OMG-ATTACK: Self-Supervised On-Manifold Generation of Transferable Evasion Attacks

Ofir Bar Tal
Tel Aviv University
ofirbartal@mail.tau.ac.il

Adi Haviv
Tel Aviv University
adi.haviv@cs.tau.ac.il

Amit H. Bermano
Tel Aviv University
amit.bermano@gmail.com

Abstract

Evasion Attacks (EA) are used to test the robustness of trained neural networks by distorting input data to misguide the model into incorrect classifications. Creating these attacks is a challenging task, especially with the ever increasing complexity of models and datasets. In this work, we introduce a self-supervised, computationally economical method for generating adversarial examples, designed for the unseen black-box setting. Adapting techniques from representation learning, our method generates on-manifold EAs that are encouraged to resemble the data distribution. These attacks are comparable in effectiveness compared to the state-of-the-art when attacking the model trained on, but are significantly more effective when attacking unseen models, as the attacks are more related to the data rather than the model itself. Our experiments consistently demonstrate the method is effective across various models, unseen data categories, and even defended models, suggesting a significant role for on-manifold EAs when targeting unseen models.

1. Introduction

Evasion attacks are a fundamental tool for model robustness validation [13, 14, 54]. As neural networks become increasingly ubiquitous, analyzing trained models for weaknesses by finding EA becomes more and more crucial. Despite their value, EAs are still disregarded during typical model development. This is because EAs are challenging to generate, especially as models become larger [9, 72, 28] and datasets more elaborate [41, 66, 44].

In terms of computational load, some approaches use optimization to generate individual attacks, which is infeasible for large-scale training [6, 49]. Methods that produce EAs more efficiently are typically white-box based, where the attack generator leverages full knowledge and access to the target model, including its architecture, parameters, weights, and gradients [26, 47]. Again, for real-life scenarios training an adversarial counterpart for each target model is infeasible.

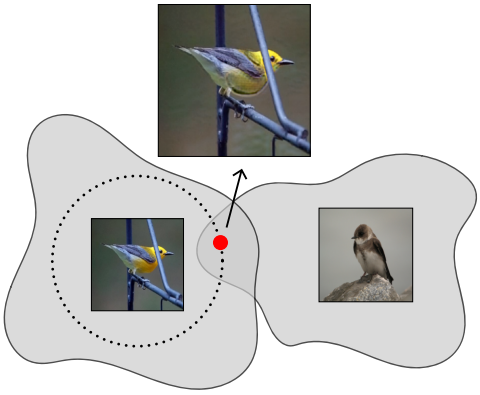


Figure 1. Visualization of the on-manifold adversarial perturbation method on the CUB-200 (Birds) dataset. The method generates examples on the general dataset manifold, but in an adversarial direction from the source image, projected on the L_1 ball, with a fixed radius *budget*.

Black-box attacks are more suitable in this aspect, where the attacker has no knowledge of the target model parameters [6, 62]. In their general variant, however, black-box attacks typically require access to the target model during inference.

Evasion attacks' transferability is a trait that enables the same attack to be effective against other unseen target models, making black-box attacks practical in real-world applications. Despite it's clear merit, transferability is challenging; optimization-based attacks typically perform poorly on transferability [6, 38]. Gradient-based methods, on the other hand, generate more transferable adversarial examples [26, 19], but they usually have low success rates making them less appealing to use in black-box scenarios [38].

In this paper, we present OMG-ATTACK, an EAs generation approach that addresses all aforementioned considerations. OMG-ATTACK is a self-supervised, computationally efficient, and *transferable*, meaning it can be trained on one model, and used to attack another, allevi-

ating the need to access the target model altogether. To realize the method, our first insight is to turn to representation learning and data augmentations. In the representation learning setting, data points are embedded as spread away from each other as possible, regardless of labeling, while being agnostic to predetermined or learned augmentations [3, 56]. These augmentations can be considered evasion attacks generated to strengthen the representation during training. For example, the ViewMaker framework, an efficient, simple-to-implement, self-supervised generative approach for enhancing a representation learning process through automatic augmentation, was proposed by Tamkin *et al*. [63]. In essence, the method predicts a perturbation of fixed strength that is most likely to change the representation of the input image. Using this framework, instead of learning a representation encoder, we learn augmentations that are designed to attack our pretrained target model. This results in a self-supervised, efficient, EAs generator.

Next, in order to promote transferability, our second insight is on-manifold generation. Intuitively, since on-manifold attacks are closer to the source distribution, they are attacks that are conceptually, affected more by the data distribution itself, rather than the target model and it's parameters. Hence, such attacks are effective for any model that is trained on the same data. In OMG-ATTACK, we encourage the EAs to remain on the data manifold, producing attacks that are closer to the data distribution and, in practice, seem more natural, as shown in Figure 2. We do this by adding a discriminator to the aforementioned ViewMaker framework [63]. This forces our attack generator to produce images that are indistinguishable from the source distribution in the eyes of the discriminator, hence producing on-manifold EAs. Indeed, we empirically show that on-manifold examples are transferable to unseen models, and even to new classes within the same dataset, that the OMG-ATTACK model has not seen during training.

Through a series of experiments, we show that on-manifold attacks (Figure 1) are highly transferable, demonstrating robustness to cross-model evaluation on several datasets. Compared to established baselines in off- and on-manifold EAs generation, we especially show promising attack success rates in the black-box settings on unseen defended and un-defended models. Additionally, we show superior performance for classes unseen during training for all of the evaluated data.

Overall, these experiments show that on-manifold EAs are highly transferable and, in conjunction with a self-supervised training process, offer a viable solution for real-life use cases.

2. Related Work

Adversarial Examples Early adversarial example generation techniques, such as L-BFGS [62], FGSM [26], and BIM & ICML [39] attacks, primarily focused on a *white-box* context, where the attacker has full model insight including architecture, parameters, gradients, and training data. This led to the development of gradient-based methods, a current research focus.

Figure 2. Example of the OMG-ATTACK applied to each dataset. The upper row presents the original images, while the lower row shows the corresponding attacked images. The bottom right of each image exhibits the introduced perturbation.

Prominent white-box methods include FSGM [26], its iterative variants [19, 69], and I-FGSM by Madry *et al*. [47], using the Projected Gradient Decent (PGD) method. While most attacks constrained their L_2 or L_∞ norms to maintain clean image visuals, others [51, 60] opted to generate perturbations via the network's forward gradient and saliency map.Our methodology aligns closely with the *semi-white-box* or *grey-box* attacks [71, 74, 67, 68]. In these methods, a generative model that has been trained (e.g., [23]) is used to construct adversarial perturbations. This introduces an innovative approach where full access to the model is only required during training. During the inference phase, there's no need for access to the target model since the generator is already trained. In contrast to earlier works, our model leverages self-supervision in a realistic setting and ensures that the perturbations follow an approximated manifold.

Black-box attacks present the most challenging scenario. These are further divided into score-based settings, where the attacker has access to the prediction distribution or confidence scores [35, 45], and more stringent decision-based settings, where the attack model solely depends on the prediction's label [64, 5, 17, 17]. Query-based attacks are dominant in decision-based approaches. They estimate the gradient using zeroth-order optimization methods when the loss values can be queried, as proposed by Chen *et al*. [11].

Although our work can be applied in a black-box setting during testing, it adheres to a stricter real-world setting where the attacker is decision-based and does not incorporate additional parameters or training steps, as in the subtitle transfer model approaches.

On-Manifold Adversarial Examples The topic of generative adversarial examples has seen considerable interest [10, 1, 73]. Of particular novelty is the generation of adversarial examples that adhere to the data manifold, a contrast to traditional adversarial attacks known to deviate from it [57]. Gilmer *et al*. [22] provided initial evidence of the existence of on-manifold adversarial examples, and subsequent research has underscored the theoretical advantages of these examples in enhancing model robustness [53, 61].

Goodfellow *et al*. [25] suggested that optimally balancing the generator's predictive capabilities and the discriminator's enforcement of on-manifold constraints requires the generator to mimic the original data distribution. Inspired by this, Li *et al*. [40] proposed manifold projection methods using generative models. Stutz *et al*. [59] introduced the concept of per-class on-manifold adversarial examples, though its application is limited in the absence of class labels. Several studies have leveraged generative models, such as Generative Adversarial Networks (GAN) [23] and Variational Auto-Encoders (VAE) [70], to generate on-manifold examples by introducing adversarial alterations to the embedding space [48, 52, 42]. Others, including AdvGan [71] and AdvGAN++ [48], ensure perturbations remain on-manifold by imposing constraints, while some approaches rely on predefined semantic attributes [35].

Our method also uses a GAN-like structure akin to AdvGAN but differentiates itself through the application of the more modern viewmaker backbone, yielding better empirical results (§4.2). Finally, IDAA [75] underscored the potential of on-manifold adversarial examples in achieving superior performance. In our research, we corroborate these findings through comprehensive evaluations, demonstrating that adherence to the data manifold can significantly boost the transferability of models to unseen models without necessitating additional training.

Self-supervised Adversarial Attacks Self-supervised learning, a methodology that involves training a model without predefined labels, has demonstrated significant potential across various computer vision tasks [21, 50, 12, 29]. There are several paradigms that have been adopted in the realm of self-supervised learning. *Instance Discrimination*, as exemplified in DINO [9], BYOL [27], and SimCLR [37], makes use of common image processing augmentations to generate varied renditions of the same image. The objective is for models to create comparable representations or activations for these diverse versions. *Masked Prediction* has found widespread use across fields such as Natural Language Processing [16, 43], Computer Vision [2], and more. *Transformation Prediction* is another paradigm centered around predicting image rotations [21]. Lastly, *Clustering* methods have also been utilized in this space [7].

The study of adversarial examples in representation learning has led to intriguing findings, with recent research indicating that the generation of adversarial examples can bolster the performance of self-supervised representation learning frameworks. The work demonstrated in CLAE [32], for instance, established that generating adversarial examples could improve the representation learning process. Similarly, the authors of the ViewMaker network [63] proposed that adversarial examples could be effectively used as a substitute for predefined expert augmentations.

In our study, we rely on and adapt the foundational model proposed in [63] to facilitate the generation of self-supervised adversarial attacks.

3. The OMG-ATTACK Model

We now introduce our method of generating on-manifold non-targeted adversarial examples. In Figure 3 we show our pipeline for learning and generating adversarial augmentations that are on-manifold examples.

3.1. Problem Definition

Let $X \subset I^{n \times n}$ be the image dataset of dimensions $n \times n$. Suppose that (x_i, y_i) is the i-th instance within the training set, which is comprised of an image $x_i \in X$, generated according to some unknown distribution $x_i \sim P_{data}$, and $y_i \in Y$ the corresponding true class labels. The learning system aims to learn a classifier $f : X \to Y$ from the domain X to the set of classification outputs Y, where $|Y|$ denotes the number of possible classification outputs. Given an instance $x \in X$, the goal of a regular adversary is to generate adversarial example x_A, which is misclassified $f(x_A) \neq y$ (i.e. untargeted attack),

In addition, the definition of a Manifold $M_y \subseteq I^{n \times n}$ is

$$M_y = \{x \in I^{n \times n} : C_y(x) = True\}$$

where C_y is the perfect binary classifier that decides if an image is of class y or not. Thus $M := \bigcup_y M_y$ is the union of all classes manifolds in the dataset.

Therefore we further augment the regular adversary definition to include the on-manifold constraint. We aim to generate an adversarial example x_A, which is misclassified $f(x_A) \neq y$ and if part of the dataset manifold $x_A \in M$,

3.2. Generator

We use a dense prediction network as a generative model, inspired by the viewmaker networks [63]. The architecture of the model consists of an encoder-decoder structure, with residual blocks that incorporate random noise in the form of extra channels.

Given an input image $x \in X$, the generator G produces a perturbation $G(x, \eta)$ based on the input and a random noise vector $\eta \sim \mathcal{N}(0, 1)$. The resulting perturbation is of the same dimensions and shape as the input image.

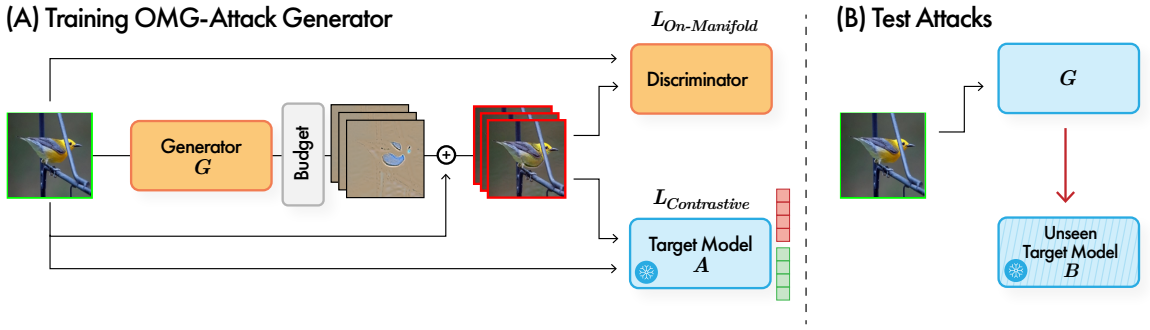


Figure 3. System Overview. (A) The Generator G creates adversarial perturbations of the input, projected onto an L_1 ball and scaled to adhere to a prescribed 'budget'. The Frozen Target Model A and the Discriminator process these images to train G. (B) the trained Generator G is used to perform adversarial attacks on the Unseen Target Model B.

Applying a *Budget* to Limit Perturbation Magnitude An example is regarded as adversarial if it is perceptually indistinguishable from benign input but is classified differently (i.e. misclassified) [4, 62]. Typically by using L_p-norms to measure the perceptual similarity between an adversarial input and its benign original [55, 51]. In our model, to scrutinize the region surrounding the input data point while simultaneously limiting the extent of modification, we introduce a fixed budget size for the perturbations. This budget size sets the limits for the perturbation magnitude, which is achieved by projecting the perturbations onto an L_1 ball and scaling it according to the pre-set budget size. The perturbation is subsequently defined as:

$$\delta = budget * \frac{G(\hat{X}, \eta)}{AvgMag(G(\hat{X}, \eta))} \tag{1}$$

where $AvgMag(v) = \frac{|v|_1}{n}$ and $|v|_1 = \sum_i^n |v_i|$.

Finally, the perturbation is superimposed onto the input image to create an adversarial example with augmentation, denoted as $X_{adv} = clamp(X + \delta, 0, 1)$. Here, $clamp$ ensures the resulting perturbation remains within the acceptable image domain. Typically the norm L_∞ is employed [36], which means every pixel is individually limited in the amount it can change. However, this is prohibitive for on-manifold attacks, where the preferred mode concentrating the budget on strategic regions of the image. Indeed, in our experiments training did not converge when using L_∞, as the discriminator was always able to identify the generated imagery. We therefore use the L_1 norm, as derived from the original view maker framework.

3.3. Discriminator

The discriminator, within our framework, is tasked with the crucial responsibility of learning the data manifold. Our discriminator's architecture is built upon the principles of the PatchGAN discriminator [33], which primarily penalizes structural inconsistencies at the scale of local image patches. It aims to classify the authenticity of each patch in an image, categorizing them as either 'real' or 'fake'.

The PatchGAN discriminator operates convolutionally across the image, averaging all responses from individual patches to generate its final output. This approach essentially models the image as a Markov Random Field, assuming independence between pixels separated by more than a patch diameter.

Let I be an input image and D be our PatchGAN discriminator. The output from D is a score matrix $D(I) \in \mathbf{R}^{n \times n}$, where each element $D(I)_{i,j}$ corresponds to the score attributed to a patch of size $p \times p$ centered at pixel (i, j) in the input image I. The binary cross entropy loss (BCE) is utilized for each patch prediction:

$$BCE(p, y) = y \log(p) + (1 - y) \log(1 - p)$$

Penalize if a fake image is considered real, and vice versa (*"real"* label is 1):

$$l_{real} = BCE(D(x_{fake}), 1), l_{fake} = BCE(D(x_{real}), 0)$$

The final discriminator loss is the average of all patch losses:

$$L_{Disc} = \frac{1}{2}(\frac{1}{n \times n} \sum l_{real} + \frac{1}{n \times n} \sum l_{fake})$$

3.4. Target Model

Throughout the learning and training processes, our methodology retains full access to the target model. Notwithstanding, our distinctively crafted generator has the capacity to execute an attack during the inference stage, irrespective of the absence of the target model's loss. This aligns with the *"semi-whitebox"* setting as detailed in [71]. Opting not to utilize the logits from the target model, we

leverage its penultimate representations instead. This strategic choice could feasibly extend the applicability of our framework to tasks beyond classification.

3.5. Training via Self-Supervision

Our pipeline uniquely operates on the learned representations rather than the classification layer, thereby obviating the need for labeled datasets. The training process for our pipeline is entirely self-supervised, leveraging two independent objectives to cultivate the generator's ability to manufacture adversarial, yet on-manifold, perturbations.

Contrastive Loss Instead of meddling with the predictions of the target model, our focus is on perturbing the learned representations. To accomplish this, we adopt a contrastive loss framework, a common technique in self-supervised and representation learning [37, 8].

The objective here is to construct analogous representations for augmentations derived from the same source. Each input is associated with a positive example, while all other data points in the batch are regarded as negative examples. The loss is assembled based on the data points' similarity, with the optimal loss being obtained when the positive pair's similarity is 1 and the similarity with all other inputs is 0.

The contrastive loss, for N groups of mutually exclusive positive pairs, is expressed as shown in the following:

$$L_{Pairs} = \frac{1}{2N} \sum_{k=1}^{N} l(2k-1, 2k) + l(2k, 2k-1)$$

$$l(i,j) = -\log \frac{\exp\left(s_{i,j}/\tau\right)}{\sum_{k=1}^{2N} 1_{[k \neq i]} \exp\left(s_{i,k}/\tau\right)}$$

and $s_{a,b}$ is the cosine similarity of the representations of examples a and b, and τ is a temperature parameter.

For all inputs in the batch, we use the generator to produce adversarial examples $X^i_{adv} = X^i + G(X^i, \eta)$. Examples generated from the same image are called positives, while all other examples in the batch (originals and generated) are considered to be negatives.

We compute this contrastive loss for three groups of positive pairs permutations: (X, X_{adv_1}), (X, X_{adv_2}), and (X_{adv_1}, X_{adv_2}). This results in three losses:

$$L_{Contrastive} = L^{(X, X_{adv1})}_{Pairs} + L^{(X, X_{adv2})}_{Pairs} + L^{(X_{adv1}, X_{adv2})}_{Pairs}$$

Applying L_{Pairs} between the two generated examples (X_{adv1}, X_{adv2}) encourages the model to learn to produce diverse augmentations and avoid settling on a deterministic specific perturbation per input.

On-Manifold Loss Our approach for generating on-manifold, in-distribution augmentations involves incorporating an auxiliary loss function that leverages a discriminator model. Previous studies indicate that discriminators can efficiently learn the underlying manifold of a training dataset [24, 34].

To integrate this into our methodology, we introduce an on-manifold loss as an adversarial objective for our generative network. The generative network aspires to generate augmentations that the discriminator deems part of the original dataset. As the discriminator is trained, it learns from its misclassifications and continually refines its comprehension of the dataset's underlying manifold.

Therefore, the on-manifold loss is derived from the discriminator's loss. It marks the areas where the generator needs to improve upon the fake patches detected by the discriminator: $L_{On-Manifold} = BCE(D(x_{fake}), 0)$.

Total Loss The overall loss employed to train the generator is a combination of both the adversarial iteration of the contrastive loss and the on-manifold loss, which is the adversarial iteration of the discriminator objective: $L_{total} = \alpha L_{On-Manifold} - \beta\, L_{Contrastive}$

Detailed information on the models' architectures and hyperparameters utilized in our study is provided in the supplementary material.

4. Experiments

The goal of our experiments is to assess the efficacy of our model in generating adversarial examples. To that end, we evaluate our method by conducting experiments in various settings, including the white-box setting, transferability to black-box models and architectures, including defended target models, and transferability of attacks to unseen data classes at train time.

4.1. Experimental Setup

Datasets & Models We evaluate our pipeline on three datasets from different domains, using target models of various scales and architectures for the classification task. The datasets used are presented in Table 1.

For each dataset, we train OMG-ATTACK (comprising a discriminator and generator) against a pre-trained model on the classification task of that particular dataset. For the MNIST dataset, we use the same target network architecture based on a Convolutional Neural Network (CNN) as used by Carlini & Wagner [6] (denoted as MNIST-CNN), which has 1.2M parameters. For the German Traffic Sign Recognition Benchmark (GTSRB), we use a CNN with three Spatial Transformers [20], which achieves over 99% accuracy on this dataset (denoted as TS-CNN-STN) with 855K parameters. Lastly, for the Caltech Birds (CUB-200)

Dataset	Resolution	#Classes	Data Type	Size
MNIST	28x28	10	Digits	60K
GTSRB	32x32	43	Traffic Signs	40K
CUB-200	224x224	200	Birds	12K

Table 1. An overview of the datasets used in our experiments. MNIST [15] is a dataset of handwritten digits. GTSRB (German Traffic Sign Recognition Benchmark) [58] is a dataset consisting of images of traffic signs. CUB-200 [31] is the Caltech Birds dataset that includes images of 200 different bird species.

dataset, we use the popular ResNet-18 model [30] with 11.3M parameters (denoted as CUB-ResNet18).

As these models were used in training, we evaluate attacks on them under *white-box* settings, where we have full access to the model parameters.

Under the *black-box* settings, we use additional models to examine transferability. We leverage the OMG-ATTACK models trained under the white-box setting on each dataset separately and generate attacks for other unseen models. For MNIST, we use two models proposed by [65] (denoted as MNIST-TR1 and MNIST-TR2) as well as a larger-scale ResNet-18 trained on MNIST. For GTSRB, we validate against a larger-scale ResNet-50. Lastly, for the CUB-200 dataset, we evaluate against a larger-scale ResNet-50 and WideResNet-50.

Further details about the models, including the hyperparameters used, are provided in the supplementary material.

Baselines To comprehensively evaluate the effectiveness of our proposed model, we draw comparisons with an array of adversarial attack methods that span from simple, yet effective strategies to more complex and robust methods. This range of baselines allows us to benchmark our model against both traditional and state-of-the-art techniques.

1. **Fast Gradient Sign Method (FGSM)** [26]: The method utilizes the gradients of the loss function with respect to the input data to create adversarial perturbations.

2. **Projected Gradient Descent (PGD)** [46]: Building upon FGSM, PGD introduces iterative attacks to adversarial research. Unlike FGSM, PGD applies the perturbations multiple times within an epsilon ball, offering more potent adversarial examples with enhanced robustness.

3. **Momentum Iterative Fast Gradient Sign Method (MI-FGSM)** [18]: This advanced adversarial attack method infuses momentum into the iterative FGSM process. The momentum term allows the method to maintain the direction of gradient update across iterations, thereby enhancing both the effectiveness and robustness of the generated adversarial examples.

4. **Variance-Tuning Momentum Iterative Fast Gradient Sign Method (VMI-FSGM)** [69]: As an extension to MI-FGSM, VMI-FGSM introduces a variance term to dynamically adjust the step size throughout the iterative process. This adaptive approach yields highly potent examples, thus placing VMI-FGSM among the most powerful adversarial attack methods to date.

5. **ADVGAN** [71]: A generative adversarial network-based method for creating adversarial examples. It employs a similar domain of operation as our proposed model, thus providing a pertinent comparison.

Evaluation Metrics We use the Attack Success Rate (ASR) to evaluate the effectiveness of our adversarial augmentations. ASR is defined as the fraction of adversarial examples that successfully cause the target model to change its prediction. Formally,

$$ASR = \frac{|\{X_{adv}|f(X) \neq f(X_{adv})\}|}{|\{X_{adv}\}|}$$

where X_{adv} is an adversarial example generated from X. ASR is typically expressed as a percentage, with higher values indicating a more effective attack.

Budget Selection Assuring comparability of the budgets across different methods is challenging, given the disparate L_p norms utilized by these methods. Thus, we conducted an examination where we studied the influence of augmentations under identical budgets across various methods. For each dataset, we used the same budget across all methods evaluated: 0.3 for the MNIST dataset, 0.015 for the GTSRB dataset, and 0.025 for the CUB-200 dataset.

For the selected budget values, we found that the difference in the intensity of the augmentations (i.e., $P = 1/(nn) * \sum(I(x,y))^2$ for each pixel (x,y) in the image I) is relatively small, less than 10% among the methods. We discuss this challenge further in §6, and additional information on the budget analysis and selection process can be found in §5.

Base-line White-box Setting We compare the ASR performance of the trained models in the white-box settings against the different baselines and present the results in Table 2. As can be seen, the performance of our method is comparable to other approaches in this setting.

Dataset	Model	PGD	FGSM	MI-FGSM	VMI-FSGM	AdvGAN	OMG-Attack
MNIST	MNIST-CNN	**99.87**	87.48	99.86	**99.87**	89.27	99.37
GTSRB	TS-CNN-STN	33.53	19.71	33.83	33.91	20.25	**67.48**
CUB-200	CUB-ResNet18	**94.27**	92.85	93.65	93.89	75.09	46.97

Table 2. White-box setting comparison. The Attack Success Rates (ASR) is reported, where all models are allocated a similar resource limit (i.e., budget). Without transferability, performance is comparable between models.

Dataset	Def. Model	PGD	FGSM	MI-FGSM	VMI-FSGM	AdvGAN	OMG-Attack
MNIST	MNIST-CNN	6.25	0.32	3.62	3.14	8.41	**99.33**
GTSRB	TS-CNN-STN	16.75	8.71	17.41	17.65	6.99	**64.58**
CUB-200	CUB-ResNet18	12.65	14.58	19.35	25.43	4.75	**38.94**

Table 3. Transferability to defended models. The Success Rates (ASR) is reported for attacks generated using undefended versions on models, but test on the models defended using adversarial training with FGSM.

Dataset	Model	PGD	FGSM	MI-FGSM	VMI-FSGM	AdvGAN	OMG-Attack
MNIST	MNIST-TR1	76.98	55.84	80.65	84.15	82.64	**99.29**
	MNIST-TR2	60.59	85.06	88.12	90.27	72.35	**99.56**
	ResNet-18	70.91	56.43	72.49	73.60	72.32	**89.65**
GTSRB	ResNet-50	10.74	8.70	11.65	11.76	9.01	**58.89**
CUB-200	ResNet-50	31.12	37.90	45.84	**54.20**	17.63	48.33
	WideResNet-50	25.14	33.42	38.67	46.02	19.93	**46.84**

Table 4. Transferability to different architectures and scales. The attacks, originally generated for the white-box settings, are tested for their transferability to more robust black-box target models. The presented results are Attack Success Rate (ASR) percentages.

4.2. Transferabilty to Unseen Models

These experiments have been conducted in a black-box setting, where the target model's parameters, logits, and architecture are inaccessible.

Defended Models These subsequent experiments aimed to evaluate the robustness of the adversarial examples (originally generated for the undefended model) in manipulating the predictions of a defended model. The defensive strategy deployed in these experiments involved training the original model using a mixture of original data and adversarial examples produced by the FGSM method on the static undefended model. Our approach outperformed other strategies in all the scenarios, demonstrating that the defensive method had a substantially reduced impact on our adversarial examples. The results are presented in Table 3.

Transferability to Architectures and Greater Scales The black-box transfer experiment was designed to examine the efficacy of our generated adversarial examples when tested against unseen and more sophisticated models. The goal was to ascertain whether these adversarial examples specifically undermine the unique target model they were trained on or whether they target challenging, underrepresented areas in the data manifold that pose inherent classification challenges for any model.

To evaluate this, we devised a simple methodology that applies our data augmentations and assesses the performance of different models, independently trained from the OMG-Attack model, on these augmentations. The results are tabulated in Table 4, where we observe that our augmentations achieve an ASR higher than the baseline by 10%-40%. The high ASR values suggest that the tested models encounter difficulties in accurately classifying the augmented data and do not significantly outperform the original white-box target model. This suggests that the adversarial examples are generalizable to unseen models. Notably, all tested models were more sophisticated, possessing a larger parameter space, supporting our hypothesis that the adversarial examples are located in less represented regions of the dataset's manifold.

4.3. Transfrability to Unseen Data

Another key advantage of our system is the ability to generate adversarial augmentations for classes that have not been seen by the target model. To evaluate this capability, we conduct an experiment where the target model is trained on only 80% of the classes in a dataset. Subsequently, OMG-Attack is trained model, and utilized to generate augmentations for the entire dataset, including the previously unseen 20%. We can see that our method consis-

Dataset	AdvGAN			OMG-Attack		
	Train	Seen	Unseen	Train	Seen	Unseen
MNIST	81.03	72.20	31.69	85.56	84.25	**46.49**
GTSRB	20.55	12.89	4.13	69.19	65.54	**60.61**
CUB-200	31.11	24.21	24.42	66.98	63.90	**70.09**

Table 5. Domain transfer from 80% to 20%. The 'Train' column represents the ASR percentage on the truncated dataset, 'Seen' shows the ASR percentage on the untruncated dataset but only on the subset with seen classes, and 'Unseen' is the ASR percentage on the subset with unseen classes.

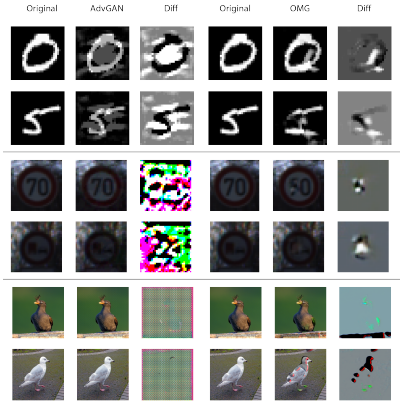


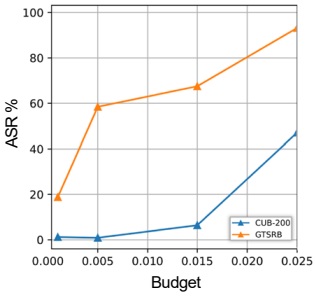
Figure 4. Input examples from classes that were not seen when the model was trained, show the ability of OMG-Attack to generalize to unseen data classes.

tently outperforms the baseline, by over 10%-50% in ASR. The results are reported in Table 5, and depicted in Figure 4.

5. Budget Analysis

The budget determines how much of the image can be altered, or how much of the signal must be preserved, and its magnitude greatly influences the model's performance. The higher the budget, more successful attacks (higher ASR) can be anticipated, but at the cost deviation from the original images and their content.

In the inset figure, we illustrate how varying the budget size influences the attack performance, particularly in terms of the ASR.



Generally, a larger budget tends to produce adversarial examples that are more challenging for the model to classify correctly.

This observation aligns with expectations since a larger budget provides the generator with greater flexibility to drastically alter the original image.

6. Limitations

The primary limitation of our study revolves around the issue of budget selection. Our approach deviates from the majority of comparable baselines due to the unique normalization function we employed. While most standard baselines—including but not limited to AdvGAN—leverage the L_∞ norm for budget allocation, our method uniquely utilizes the L_1 norm. Despite our concerted efforts to ensure similar budget allocations across methods for fair comparison, we acknowledge that the comparison might not be entirely balanced due to the disparate norms used.

A further limitation is that certain adversarial examples generated by our method, particularly within the MNIST dataset, lead to modifications that are noticeably perceptible to human observers. This challenges their classification as genuine adversarial examples, suggesting that a more rigorous budget constraint might be necessary for these types of datasets. This observation echoes the findings of previous research that has shown perceptible variations in MNIST dataset images, even when the L_p norm of their differences is below thresholds utilized in earlier works [55]. The evidence, therefore, suggests that these changes can still significantly alter human perception of the images.

In conclusion, while our method has shown promising results, these limitations highlight the need for further exploration in the area of budget allocation and the generation of truly imperceptible adversarial examples.

7. Conclusions

Adversarial augmentations serve as an insightful method for uncovering weaknesses in machine learning models. By directing adversarial perturbations to adhere to the data manifold, we can design data-dependent attacks that have transferability to various models.

In this study, we proposed a self-supervised framework for learning and creating such attacks. Our approach draws upon adversarial training, guiding the generator to produce perturbations that challenge the encoder while also remaining within the bounds of the manifold.

Our experiments illustrate the potential of our method in generating transferable adversarial examples, providing a closer look at the visual semantics these augmentations carry. Our findings highlight the need to account for data-dependent attacks in developing and evaluating machine-learning models. Overall, our work adds to the ongoing discussions about adversarial attacks and their effects on machine-learning models. We hope our insights encourage further research in this field, leading toward machine learning models that are more resilient to adversarial perturbations.

References

[1] Naveed Akhtar and Ajmal Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *Ieee Access*, 6:14410–14430, 2018. 3

[2] Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. data2vec: A general framework for self-supervised learning in speech, vision and language. In *International Conference on Machine Learning*, 2022. 3

[3] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013. 2

[4] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Srndic, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *ECML/PKDD*, 2013. 4

[5] Thomas Brunner, Frederik Diehl, Michael Truong-Le, and Alois Knoll. Guessing smart: Biased sampling for efficient black-box adversarial attacks. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4957–4965, 2018. 2

[6] Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57, 2016. 1, 5

[7] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *ArXiv*, abs/2006.09882, 2020. 3

[8] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments, 2020. 5

[9] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 1, 3

[10] Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. A survey on adversarial attacks and defences. *CAAI Transactions on Intelligence Technology*, 6(1):25–45, 2021. 3

[11] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models, 2017. 2

[12] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *ArXiv*, abs/2002.05709, 2020. 3

[13] Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Edoardo Debenedetti, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. *ArXiv*, abs/2010.09670, 2020. 1

[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*, 2020. 1

[15] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. 6

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 3

[17] Hadi M. Dolatabadi, Sarah Monazam Erfani, and Christopher Leckie. Advflow: Inconspicuous black-box adversarial attacks using normalizing flows. *ArXiv*, abs/2007.07435, 2020. 2

[18] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9185–9193, 2017. 6

[19] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li. Boosting adversarial attacks with momentum. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9185–9193, Los Alamitos, CA, USA, jun 2018. IEEE Computer Society. 1, 2

[20] Álvaro Arcos García, Juan Antonio Álvarez-García, and Luis Miguel Soria-Morillo. Deep neural network for traffic sign recognition systems: An analysis of spatial transformers and stochastic optimisation methods. *Neural networks : the official journal of the International Neural Network Society*, 99:158–165, 2018. 5

[21] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *ArXiv*, abs/1803.07728, 2018. 3

[22] Justin Gilmer, Luke Metz, Fartash Faghri, Samuel S. Schoenholz, Maithra Raghu, Martin Wattenberg, and Ian J. Goodfellow. Adversarial spheres. *ArXiv*, abs/1801.02774, 2018. 3

[23] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. 2, 3

[24] 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 processing systems*, 27, 2014. 5

[25] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 3

[26] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572, 2014. 1, 2, 6

[27] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach

to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020. 3

[28] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022. 1

[29] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9726–9735, 2019. 3

[30] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2015. 6

[31] Xiangteng He and Yuxin Peng. Fine-grained visual-textual representation learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(2):520–531, feb 2020. 6

[32] Chih-Hui Ho and Nuno Vasconcelos. Contrastive learning with adversarial examples, 2020. 3

[33] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, 2016. 4

[34] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks, 2017. 5

[35] Ameya Joshi, Amitangshu Mukherjee, Soumik Sarkar, and Chinmay Hegde. Semantic adversarial attacks: Parametric transformations that fool deep classifiers, 2019. 2, 3

[36] Shashank Kotyan and Danilo Vasconcellos Vargas. Adversarial robustness assessment: Why in evaluation both l0 and l inf attacks are necessary. *PLoS ONE*, 17, 2022. 4

[37] Ajay Krishnan. Eli5: A simple framework for contrastive learning of visual representations, 2021. 3, 5

[38] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *ArXiv*, abs/1611.01236, 2016. 1

[39] Kimin Lee, Jaehyung Kim, Song Chong, and Jinwoo Shin. Making stochastic neural networks from deterministic ones, 2017. 2

[40] Yao Li, Minhao Cheng, Cho-Jui Hsieh, and Thomas CM Lee. A review of adversarial attack and defense for classification methods. *The American Statistician*, 76(4):329–345, 2022. 3

[41] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 1

[42] Guanxiong Liu, Issa Khalil, Abdallah Khreishah, Abdulelah Algosaibi, Adel Aldalbahi, Mohammed Alnaeem, Abdulaziz Alhumam, and Muhammad Anan. Manigen: A manifold aided black-box generator of adversarial examples. *IEEE Access*, 8:197086–197096, 2020. 3

[43] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 3

[44] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015. 1

[45] Chen Ma, L. Chen, and Junhai Yong. Simulating unknown target models for query-efficient black-box attacks. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11830–11839, 2020. 2

[46] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *ArXiv*, abs/1706.06083, 2017. 6

[47] 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. 1, 2

[48] Puneet Mangla, Surgan Jandial, Sakshi Varshney, and Vineeth N. Balasubramanian. Advgan++: Harnessing latent layers for adversary generation. *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 2045–2048, 2019. 3

[49] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2574–2582, 2016. 1

[50] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, 2016. 3

[51] Nicolas Papernot, Patrick Mcdaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 372–387, 2015. 2, 4

[52] Kanil Patel, William Beluch, Dan Zhang, Michael Pfeiffer, and Bin Yang. On-manifold adversarial data augmentation improves uncertainty calibration, 2019. 3

[53] Kanil Patel, William H. Beluch, Dan Zhang, Michael Pfeiffer, and Bin Yang. On-manifold adversarial data augmentation improves uncertainty calibration. *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 8029–8036, 2019. 3

[54] Rulin Shao, Zhouxing Shi, Jinfeng Yi, Pin-Yu Chen, and Cho-Jui Hsieh. On the adversarial robustness of vision transformers. *Trans. Mach. Learn. Res.*, 2022, 2022. 1

[55] Mahmood Sharif, Lujo Bauer, and Michael K. Reiter. On the suitability of lp-norms for creating and preventing adversarial examples. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1686–16868, 2018. 4, 8

[56] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019. 2

[57] Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. *arXiv preprint arXiv:1710.10766*, 2017. 3

[58] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural networks*, 32:323–332, 2012. 6

[59] David Stutz, Matthias Hein, and Bernt Schiele. Disentangling adversarial robustness and generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6976–6987, 2019. 3

[60] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23:828–841, 2017. 2

[61] Arun Sai Suggala, Adarsh Prasad, Vaishnavh Nagarajan, and Pradeep Ravikumar. Revisiting adversarial risk. In *International Conference on Artificial Intelligence and Statistics*, 2018. 3

[62] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, D. Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *CoRR*, abs/1312.6199, 2013. 1, 2, 4

[63] Alex Tamkin, Mike Wu, and Noah Goodman. Viewmaker networks: Learning views for unsupervised representation learning, 2020. 2, 3

[64] Giorgos Tolias, Filip Radenović, and Ondrej Chum. Targeted mismatch adversarial attack: Query with a flower to retrieve the tower. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5036–5045, 2019. 2

[65] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Dan Boneh, and Patrick Mcdaniel. Ensemble adversarial training: Attacks and defenses. *ArXiv*, abs/1705.07204, 2017. 6

[66] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8769–8778, 2018. 1

[67] BS Vivek, Konda Reddy Mopuri, and R Venkatesh Babu. Gray-box adversarial training. In *Proceedings of the European conference on computer vision (ECCV)*, pages 203–218, 2018. 2

[68] Hanrui Wang, Shuo Wang, Zhe Jin, Yandan Wang, Cunjian Chen, and Massimo Tistarelli. Similarity-based gray-box adversarial attack against deep face recognition. In *2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021)*, pages 1–8. IEEE, 2021. 2

[69] Xiaosen Wang and Kun He. Enhancing the transferability of adversarial attacks through variance tuning. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1924–1933, 2021. 2, 6

[70] Max Welling and Diederik P Kingma. Auto-encoding variational bayes. In *ICLR*, 2014. 3

[71] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating adversarial examples with adversarial networks. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 3905–3911, 2018. 2, 3, 4, 6

[72] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 1

[73] Han Xu, Yao Ma, Hao-Chen Liu, Debayan Deb, Hui Liu, Ji-Liang Tang, and Anil K Jain. Adversarial attacks and defenses in images, graphs and text: A review. *International Journal of Automation and Computing*, 17:151–178, 2020. 3

[74] Ying Xu, Xu Zhong, Antonio Jimeno Yepes, and Jey Han Lau. Grey-box adversarial attack and defence for sentiment classification. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4078–4087, Online, June 2021. Association for Computational Linguistics. 2

[75] Kaiwen Yang, Tianyi Zhou, Xinmei Tian, and Dacheng Tao. Identity-disentangled adversarial augmentation for self-supervised learning, 2022. 3