of these properties to perform purification of adversarial images.
From an MLVGMs perspective, we explore their potential as powerful foundation models, exploiting their abilities in image generation on the zero-shot task of adversarial purification. Foundation models, such as CLIP [53] or BERT [16], are nowadays seen as an easy-to-deploy solution to various downstream tasks, thanks to their generalization capabilities [5]. Unfortunately, an open-source MLVGM, trained on huge amounts of data and thus serving as a proper foundation model, is not available yet. Interestingly, we find that smaller models, such as StyleGan2 [33] or NVAE [67], are already powerful enough to be used as purification methods, with no further fine-tuning required. We hope that our findings will set the base for more powerful MLVGMs, trained on billions of samples and deployable on further and diverse downstream tasks.

The motivation for our approach arises from the fact that adversarial images are designed to closely resemble the reference, "clean" sample, enforcing an L_p norm bound (usually $p \in \{0, 2, \text{or } \infty\}$). In the pixel space, this imperceptible noise overlaps with class-relevant information, moving the sample to a low-probability region of the targeted classifier's learned manifold [61]. Conversely, when encoded in the multiple latent manifolds of MLVGMs, these two types of information likely reside on different levels, since class-relevant information usually has a global impact on the image content, while adversarial information is enforced to change imperceptible, local details. Given these observations, we do not limit to remove adversarial noise, as in diffusion-based methods [48, 76]. Instead, we aim to preserve the class-relevant features (essential to determine the final label), while discarding *any* remaining information, including adversarial noise. Since the discarded information does not alter the class label, it can be re-sampled using the generative part of the model, producing clean details.

Our algorithm can be described in three main steps: encoding, sampling, and interpolation. Given an unknown image (adversarial or not), we first *encode* it to obtain N latent variables $\mathbf{z}^{\text{e}}_0, \mathbf{z}^{\text{e}}_1, \dots, \mathbf{z}^{\text{e}}_{N-1}$. These contain relevant class information, which we want to keep, and possible adversarial information, which we need to discard. Second, we *sample* N new latent variables $\mathbf{z}^{\text{s}}_0, \mathbf{z}^{\text{s}}_1, \dots, \mathbf{z}^{\text{s}}_{N-1}$ from the pre-trained generator's prior distribution, as if we would like to synthesize a novel image. Since the MLVGM has learned to represent a clean data distribution during training, these codes do not contain adversarial noise. However, the specific class-relevant information of the codes is unknown. In the third step, we linearly *interpolate* each $\mathbf{z}^{\text{e}}_i$ and $\mathbf{z}^{\text{s}}_i$ according to a certain $0 \leq \alpha_i \leq 1$. This is the core of the purification algorithm: we would like to assign low α_i to latent levels representing class-relevant information, thus maintaining the original $\mathbf{z}^{\text{e}}_i$, and high α_i to the latent levels representing irrelevant details, thus practically using the new clean code $\mathbf{z}^{\text{s}}_i$. Finally, we use the generative model to decode the interpolated codes, obtaining the purified image.

With this framework, we leverage the power of VAE-based purification methods, but employ pre-trained MLVGMs, resulting in a training-free procedure and removing the additional overhead. The only adjustable

parameters are $\alpha_0, \alpha_1, \ldots, \alpha_{N-1}$ which, given a specific problem (target-model and dataset), can be estimated via optimization algorithms, such as Bayesian Optimization (BO) [18, 59]. However, since the global, class-relevant information (which we need to maintain) is often contained in the first codes (see Figure 1 (c)), we show that it is possible to obtain good values of hyperparameters also with reasonable heuristics, by maintaining encoder-information in the first codes while changing the later ones.

We test our framework employing two different MLVGMs (StyleGan2 [33], coupled with appropriate encoders [25, 65], and NVAE [67]) on three different scenarios: binary classification (male, female) and fine-grained identity classification (100 classes) on the Celeb-A dataset [41], and Cars type classification (4 classes) on a subset of the Stanford Cars dataset [36]. In each scenario, we apply state-of-the-art attacks (DeepFool [46] and Carlini&Wagner [9]) to compare performances of our model with the base (undefended) classifier, adversarial learning [75], and similar generative autoencoding purification methods, namely A-VAE [78] and ND-VAE [28]. The experimental results show that, despite the used MLVGMs are not originally trained on very large amounts of data, as proper foundation models, they can already compete at the same level of specifically designed techniques, while being completely training-free.

To summarize, our contributions are: 1) We propose a novel autoencoding-based purification framework, which employs an off-the-shelf pre-trained architecture, namely, Multiple Latent Variable Generative Models (MLVGMs), requiring at most the estimation of a handful of hyperparameters; 2) We show the potential of MLVGMs as strong foundation models for adversarial purification: thanks to the regularization and disentangling properties of the multiple latent spaces, these models are proved to be effective for downstream tasks without the need to be specifically trained. 3) Our experiments encompass different image domains and MLVGMs, setting them as a valid alternative to other purification methods, regardless of the specific training procedure (VAE or GAN) and despite not being pre-trained on billions of samples, as proper foundation models.

2. Background and Related Works

In this section, we provide the minimal background on adversarial attacks, before delving into the literature on adversarial defenses and MLVGMs.

Adversarial attacks. As noted in [63], subtle changes to a classifier's input can lead to incorrect predictions. These often imperceptible perturbations do not affect the input's semantic content, revealing vulnerabilities in the model's robustness and security. Formally, these adversarial perturbations are defined as:

$$\arg\min_{\delta} \|\delta\| \text{ s.t. } \mathbb{I}(f(\mathbf{x}+\delta) \neq \mathbf{y}), \tag{1}$$

where $f(\mathbf{x}+\delta)$ is the prediction of the model $f(\cdot)$ for the input $\mathbf{x}$ with perturbation δ, $\mathbf{y}$ is the ground truth label, and $\mathbb{I}(\cdot)$ is the indicator function. Opposed to the untargeted case of Equation (1), an attack is targeted when $f(\mathbf{x}+\delta)$ is forced to output a specific incorrect label $\mathbf{y}' \neq \mathbf{y}$. To measure the perturbation and evaluate attack effectiveness, L_p norm is used, where $p \in \{0, 2, \infty\}$. An attack is considered successful within the bound ϵ if the minimal perturbation found satisfies $\|\delta\|_p \leq \epsilon$.

Based on the attacker's knowledge of the target model, adversarial attacks can be categorized into *blackbox* and *whitebox*. The former class [6, 10] relies only on the model's final predictions, while the latter [7, 20, 43] assumes full access to the model. For a complete overview of the numerous existing attacks, we refer to recent surveys like [1, 13, 42, 45]. In this work we test against *whitebox* and untargeted attacks, representing the most challenging setup for a defense mechanism. We select two different attacks, DeepFool [46] and Carlini-Wagner (C&W) [9], as representative state-of-the-art methods employing different techniques to find the best perturbation. DeepFool iteratively perturbs the input image to find the minimal change required to alter the classification result. At each step, it linearly approximates the decision boundary and projects the image toward it until misclassification occurs. Conversely, C&W formulates the attack as an optimization problem, utilizing gradient information to minimize the perturbation while maximizing classifier's loss.

Defending against adversarial attacks. The first proposed approach to reduce the effectiveness of attacks is known as adversarial training [20, 57, 63, 66], which involves generating adversarial examples during the training phase to enhance the classifier's robustness. Although scalable to large datasets [37], this method has drawbacks, including a significant computational cost and the risk of overfitting to specific attacks. Noteworthy is also the area of certified robustness [12, 22, 38, 64, 70] providing theoretical guarantees that any perturbation within a certain L_p norm ball is classified correctly. While promising, these methods often give guarantees only to a specific p norm, or excessively reduce the classifier's accuracy on clean data.

Our work belongs instead to the well established concept of adversarial purification, which involves removing adversarial noise from input images. Seminal works such as [44, 61] postulate that adversarial samples reside in

low-probability regions of target classifier's manifold, reducing confidence. Thus, the objective of purification is to move adversarial samples back towards high-probability regions, removing the applied noise. This approach acts as a preprocessing filter of input images, maintaining the classifier unaltered and allowing the combination with other techniques. Subsequent studies leveraged the regularization properties of latent variable generative models, including GANs [19] and VAEs [35, 54]. Specifically, [55] proposed identifying the latent space vector $\mathbf{z}$ of a pre-trained GAN $g_{\boldsymbol{\theta}}$ that minimizes the distance between the real input $\mathbf{x}$ and the generated $g_{\boldsymbol{\theta}}(\mathbf{z})$, which serves as the purified image. Other methods like [27, 28, 39] train a VAE to reconstruct clean images from adversarial examples. Further VAE-based approaches like [73, 78] hypothesize that adversarial attacks primarily manipulate local information. Therefore, they aim at preserving coarse features while purifying the details. Our method follows this principle, but we leverage the powerful representations of *pre-trained* models, arguing that the training of specific autoencoders is not necessary. In general, the use of pre-trained generators has been proposed also in the context of diffusion-based purification [48, 76]. However, these approaches act directly in the pixel-space, and can be countered by specific attacks [29]. Instead, we leverage the regularization and disentanglement properties of MLVGMs latent spaces to discard and re-sample any information that is not relevant for the final class label, increasing the attacker's challenge.

Similarly to other pipelines, our method includes random sampling operations. As seen in [2], randomized defenses can give a false sense of robustness, by masking true gradients to the attacker. Gradient obfuscation can be easily circumvented by averaging the gradients over multiple input transformations, taking the Expectation over Transformation (EoT) [3]. In our experimental protocol, we consider this aspect and appropriately use EoT to avoid gradient masking.

Multiple Latent Variable Generative Models. Current literature encompasses a large collection of generative models employing multiple random variables, especially those based on GANs [8, 15, 30] and VAEs [11, 40, 60, 67, 77]. Notably, these have been previously leveraged in image editing tasks [25, 52, 65], particularly employing the StyleGan family of networks [31–33, 56]. Image editing tasks are feasible only when the StyleGan is coupled with an appropriate encoder network, allowing for the extraction of latent codes from real images [34, 69, 79], a task known as GAN inversion. In this work, we also couple StyleGan with appropriate encoder networks, but we use the disentanglement properties of coarse from finer information for the task of adversarial purification. By doing so, we show that MLVGMs have the potential to be used as foundation models, despite generators trained on billions of samples, like GigaGAN [30] are not fully available yet. Interestingly, we observe that the field of MLVGMs is expanding, with promising approaches based on Normalizing-Flow [24], and SODA [26], a diffusion model coupled with a multi-latent encoder.

3. Methodology

Purification Framework with MLVGMs. The proposed framework's overview is shown in Figure 2, while we provide the pseudocode in Appendix A. Initially, the input image $\mathbf{x}$ undergoes an *optional* preprocessing stage, consisting of noise enhancement (adding Gaussian noise) or suppression (applying Gaussian blur). Similarly to previous methods like [28, 78], we experimentally observe that this step can sometimes increase the challenge for the attacker. Even tough our method works even with no preprocessing on the input images, these optional operations demonstrate that it can be successfully coupled with other pipelines to increase overall performance, while maintaining its stability.

After the initial preprocessing, we leverage the pre-trained generative autoencoder for the core purification procedure (big rectangle in Figure 2). First, we encode the input image to obtain multiple latent variables $\mathbf{z}_0^{\text{e}}, \mathbf{z}_1^{\text{e}}, \dots, \mathbf{z}_{N-1}^{\text{e}}$ (Figure 2, ①). These codes represent different levels of information, from coarse to fine, capturing both class-relevant features (to be maintained) and irrelevant details/adversarial noise (to be suppressed). Next, we sample N new latent codes $\mathbf{z}_0^{\text{s}}, \mathbf{z}_1^{\text{s}}, \dots, \mathbf{z}_{N-1}^{\text{s}}$ from each latent prior distribution, utilizing the model's generative properties (Figure 2, ②). These latent variables form a novel image with an unknown final classification label. However, since the generative model is trained to represent the real data distribution accurately, the sampled codes do not contain adversarial information.

Given the two types of latent variables ($\mathbf{z}_i^{\text{e}}$ and $\mathbf{z}_i^{\text{s}}$), we aim to obtain a new set of codes, retaining the class-relevant features of $\mathbf{z}_i^{\text{e}}$ and the clean details of $\mathbf{z}_i^{\text{s}}$. Therefore, we linearly interpolate each $\mathbf{z}_i^{\text{e}}$ and $\mathbf{z}_i^{\text{s}}$ (Figure 2, ③):

$$\mathbf{z}_i = (1 - \alpha_i)\mathbf{z}_i^{\text{e}} + \alpha_i \mathbf{z}_i^{\text{s}}, \tag{2}$$

where $0 \leq \alpha_i \leq 1$, with $\alpha_i = 0$ using only the encoding information and $\alpha_i = 1$ using only the new information. The outcome of this interpolation process is N purified codes $\mathbf{z}_0, \mathbf{z}_1, \dots, \mathbf{z}_{N-1}$, which, once decoded, produce a purified version of the input image, becoming the classifier's input (rightmost part of Figure 2). The crucial aspect of this process, discussed next, resides in the selection of the optimal $\alpha_0, \alpha_1, \dots, \alpha_{N-1}$ to maintain the input's class-relevant information while effectively removing anything else. As seen

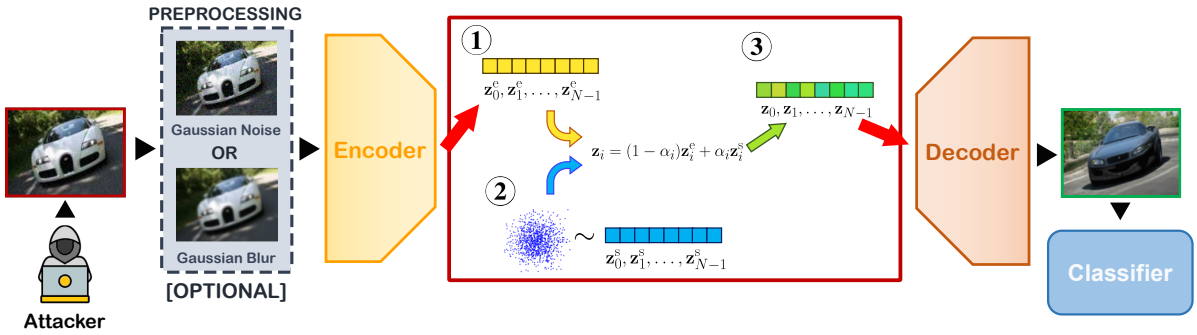


Figure 2. The overall architecture of our framework, depicting the *optional* preprocessing phase (adding gaussian noise or blurring the input image) and autoencoding with pre-trained MLVGMs, which *do not require further training*. Inside the central rectangle, we show the latent purification process that is the core of our method, consisting of three main steps: (1) encoding, (2) sampling and (3) interpolation.

in the purified image in Figure 2, we do more than just remove adversarial noise. The goal is to find the parameters that alter *any* features irrelevant to class labels, such as the car's color or background. By maximizing the discarded and resampled information, we restrict the attacker's degrees of freedom and effectiveness. For instance, if only $\mathbf{z}_0^e$ is retained after purification, to affect the final classifier the attacker is *forced* to alter parts of the input that are encoded at that level. However, since in MLVGMs the first codes typically influence global parts of the image, this constraint struggles with the enforced L_p bound on adversarial noise, which affects only imperceptible details. This double constraint (one in the latent space, one in the pixel space), poses a great challenge for the attacker, limiting the overall effectiveness of the perturbation.

Selection of α hyperparameters. We now discuss possible strategies to find the optimal choice of the α parameters, which should maintain class-relevant information from $\mathbf{z}^e$ codes and introduce novel details thanks to $\mathbf{z}^s$. Specifically, we propose using Bayesian Optimization [18, 59], an effective algorithm for finding the best hyperparameters in machine learning when the search space is small and continuous. For α selection, each combination has N dimensions, corresponding to the number of latent levels (up to 24 in our experiments), with $\alpha \in \mathbb{R}^N : 0 \leq \alpha_i \leq 1$. Bayesian Optimization is also suitable when evaluating the *objective function* is complex, making simpler algorithms like grid search impractical. In our setup, the *objective function* evaluates purification performance for a given set of hyperparameters. Specifically, we define a "base model" as the framework described in Figure 2, without preprocessing, and with $\alpha_i = 0, \forall i$, corresponding to simple autoencoding + classification. Then, we attack this model by applying FGSM [20] to every image in the source dataset, obtaining its adversarial version. We define the *objective function* of Bayesian Optimization as the accuracy computation of the base model on the adversarial dataset. At each step, we change the set of hyperparameters to match the one suggested by the optimization algorithm.

The use of optimization algorithms, however, presents some drawbacks: the requirement of specific adversarial images to compute the *objective function* and the overall computational cost. In fact, evaluating the *objective function* may still be infeasible in resource-constrained scenarios with very limited resources. In other terms, it would be desirable to have a complete optimization-free purification, using pre-trained MLVGMs without hyperparameters tuning. In the following, we propose such an alternative, using the properties of MLVGMs to define what characteristics a good combination should have.

As discussed in Section 1, MLVGMs encode information at various granularities, from global to local details (Figure 1 (**c**)), a behavior observed in studies like [8, 32, 67]. In most classification problems, global aspects are more relevant for label definition than pixel-level details. For example, in identity classification, general face traits are more crucial than localized features, which can change without affecting the label. Based on this insight, we hypothesize that a good set of α parameters should retain encoding information from initial latent levels while replacing the later ones with the sampled codes. Thus, when optimization is infeasible, we propose selecting monotonic sets of values, where $\alpha_i > \alpha_j$ for all $i > j$. Specifically, we experiment with two methods:

$$\textbf{linear} \quad \alpha_i = \frac{(i+1)}{N} \qquad \forall i \in \{0, 1, \ldots, N-1\}; \quad (3)$$

$$\textbf{cosine} \quad \alpha_i = \frac{1 - \cos\frac{\pi(i+1)}{N}}{2} \qquad \forall i \in \{0, 1, \ldots, N-1\}; \quad (4)$$

finding them to be stable across various scenarios, and competitive with the combination found after optimization.

4. Experiments

Pre-trained MLVGMs, datasets and classifiers. Unfortunately, no proper foundation MLVGM (trained on

billions of images) is available yet. GigaGAN [30] shows impressive results, but no pre-trained model has been released. Therefore, we experiment with three types of smaller MLVGMs: a StyleGan2 [33] model pre-trained on the Celeb-A HQ dataset [41], coupled with the E4E Encoder proposed by [65]; an NVAE model [67], also pre-trained on Celeb-A; and a StyleGan2 model pre-trained on LSUN Cars [74], coupled with the Style Transformer Encoder (STE) proposed by [25]. To further enhance the broadness of our experiments, each model is tested for its purification capability on a dedicated classification problem, considering the inherent drawbacks of each.

The E4E-StyleGan2 model suffers from imperfect reconstructions, due to the post hoc training of the encoder. Therefore, we choose a binary classification task (male/female), ensuring that the relevant global aspects (gender) are maintained after reconstruction. Specifically, we employ the gender classification dataset introduced in [47], containing 24k training and 6k validation images, at a resolution of 256×256. Next, we train an NVAE to demonstrate our framework's compatibility also with VAE-based architectures and to tackle a more challenging classification task, possible thanks to precise reconstructions. We used a subset of the Celeba-Identities [41] dataset, comprising 2600 training and 600 validation images, on 100 identity classes and a 64×64 resolution. Lastly, for the STE-StyleGAN2 model we select cars classification, to show applicability across different image domains. For the same reasons as E4E-StyleGan2, also STE-StyleGAN2 suffers from imperfect reconstructions. Therefore, we use a subset of the Stanford Cars dataset [36], grouping the provided label names into four classes: Coupe, Hatchback, SUV and Minivan. The dataset contains 800 128×128-resolution images per class, keeping 100 of them for validation.

As classifiers, we trained a Resnet-50 [21] for 50 epochs, a Vgg-11 [58] for 200 epochs and a Resnext-50 [71] model for 150 epochs, respectively. All use SGD optimizer and a learning rate of $1e-3$. We provide more details and hyperparameters in Appendix C.

Threat model and baselines. As anticipated in Section 2, we evaluate each purification model under the challenging scenario of *whitebox* untargeted attacks, constrained by an L_2 norm. This means that the attacker has complete access to the entire model, including the purification framework, and the attack is deemed successful if the target classifier predicts *any* incorrect class. Specifically, we test our model against two prominent adversarial attacks: DeepFool [46] and Carlini & Wagner (C&W) [9][1]. To obviate the problem of gradient obfuscation, we couple each attack with Expectation over Transformation (EoT) [3]. In all cases, we average gradients over 32 forward passes, or the maximum amount allowed by the available resources.

[1]Further experiments on Autoattack [14] are provided in Appendix F.

Method	Optimized Params.
Trades [75]	10^7
A-VAE [78]	10^7
ND-VAE [28]	10^6
ours w/ BO	10^1
ours w/o BO	0

Table 1. Average (on each task) order of magnitude of parameters that need to be optimized for each compared method.

Given an attack, we seek for each sample the minimal perturbation needed to cause a misclassification, obtaining an adversarial set of images $\{\hat{\mathbf{x}}_i^{\delta_i}\}_{i=1}^N$ where N is the size of the test set, δ_i is the minimum perturbation found for sample $\mathbf{x}_i$ and $\hat{\mathbf{x}}_i^{\delta_i} = \mathbf{x}_i + \delta_i$. Then, we compute the average success rate (SR) at a certain L_2 bound as:

$$\text{SR}_{L_2=\epsilon} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}((\hat{y}_i \neq y_i) \wedge (\delta_i \leq \epsilon)), \tag{5}$$

where $\hat{y}_i$ is the predicted label of the target model given the sample $\hat{x}_i^{\delta_i}$, y is the ground truth label, ϵ the considered L_2 bound and $\mathbb{I}(\cdot)$ the indicator function. In each experiment, we report the success rate obtained on $N = 100$ images uniformly sampled from the available classes. Further details and a comprehensive list of hyperparameters used are included in Appendix C.

For each scenario, we first measure the success rate of the different hyperparameter choices (**learned**, **linear**, and **cosine**). Then, we ablate on the effects of adding a preprocessing operation (see Appendix D). Lastly, we benchmark the best-performing configuration (hyperparameters and preprocessing) against: the classification model alone (no defense), the classification model regularized with adversarial training, using TRADES [75], and similar adversarial-purification methods, A-VAE [78] and ND-VAE [28]. Table 1 shows the cost of each method, in terms of the number of parameters to optimize. For TRADES, we fine-tuned the classifier for 50 epochs, while for ND-VAE and A-VAE we trained the additional Generative-Autoencoder from scratch, on each dataset (further details in Appendix C). Conversely, our method requires optimizing just a few parameters in the **learned** case, and none if **linear** or **cosine** configurations are used.

Effects of α configurations. To offer a broader perspective on the effectiveness of choosing a monotonic set of hyperparameters, we propose to measure it statistically. Specifically, we use the "base model" detailed in Section 3 to measure

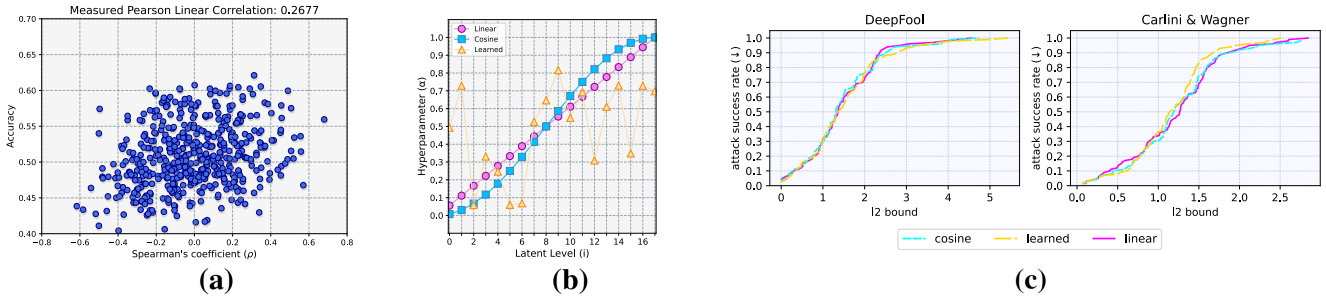


Figure 3. Analyisis of α combinations on the Celeb-A HQ Gender task. **(a)** Spearman's index (ρ) vs accuracy for 512 random combinations, obtaining a Pearson's linear correlation value of 0.267. **(b)** Comparison of the 18 final α values for the **linear**, **cosine** and **learned** combinations. **(c)** Attack success rates (the lower the better) for increasing L_2 bounds on each tested attack and combination.

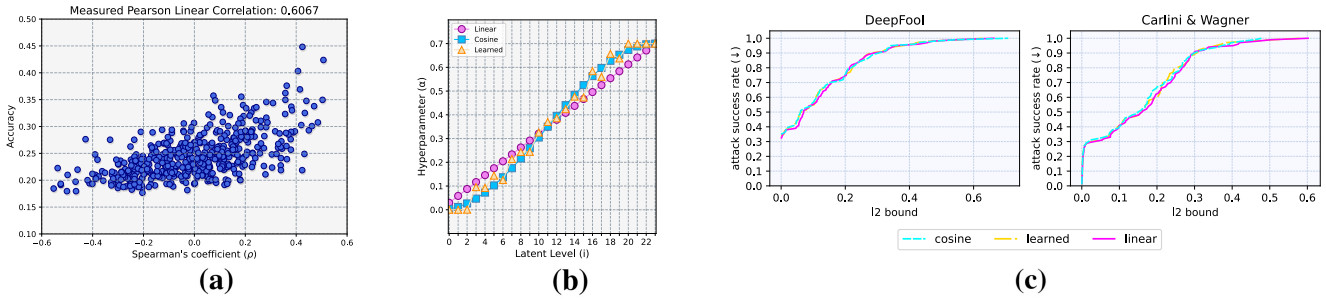


Figure 4. Analyisis of α combinations on the Celeb-A 64 Identities task. **(a)** Spearman's index (ρ) vs accuracy for 512 random combinations, obtaining a Pearson's linear correlation value of 0.607. **(b)** Comparison of the 24 final α values for the **linear**, **cosine** and **learned** combinations. **(c)** Attack success rates (the lower the better) for increasing L_2 bounds on each tested attack and combination.

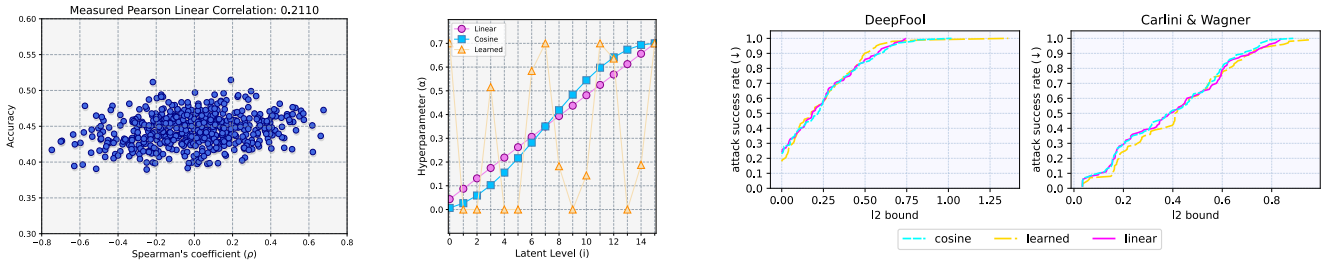
the accuracy of 512 random combinations of α values, sampled uniformly. For each combination, we also calculate the Spearman's rank correlation index [62], to gauge its monotonicity degree:

$$\rho = 1 - \frac{\sum_{i=0}^{N-1}(i - R_{\alpha_i})^2}{N(N^2-1)}; \tag{6}$$

where N is the number of latent levels and R_{α_i} is the rank of the parameter α_i, ranging from 0 to $N-1$. We plot the two variables (Spearman's rank vs Accuracy) for each task in Figures 3 to 5 (a), and measure their Pearson linear correlation coefficient [51]. We obtain values of $0.267, 0.607, 0.211$ respectively, where a value close to one indicates that a monotonic set of hyperparameters is particularly effective. In all cases, linear correlation exhibits a positive value, meaning that a relation between monotonicity and accuracy exists. This is particularly strong in the fine-grained problem of ids classification. We hypothesise that in this case relevant class-features are spread across various latents, in a coarse to fine manner. Thus, a gradually increasing set of α values is particularly effective. On the other hand, in coarse-grained classification, relevant features are mainly concentrated in a few, initial latents, implying that a monotonic set of values performs well, but is not essential. Therefore, learning the best combination may give better results in such cases.

To learn the best combination with Bayesian Optimization, we use the BoTorch library [4] and fit a Gaussian Process model for 95 steps, using Expected Improvement as the acquisition function. Five additional combinations are given for initialization: **linear** (Eq. (3)), **cosine** (Eq. (4)), uniform ($\alpha_i = 0.5, \forall i \in \{0, 1, \dots, N-1\}$), $1-$ linear, $1-$ cosine. In Figures 3 to 5 (b) we visually compare the **learned** combination with the fixed ones. In ids and cars tasks, we force the maximum α value to 0.7, since higher values caused an high degradation of clean accuracy on **linear** and **cosine** cases. Aligning with previous observations, the best results for ids-classification are obtained with a monotonic set of values, while in other cases BO highly penalizes some latent levels, maintaining information unaltered in other ones. We qualitatively analyze these **learned** combinations in Appendix B. Figures 3 to 5 (c) shows the purification abilities on the DeepFool and C&W attacks. The combination learned on FGSM for gender classification works well on DeepFool, but no on C&W. On ids-classification, all combinations unsurprisingly perform similarly. For cars-classification, the **learned** combination allows an extra performance boost on both attacks.

Comparison with other methods. Figures 6 to 8 show the comparison between the best configuration of our framework



(a) **(b)** **(c)**

Figure 5. Analyisis of α combinations on the Stanford Cars 128 task. **(a)** Spearman's index (ρ) vs accuracy for 512 random combinations, obtaining a Pearson's linear correlation value of 0.211. **(b)** Comparison of the 16 final α values for the **linear**, **cosine** and **learned** combinations. **(c)** Attack success rates (the lower the better) for increasing L_2 bounds on each tested attack and combination.

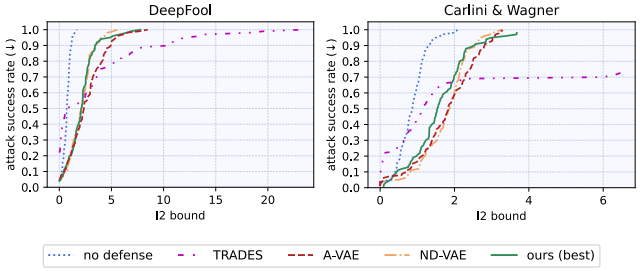


Figure 6. Attack success rates (lower is better) for increasing L_2 bounds. Comparison of different defenses on Celeb-A HQ Gender.

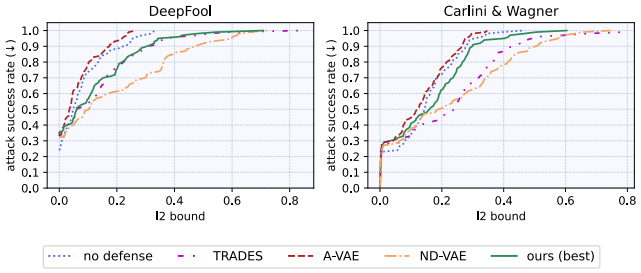


Figure 7. Attack success rates (lower is better) for increasing L_2 bounds. Comparison of different defenses on Celeb-A 64 Identities.

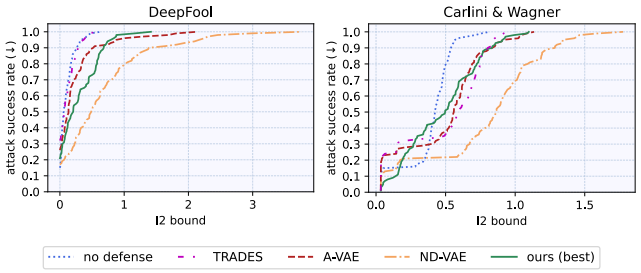


Figure 8. Attack success rates (lower is better) for increasing L_2 bounds. Comparison of different defenses on Stanford Cars 128.

and similar defense methods. In Appendix D we ablate on pre-processing operations, finding that gaussian blur is beneficial on gender classification, while gaussian noise performs best on the other tasks. We refer to Appendix E for some qualitative purification examples. Looking at the figures, TRADES shows very strong results on gender classification, outlining the power of Adversarial Training in 2-classes problems; while all purification methods (including ours) perform similarly. In the more challenging scenarios of ids and cars classification, conversely, ND-VAE generally performs best, with our method always showing competitive results. More broadly, MLVGMs prove to be good adversarial purifiers, despite not being specifically trained for the task and the relatively small size of the used models. We hypothesise that this is due to the coarse-to-fine disentenglement of their features, which offers promising reasearch directions for the development of more powerful MLVGMs, acting as proper foundation models.

5. Discussion and Conclusions

In this paper we proposed a novel adversarial purification method, using Multiple Latent Variable Generative Models (MLVGMs) as foundation models. Our approach takes advantage of their latent disentanglement properties, leveraging them on the adversarial purification downstream task in a training-free manner. In the worst case, only a few hyperparameters need to be optimized via Bayesian Optimization. However, thanks to the global-to-local features of MLVGMs, good values can be defined a priori.

While promising, the use of MLVGMs as foundation models still poses some challenges. Specifically, the lack of strong open-source models, trained on billions of samples, is a limiting factor in obtaining stronger results. In our study, we employed StyleGan2 [33] and NVAE [67] networks, which are smaller models that present some intrinsic limitations. Despite this, the proposed framework already shows competitive results, when compared to similar defense models that require specific training. Therefore, our study highlights the significant potential of MLVGMs as strong foundation models, encouraging research to release more powerful generators.

References

[1] Naveed Akhtar and Ajmal Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access*, 2018. 1, 3

[2] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning*, pages 274–283. PMLR, 2018. 4

[3] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *International Conference on Machine Learning*, pages 284–293. PMLR, 2018. 4, 6

[4] Maximilian Balandat, Brian Karrer, Daniel R. Jiang, Samuel Daulton, Benjamin Letham, Andrew Gordon Wilson, and Eytan Bakshy. Botorch: A framework for efficient monte-carlo bayesian optimization. In *Advances in Neural Information Processisng Systems*, 2020. 7

[5] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint*, 2021. 2

[6] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *International Conference on Learning Representations*, 2018. 3

[7] Wieland Brendel, Jonas Rauber, Matthias Kümmerer, Ivan Ustyuzhaninov, and Matthias Bethge. Accurate, reliable and fast robustness evaluation. *Advances in Neural Information Processisng Systems*, 32, 2019. 3

[8] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2018. 4, 5

[9] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, pages 39–57. IEEE, 2017. 1, 3, 6

[10] Jianbo Chen, Michael I. Jordan, and Martin J. Wainwright. Hopskipjumpattack: A query-efficient decision-based attack. In *IEEE Symposium on Security and Privacy*, pages 1277–1294. IEEE, 2020. 3

[11] Rewon Child. Very deep vaes generalize autoregressive models and can outperform them on images. In *International Conference on Learning Representations*, 2020. 4

[12] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pages 1310–1320. PMLR, 2019. 3

[13] Joana C Costa, Tiago Roxo, Hugo Proença, and Pedro RM Inácio. How deep learning sees the world: A survey on adversarial attacks & defenses. *IEEE Access*, 2024. 1, 3

[14] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International Conference on Machine Learning*, pages 2206–2216. PMLR, 2020. 6, 1, 4, 7

[15] Emily L Denton, Soumith Chintala, Rob Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. *Advances in Neural Information Processisng Systems*, 28, 2015. 4

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*, 2019. 2

[17] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1625–1634, 2018. 1

[18] Peter I Frazier. A tutorial on bayesian optimization. *arXiv preprint*, 2018. 3, 5

[19] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in Neural Information Processisng Systems*, 27, 2014. 2, 4

[20] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. 1, 3, 5

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 6

[22] Matthias Hein and Maksym Andriushchenko. Formal guarantees on the robustness of a classifier against adversarial manipulation. *Advances in Neural Information Processisng Systems*, 30, 2017. 3

[23] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processisng Systems*, 33, 2020. 2

[24] Hong-Ye Hu, Dian Wu, Yi-Zhuang You, Bruno Olshausen, and Yubei Chen. Rg-flow: A hierarchical and explainable flow model based on renormalization group and sparse prior. *Machine Learning: Science and Technology*, page 035009, 2022. 4

[25] Xueqi Hu, Qiusheng Huang, Zhengyi Shi, Siyuan Li, Changxin Gao, Li Sun, and Qingli Li. Style transformer for image inversion and editing. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 11337–11346, 2022. 2, 3, 4, 6

[26] Drew A Hudson, Daniel Zoran, Mateusz Malinowski, Andrew K Lampinen, Andrew Jaegle, James L McClelland, Loic Matthey, Felix Hill, and Alexander Lerchner. Soda: Bottleneck diffusion models for representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 23115–23127, 2024. 4

[27] Uiwon Hwang, Jaewoo Park, Hyemi Jang, Sungroh Yoon, and Nam Ik Cho. Puvae: A variational autoencoder to purify adversarial examples. *IEEE Access*, 2019. 2, 4

[28] Shayan Jalalinour and Banafsheh Rekabdar. Noisy-defense variational auto-encoder (nd-vae): An adversarial defense framework to eliminate adversarial attacks. In *International Conference on Transdisciplinary AI*, pages 50–57. IEEE, 2023. 2, 3, 4, 6, 1

[29] Mintong Kang, Dawn Song, and Bo Li. Diffattack: Evasion attacks against diffusion-based adversarial purification. *Advances in Neural Information Processisng Systems*, 36, 2024. 2, 4

[30] Minguk Kang, Jun-Yan Zhu, Richard Zhang, Jaesik Park, Eli Shechtman, Sylvain Paris, and Taesung Park. Scaling up gans for text-to-image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 10124–10134, 2023. 2, 4, 6

[31] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *Advances in Neural Information Processisng Systems*, 34, 2021. 2, 4

[32] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 2, 4, 5

[33] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8107–8116, 2020. 2, 3, 4, 6, 8

[34] Kai Katsumata, Duc Minh Vo, Bei Liu, and Hideki Nakayama. Revisiting latent space of gan inversion for robust real image editing. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5313–5322, 2024. 2, 4

[35] Diederik P Kingma and Max Welling. Autoencoding variational bayes. In *International Conference on Learning Representations*, 2014. 2, 4

[36] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *International Conference on Computer Vision, Workshop*, 2013. 3, 6

[37] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In *International Conference on Learning Representations*, 2017. 3

[38] Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *IEEE Symposium on Security and Privacy*, pages 656–672. IEEE, 2019. 3

[39] Xiang Li and Shihao Ji. Defense-vae: A fast and accurate defense against adversarial attacks. In *International Workshops of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 191–207. Springer, 2020. 4

[40] Zhiyuan Li, Jaideep Vitthal Murkute, Prashnna Kumar Gyawali, and Linwei Wang. Progressive learning and disentanglement of hierarchical representations. In *International Conference on Learning Representations*, 2019. 4

[41] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *International Conference on Computer Vision*, 2015. 3, 6

[42] Teng Long, Qi Gao, Lili Xu, and Zhangbing Zhou. A survey on adversarial attacks in computer vision: Taxonomy, visualization and future directions. *Computers & Security*, page 102847, 2022. 1, 3

[43] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. 3

[44] Dongyu Meng and Hao Chen. Magnet: A two-pronged defense against adversarial examples. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, page 135–147. Association for Computing Machinery, 2017. 2, 3

[45] Charles Meyers, Tommy Löfstedt, and Erik Elmroth. Safety-critical computer vision: an empirical survey of adversarial evasion attacks and defenses on computer vision systems. *Artificial Intelligence Review*, 56:217–251, 2023. 1, 3

[46] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2574–2582, 2016. 3, 6

[47] Dongbin Na, Sangwoo Ji, and Jong Kim. Unrestricted black-box adversarial attack using gan with limited queries. In *European Conference on Computer Vision*, pages 467–482. Springer, 2022. 6

[48] Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Animashree Anandkumar. Diffusion models for adversarial purification. In *International Conference on Machine Learning*, pages 16805–16827. PMLR, 2022. 2, 4

[49] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the ACM on Asia Conference on Computer and Communications Security*, page 506–519. Association for Computing Machinery, 2017. 1

[50] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *European Symposium on Security and Privacy*, pages 372–387. IEEE, 2016. 1

[51] Karl Pearson. Vii. note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, 58(347-352), 1895. 7

[52] Hamza Pehlivan, Yusuf Dalva, and Aysegul Dundar. Styleres: Transforming the residuals for real image editing with stylegan. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1828–1837, 2023. 2, 4

[53] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 2

[54] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pages 1278–1286. PMLR, 2014. 2, 4

[55] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. In *International Conference on Learning Representations*, 2018. 2, 4

[56] Axel Sauer, Katja Schwarz, and Andreas Geiger. Stylegan-xl: Scaling stylegan to large diverse datasets. *ACM SIGGRAPH Conference Proceedings*, 2022. 2, 4

[57] Uri Shaham, Yutaro Yamada, and Sahand Negahban. Understanding adversarial training: Increasing local stability of supervised models through robust optimization. *Neurocomputing*, 307:195–204, 2018. 2, 3

[58] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint*, 2014. 6

[59] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. *Advances in Neural Information Processisng Systems*, 25, 2012. 3, 5

[60] Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder variational autoencoders. *Advances in Neural Information Processisng Systems*, 29, 2016. 4

[61] Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. In *International Conference on Learning Representations*, 2018. 2, 3

[62] C Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1), 1904. 7

[63] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014. 1, 2, 3

[64] Vincent Tjeng, Kai Y Xiao, and Russ Tedrake. Evaluating robustness of neural networks with mixed integer programming. In *International Conference on Learning Representations*, 2018. 3

[65] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *ACM Transactions on Graphics*, 40:1–14, 2021. 2, 3, 4, 6

[66] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. In *International Conference on Learning Representations*, 2018. 2, 3

[67] Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *Advances in Neural Information Processisng Systems*, 33, 2020. 2, 3, 4, 5, 6, 8

[68] Aäron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International Conference on Machine Learning*, pages 1747–1756. PMLR, 2016. 2

[69] Tengfei Wang, Yong Zhang, Yanbo Fan, Jue Wang, and Qifeng Chen. High-fidelity gan inversion for image attribute editing. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 11379–11388, 2022. 2, 4

[70] Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*, pages 5286–5295. PMLR, 2018. 3

[71] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1492–1500, 2017. 6

[72] Zhaoyuan Yang, Zhiwei Xu, Jing Zhang, Richard Hartley, and Peter Tu. Adversarial purification with the manifold hypothesis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 16379–16387, 2024. 2

[73] Sheng-lin Yin, Xing-lan Zhang, and Li-yu Zuo. Defending against adversarial attacks using spherical sampling-based variational auto-encoder. *Neurocomputing*, 478:1–10, 2022. 4

[74] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop, 2016. 6

[75] 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 *International Conference on Learning Representations*, 2019. 2, 3, 6, 4

[76] Jiawei Zhang, Zhongzhu Chen, Huan Zhang, Chaowei Xiao, and Bo Li. DiffSmooth: Certifiably robust learning via diffusion models and local smoothing. In *USENIX Security Symposium*, pages 4787–4804, 2023. 2, 4

[77] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Learning hierarchical features from deep generative models. In *International Conference on Machine Learning*, pages 4091–4099. PMLR, 2017. 4

[78] Jianli Zhou, Chao Liang, and Jun Chen. Manifold projection for adversarial defense on face recognition. In *European Conference on Computer Vision*, pages 288–305. Springer, 2020. 3, 4, 6, 1

[79] Jiapeng Zhu, Yujun Shen, Yinghao Xu, Deli Zhao, Qifeng Chen, and Bolei Zhou. In-domain gan inversion for faithful reconstruction and editability. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 2, 4