Semantic Perturbations with Normalizing Flows for Improved Generalization

Oğuz Kaan Yüksel[†] Sebastian U. Stich[†] Martin Jaggi[†] Tatjana Chavdarova[†,‡]
[†] Machine Learning and Optimization Lab, EPFL
[‡] Department of Electrical Engineering and Computer Sciences, UC Berkeley

Abstract

Data augmentation is a widely adopted technique for avoiding overfitting when training deep neural networks. However, this approach requires domain-specific knowledge and is often limited to a fixed set of hard-coded transformations. Recently, several works proposed to use generative models for generating semantically meaningful perturbations to train a classifier. However, because accurate encoding and decoding are critical, these methods, which use architectures that approximate the latent-variable inference, remained limited to pilot studies on small datasets.

Exploiting the exactly reversible encoder-decoder structure of normalizing flows, we perform on-manifold perturbations in the latent space to define fully unsupervised data augmentations. We demonstrate that such perturbations match the performance of advanced data augmentation techniques—reaching 96.6% *test accuracy for CIFAR-10 using ResNet-18 and outperform existing methods, particularly in low data regimes—yielding* 10–25% *relative improvement of test accuracy from classical training. We find that our latent* adversarial *perturbations adaptive to the classifier throughout its training are most effective, yielding the first test accuracy improvement results on real-world datasets—CIFAR-10/100—via latent-space perturbations.*

1. Introduction

Deep Neural Networks (DNNs) have shown impressive results across several machine learning tasks [17, 40], and—due to their automatic feature learning—have revolutionized the field of computer vision. However, their success depends on the availability of large annotated datasets for the task at hand. Thus, among other overfitting techniques—such as L1/L2 regularization, dropout [52], early stopping, among others—data augmentation remains a mandatory component that is frequently used in practice.

Traditional data augmentation (DA) techniques apply a predefined set of transformations to the training samples that do not change the corresponding class label, to increase the number of training samples. As this approach is limited to making the classifier robust only to the fixed set of hard-coded transformations, advanced methods incorporate more loosely defined transformations in the data space. For example, *mixup* [66] uses convex combinations of pairs of examples and their labels, and *cutout* [9] randomly masks square regions of the input sample. Albeit implicitly, these methods still require domain-specific knowledge that, for example, such masking will not change the label.

Surprisingly, in the context of computer vision, it has been shown that small perturbations in image space that are not visible to the human eye can fool a well-performing classifier into making wrong predictions. This observation motivated an active line of research on adversarial training [see 4, and references therein]—namely, training with such adversarial samples to obtain robust classifiers. However, further empirical studies showed that such training reduces the training accuracy, indicating the two objectives are competing [58, 54].

Stutz et al. [53] postulate that this robustness-generalization trade-off appears due to using off-manifold adversarial attacks that leave the data-manifold and that *on-manifold adversarial attacks* can improve generalization. For verifying this hypothesis, the authors proposed to use perturbations in the latent space of a generative model. Their proposed method employs (class-specific) models named VAE-GANs [33, 48]—which are based on Generative Adversarial Networks [16] and, to tackle their non-invertibility, further combine GANs with Variational Autoencoders [28]. However, the VAE-GAN model introduces hard-to-tune hyperparameters, and notably, it optimizes a lower bound on the log-likelihood of the data. Moreover, improved test accuracy was only shown on toy datasets [53, Fig. 5], and yet in some cases, the test accuracy did not improve relative to classical training. We observe that on real-world datasets, such training can decrease the test accuracy, see §5.

In this work, we focus on the possibility of employing advanced normalizing flows such as *Glow* [27], to define entirely unsupervised augmentations—contrasting with

Correspondence to `oguz.yuksel@epfl.ch`.

pre-defined fixed transformations—with the same goal of improving the generalization of deep classifiers. Although normalizing flows have gained little attention in our community relative to GANs and Autoregressive models, they offer appealing advantages over these models, namely: (i) exact latent-variable inference and log-likelihood evaluation, and (ii) efficient inference and synthesis that can be parallelized [27], respectively. We exploit the *exactly reversible* encoder-decoder structure of normalizing flows to perform efficient and controllable augmentations in the learned manifold space.

Contributions. Our contributions can be summarized as:

- Firstly, we demonstrate through numerical experiments that the previously proposed methods to generate on-manifold perturbations fail to improve the generalization of a trained classifier on real-world datasets. In particular, the test accuracy decreases with such training on CIFRAR-10/100. In this work, we postulate that this occurs due to *approximate* encoder-decoder mappings.
- Motivated by this observation, we propose a data augmentation method based on *exactly reversible* normalizing flows. Namely, it first trains the generative model and then uses simplistic random or adversarial domain-agnostic semantic perturbations to train the classifier, defined in §4.
- We demonstrate that our adversarial data augmentation method generates on-manifold and semantically meaningful data perturbations. Hence, we argue that our technique is a novel approach for generating perceptually meaningful (natural adversarial examples), different from previous proposals.
- Finally, we empirically demonstrate that our on-manifold perturbations consistently outperform the standard training on CIFAR-10/100 using ResNet-18. Moreover, in a low-data regime, such training yields up to 25% relative improvement from classical training, of which—as most effective—we find the adversarial perturbations that are adaptive to the classifier, see §5.

2. Related Work

Data augmentation techniques are routinely used to improve the generalization of classifiers [50, 31]. While most classic techniques require a priori expert knowledge of invariances in the dataset to generate *virtual* examples in the vicinity around each sample in the training data, many automated techniques have been proposed recently, such as linearly interpolating between images and their labels [66], replacing a part of the image with either a black-colored patch [9] or a part of another image [64].

In contrast to these data-agnostic procedures, a few recent works proposed to learn useful data augmentations policies, for instance by optimization [14, 45], reinforcement learning techniques [7, 8, 69], specifically trained augmentation networks [43, 56] or assisted by generative adversarial networks [44, 1, 68, 57], such as also in [39] that proposes neural style transfer for augmenting datasets.

Perturbations in Latent Space allow natural data augmentation with GANs. For instance, Antoniou et al. [1], Zhao et al. [70] propose to apply random perturbations in the latent space, and recently Manjunath et al. [38] used StyleGAN2 [24] to generate novel views of the image through latent space manipulation. However, a critical weakness in these techniques is that the mapping from the latent space to the training data space is typically not invertible, *i.e.*, finding the representation of a data sample in the latent space (to start the search procedure) is a non-trivial task. For instance, Zhao et al. [70] propose to separately train an inverter for the inverse-mapping to the latent space. This critical bottleneck is omitted in our approach since we rely on an invertible architecture which renders the learning of an inverter superfluous.

Latent attacks, *i.e.*, searching in the latent space to find virtual data samples that are misclassified, were proposed in [3, 51, 62, 67]. Volpi et al. [60] proposed an adaptive data augmentation method that appends adversarial examples at each iteration and note that generalization is improved across a range of a priori unknown target domains. Complementary, the connection of adversarial learning and generalization has also been studied in [55, 49, 22, 59, 15, 70]. Stutz et al. [53] clarify the relationship between robustness and generalization by showing in particular that regular adversarial examples leave the data manifold and that on-manifold adversarial training boosts generalization. These important insights endorse previous findings that data augmentation assisted by generative models—as we suggest here—can improve generalization [60].

Perceptual (or *Natural*) Adversarial examples are getting increasing interest in the community recently, as alternative to—from human perceptive—often hard to interpret standard adversarial threat models [70, 47, 61, 36, 32, 29, 13]. We argue that on-manifold perturbations, as obtained with our method or similar generative techniques, can *implicitly learn* such natural transformations and could be used as an alternative method to define and generate perceptually and semantically meaningful data augmentation. In contrast to Wong and Kolter [61] who propose to learn *perturbation* sets via a latent space of a conditional variational autoencoder using a set of predefined image-space transformations, in our approach, we are not restricted to a fixed transformation set as we utilize implicit transformations learned by the invertible mapping provided by normalizing flows.

3. Normalizing Flows and their Advantages for Semantic Perturbations

In this section, we first describe the fundamental concepts of normalizing flows. We then discuss how their ability to perform exact inference helps to apply perturbations in latent space.

3.1. Background: Normalizing Flows

Assume observations $\boldsymbol{x} \in \mathbb{R}^d$ sampled from an unknown data distribution $p_{\mathcal{X}}$ over $\mathcal{X} \subset \mathbb{R}^d$, and a tractable prior probability distribution $p_{\mathcal{Z}}$ over $\mathcal{Z} \subset \mathbb{R}^k$ according to which we sample a latent variable $\boldsymbol{z}$. Flow-based generative models seek to find an invertible, also called *bijective* function $\mathcal{F}: \mathcal{X} \to \mathcal{Z}$ such that:

$$\mathcal{F}(\boldsymbol{x}) = \boldsymbol{z} \quad \text{and} \quad \mathcal{F}^{-1}(\boldsymbol{z}) = \boldsymbol{x}\,, \tag{NF}$$

with $\boldsymbol{z} \in \mathcal{Z}$ and $\boldsymbol{x} \in \mathcal{X}$. That is, $\mathcal{F}$ maps observations $\boldsymbol{x}$ to latent codes $\boldsymbol{z}$, and $\mathcal{F}^{-1}$ maps latent codes $\boldsymbol{z}$ back to original observations $\boldsymbol{x}$.

The key idea behind normalizing flows is to use change of variables, *i.e.*, by using invertible transformation we keep track of the change in distribution. Thus, $p_{\mathcal{X}}$ induces $p_{\mathcal{Z}}$ through $\mathcal{F}$ and the opposite holds through $\mathcal{F}^{-1}$. We have:

$$p_{\mathcal{X}}(\boldsymbol{x}) = p_{\mathcal{Z}}(\mathcal{F}(\boldsymbol{x})) \cdot \left| \det \left(\frac{\partial \mathcal{F}(\boldsymbol{x})}{\partial \boldsymbol{x}^{\top}} \right) \right|,$$

where the determinant of the Jacobian $\frac{\partial \mathcal{F}(\boldsymbol{x})}{\partial \boldsymbol{x}^{\top}}$ is used as volume correction. In practice, $\mathcal{F}$ is also differentiable and is parameterized with parameters $\boldsymbol{\omega}$, we have finite samples $\boldsymbol{x}_i \sim p_{\mathcal{D}}, 1 \leq i \leq N$ and training is done via maximum log-likelihood:

$$\boldsymbol{\omega}^{\star} = \arg\max_{\boldsymbol{\omega}} \sum_{i=1}^{N} \log p_{\mathcal{Z}}(\mathcal{F}(\boldsymbol{x}_i | \boldsymbol{\omega})) + \log \left| \det \left(\frac{\partial \mathcal{F}(\boldsymbol{x}_i | \boldsymbol{\omega})}{\partial \boldsymbol{x}_i^{\top}} \right) \right|.$$

Because computing the inverse and the determinant is computationally expensive for high-dimensional spaces, $\mathcal{F}$ is constrained to linear transformations that have some structure—often chosen to be *triangular* Jacobian matrices, which provide efficient computations in both directions.

To build an expressive but tractable $\mathcal{F}$, we rely on the fact that differentiable functions are closed under composition, thus $\mathcal{F} = f_{\ell} \circ f_{\ell-1} \circ \cdots \circ f_1$, $\ell{>}1$, is also invertible. In the context of deep learning, this implies that we can stack ℓ layers of simple invertible mappings. However, as this still yields a single linear transformation, *coupling* layers [11] $f(\mathrm{x}) = \boldsymbol{y}$, with $f: \mathbb{R}^C \to \mathbb{R}^C$ are inserted, which can be defined in several ways [10, 21]. In this work we use affine coupling transforms, which are empirically shown to perform particularly well for images, and which are used in the *Glow* model [27]:

$$\boldsymbol{y}_{1:c} {=} \boldsymbol{x}_{1:c} \quad \text{and} \quad \boldsymbol{y}_{c+1:C} {=} \boldsymbol{x}_{c+1:C} \odot \exp(s(\boldsymbol{x}_{1:c})) + t(\boldsymbol{x}_{1:c})\,,$$

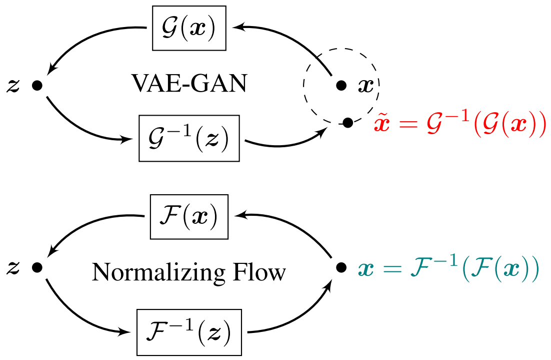
$\mathcal{G}(\boldsymbol{x})$
$\boldsymbol{z}$
VAE-GAN
$\boldsymbol{x}$
$\tilde{\boldsymbol{x}} = \mathcal{G}^{-1}(\mathcal{G}(\boldsymbol{x}))$
$\mathcal{G}^{-1}(\boldsymbol{z})$
$\mathcal{F}(\boldsymbol{x})$
$\boldsymbol{z}$
Normalizing Flow
$\boldsymbol{x} = \mathcal{F}^{-1}(\mathcal{F}(\boldsymbol{x}))$
$\mathcal{F}^{-1}(\boldsymbol{z})$

Figure 1. **Exactness of NF encoding-decoding**. Here $\mathcal{F}$ denotes the bijective NF, and $\mathcal{G}/\mathcal{G}^{-1}$ encoder/decoder pair of inexact methods such as VAE or VAE-GAN which, due to inherent decoder noise, is only approximately bijective.

where $\odot$ is the Hadamard product and, s and t are scaling and translations functions from $\mathbb{R}^c \to \mathbb{R}^{C-c}$. Moreover, the Jacobian does not require any derivative over s and t, meaning that we can model these functions with arbitrary deep neural networks. To allow that each component can change, usually, $\mathcal{F}$ is composed so that coupling layers are placed in the middle of permutation layers that work in alternating patterns. The *Glow* model [27] uses an invertible $1{\times}1$ convolution layer that generalizes this permutation operation, see Appendix A.1.

3.2. Advantages of Normalizing Flows

Most popular generative models for computer vision tasks are Variational Autoencoders [VAEs, 28] or Generative Adversarial Networks [GANs, 16].

GANs are widely used in deep learning mainly due to their impressive sample quality, as well as efficient sampling. Nonetheless, by construction, these methods do not provide an invertible mapping from an image $\boldsymbol{x}$ to its latent representation $\boldsymbol{z}$, nor estimating its likelihood under the implicitly learned data distribution $p_{\mathcal{X}}(\boldsymbol{x})$, except with significant additional compromises [25]. Moreover, despite the notable progress, designing a stable two-player optimization method remains an active research area [6].

VAEs, on the other hand, seemingly resolve these two problems as this class of algorithms is both approximately invertible and notably easier to train. However, VAEs are trained via maximizing a bound on the marginal likelihood and provide only approximate evaluation of $p_{\mathcal{X}}(\boldsymbol{x})$. Moreover, due to their worse sample quality relative to GANs, researchers propose combining the two [33, 48]—making their performance highly sensitive to their hyperparameter tuning.

In contrast, normalizing flows: (i) perform *exact* encoding and decoding—due to their construction (see above, and also the illustration in Figure 1), (ii) are highly expressive, (iii) are efficient to sample from, as well as to evalu-

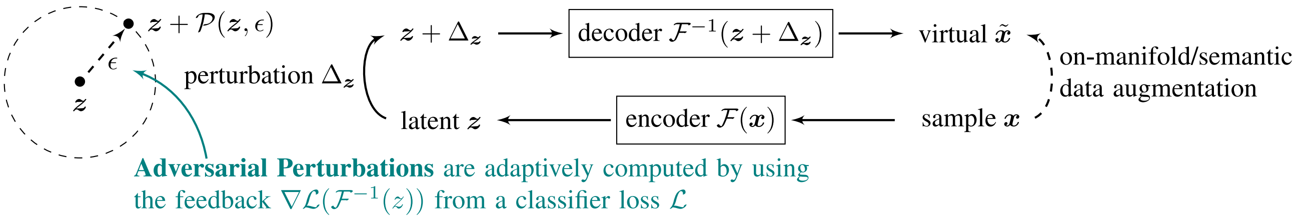
$\boldsymbol{z} + \mathcal{P}(\boldsymbol{z}, \epsilon)$
ϵ
$\boldsymbol{z}$
perturbation $\Delta_{\boldsymbol{z}}$
$\boldsymbol{z} + \Delta_{\boldsymbol{z}}$
decoder $\mathcal{F}^{-1}(\boldsymbol{z} + \Delta_{\boldsymbol{z}})$
virtual $\tilde{\boldsymbol{x}}$
on-manifold/semantic
data augmentation
latent $\boldsymbol{z}$
encoder $\mathcal{F}(\boldsymbol{x})$
sample $\boldsymbol{x}$
Adversarial Perturbations are adaptively computed by using
the feedback $\nabla\mathcal{L}(\mathcal{F}^{-1}(z))$ from a classifier loss $\mathcal{L}$

Figure 2. **Data augmentation via perturbation in the latent space**. Given a data sample $\boldsymbol{x}$, natural on-manifold data augmentations are generated by perturbing the encoded $\boldsymbol{z} = \mathcal{F}(\boldsymbol{x})$ in latent space, and decoding the perturbed $\boldsymbol{z} + \Delta_{\boldsymbol{z}}$. **Adversarial perturbations** require access to the loss function $\mathcal{L}$ to either find samples that are misclassified, or most difficult for the current model parameters.

ate $p_{\mathcal{X}}(\boldsymbol{x})$, (iv) are straightforward to train, and (v) they have useful latent representation—due to their immediate mapping from image to latent representation.

In summary, apart from the obvious benefit of fast encoding and decoding when performing latent-space perturbations, to guarantee that small latent-space perturbations will not modify the sample's label, the most prominent characteristic of normalizing flows is their *exact latent variable inference*. After presenting our method for perturbations in latent space and our experimental results, we further discuss the advantages of normalizing flows in §6.

4. Perturbations in Latent Space

The invertibility of normalizing flows enables bidirectional transitions between image and latent spaces, see above §3. This, in turn, allows for applying perturbations directly in the latent space rather than image space. We recall that we denote by $\mathcal{F} : \mathcal{X} \to \mathcal{Z}$ a trained normalizing flow, mapping from data manifold $\mathcal{X}$ to latent space $\mathcal{Z}$. Given a perturbation function $\mathcal{P} : \mathcal{Z} \to \mathcal{Z}$, defined over the latent space, we define its counterpart in image space as $\mathcal{F}^{-1}(\mathcal{P}(\mathcal{F}(\boldsymbol{x})))$.

Our goal is to define latent perturbation function $\mathcal{P}(\cdot)$ such that we obtain identity-preserving semantic modifications over the original image $\boldsymbol{x}$ in the image domain. To this end, we limit the structure of possible $\mathcal{P}$ in two ways. Firstly, we directly consider incremental perturbations of the form $\boldsymbol{z} + \mathcal{P}(\boldsymbol{z})$. Secondly, we use an extra ϵ parameter to control the size of perturbation allowed (see illustration in Figure 2). More precisely, we have:

$$\mathcal{F}^{-1}\big(\mathcal{F}(\boldsymbol{x}) + \mathcal{P}(\mathcal{F}(\boldsymbol{x}), \epsilon)\big) \,.$$

For brevity, we refer to $\mathcal{P}$ as *latent attacks* (LA), and we consider two variants described below.

4.1. Randomized Latent Attacks

At training time, given a datapoint $\boldsymbol{x}_i$, with $1 \le i \le N$, using the trained normalizing flow we obtain its corresponding latent code $\boldsymbol{z}_i = \mathcal{F}(\boldsymbol{x}_i)$.

Primarily, as perturbation function we consider a simplistic Gaussian noise in the latent space:

$$\mathcal{P}_{rand}(\cdot, \epsilon) = \epsilon \cdot \mathcal{N}(0, \mathbf{I}) \,, \tag{R–LA}$$

which is independent from $\boldsymbol{z}_i$. Any such distribution around the original $\boldsymbol{z}_i$ is equivalent to sampling from the learned manifold. In this case, the normalizing flow *pushes forward* this simple Gaussian distribution centered around $\boldsymbol{z}_i$ to a distribution on the image space around $\boldsymbol{x}_i = \mathcal{F}^{-1}(\boldsymbol{z}_i)$. Thus, sampling from the simple prior distribution $\mathcal{N}(0, \mathbf{I})$ is equivalent to sampling from a complex conditional distribution around the original image over the data manifold.

We also define norm truncated versions as follows:

$$\mathcal{P}^{\ell_p}_{rand}(\cdot, \epsilon) = \Pi(\epsilon \cdot \mathcal{N}(0, \mathbf{I})) \,,$$

where ℓ_p denotes the selected norm, *e.g.*, ℓ_2 or ℓ_∞. For ℓ_2 norm, Π is defined as ℓ_2 norm scaling, and for ℓ_∞, Π is the component-wise clipping operation defined below:

$$(\Pi(\boldsymbol{x}))_i := \max(-\epsilon, \min(+\epsilon, \boldsymbol{x}_i)) \,.$$

4.2. Adversarial Latent Attacks

Analogous to the above randomized latent attacks, at train time, given a datapoint $\boldsymbol{x}_i$ and it's associated label l_i, with $1 \le i \le N$, using the trained normalizing flow we obtain its corresponding latent code $\boldsymbol{z}_i = \mathcal{F}(\boldsymbol{x}_i)$.

We search for $\Delta_{\boldsymbol{z}_i} \in \mathcal{Z}$ such that the loss obtained of the generated image $\tilde{\boldsymbol{x}}_i = \mathcal{F}^{-1}(\boldsymbol{z}_i + \Delta_{\boldsymbol{z}_i})$ is maximal:

$$\begin{aligned} \Delta^{\star}_{\boldsymbol{z}_i} &= \mathop{\arg\max}_{\|\Delta_{\boldsymbol{z}_i}\|_{l_p} \le \epsilon} \mathcal{L}_\theta(\mathcal{F}^{-1}(\boldsymbol{z}_i + \Delta_{\boldsymbol{z}_i}), l_i) \,, \\ \mathcal{P}^{\ell_p}_{adv}(\boldsymbol{z}_i, \epsilon) &= \Delta^{\star}_{\boldsymbol{z}_i} \,, \end{aligned} \tag{A–LA}$$

where $\mathcal{L}_\theta$ is the loss function of the classifier, and ℓ_p denotes the selected norm, *e.g.*, ℓ_2 or ℓ_∞.

In practice, we define the number of steps k to optimize for $\Delta^{\star}_{\boldsymbol{z}_i} \in \mathcal{Z}$, as well as the step size α [similar to 53, 61], and we have the following procedure:

- Initialize a random $\Delta^0_{\boldsymbol{z}_i}$ with $\|\Delta^0_{\boldsymbol{z}_i}\|_{\ell_p} \le \epsilon$.

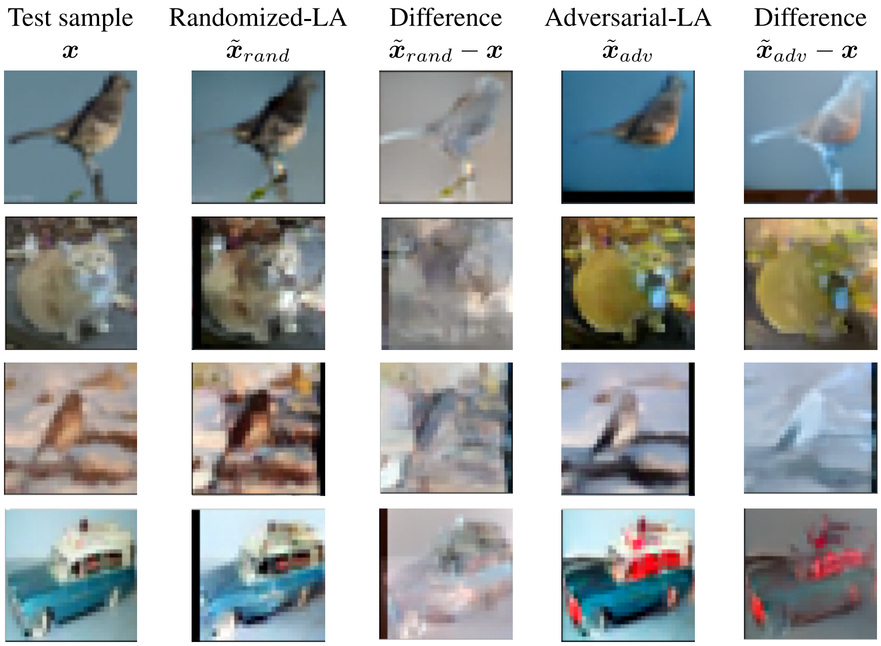
Test sample
$\boldsymbol{x}$
Randomized-LA
$\tilde{\boldsymbol{x}}_{rand}$
Difference
$\tilde{\boldsymbol{x}}_{rand} - \boldsymbol{x}$
Adversarial-LA
$\tilde{\boldsymbol{x}}_{adv}$
Difference
$\tilde{\boldsymbol{x}}_{adv} - \boldsymbol{x}$

Figure 3. **Illustrative results of our latent-space perturbations.** The models are trained on **CIFAR-10**. The first column depicts randomly selected samples from the test set. We depict the outputs obtained with Eq. R–LA and Eq. A–LA as well as their differences with the test samples. By observing the differences, we see that the added perturbations depend on the semantic content of the input image. See §5.4 for further discussion.

- Iteratively update $\Delta^j_{\boldsymbol{z}_i}$ for $j = 1, \ldots, k$ number of steps with step size α as follows:

$$\Delta^j_{\boldsymbol{z}_i} = \Pi\Big(\Delta^{j-1}_{\boldsymbol{z}_i} + \alpha \cdot \frac{\nabla \mathcal{L}_\theta(\mathcal{F}^{-1}(\boldsymbol{z}_i + \Delta^{j-1}_{\boldsymbol{z}_i}), l_i)}{\|\nabla \mathcal{L}_\theta(\mathcal{F}^{-1}(\boldsymbol{z}_i + \Delta^{j-1}_{\boldsymbol{z}_i}), l_i)\|_{\ell_p}}\Big)$$

  where Π is the projection operator that ensures condition $\|\Delta^j_{\boldsymbol{z}_i}\|_{\ell_p} \leq \epsilon$ and gradient is with respect to $\Delta^{j-1}_{\boldsymbol{z}_i}$.

- Output $\mathcal{P}_{adv}(\boldsymbol{z}_i, \epsilon) = \Delta^k_{\boldsymbol{z}_i}$

For the case of ℓ_∞, we replace normalization of gradient with $sign(\cdot)$ operator, *i.e.*:

$$\Delta^j_{\boldsymbol{z}_i} = \Pi\Big(\Delta^{j-1}_{\boldsymbol{z}_i} + \alpha \cdot sign\big(\nabla \mathcal{L}_\theta(\mathcal{F}^{-1}(\boldsymbol{z}_i + \Delta^{j-1}_{\boldsymbol{z}_i}), l_i)\big)\Big)$$

and use component-wise clipping for projection, which is equivalent to the standard ℓ_∞-PGD adversarial attack of Madry et al. [37].

Similarly, as the normalizing flow directly models the underlying data manifold, this perturbation is equivalent to a search over the *on-manifold* adversarial samples [53].

5. Experiments

Datasets. We evaluate our proposed semantic perturbations on the **FashionMNIST**, **SVHN**, **CIFAR-10**, and **CIFAR-100** datasets. See Appendix B for additional results on **MNIST**. For experiments on restricted datasets, *e.g.*, 5% of CIFAR-10, we always use the same sample set for a fair comparison.

Models. For FashionMNIST, we use a conditional 12-step normalizing flow based on *Glow* coupling blocks and a convolutional network of approximately $100K$ parameters, as in [53]. For experiments on SVHN and CIFAR-10/100, we use Glow [27] and ResNet-18 [17], respectively. See Appendix A for further details on the implementation.

Metrics. To evaluate the classifier's generalization, we use standard test accuracy. Adopting from the literature on GANs, we use Fréchet Inception Distance [FID, lower is better, see Appendix A.3, 20] to measure the similarity between the CIFAR-10 training data and samples produced by our latent perturbations.

Methods. We compare the following methods: (i) **standard**–classical training with no attacks, (ii) **Image-space PGD**: Projected Gradient Descent as an image-space, adversarial perturbation baseline [37], (iii) **VAE–GAN** [53]–on-manifold perturbation method that uses VAE-GANs, (iv) **Cutout** [9]–input masking, (v) **Mixup** [66]–data-agnostic data augmentation routine, (vi) **Randomized-LA (ours)**–randomized latent attacks using normalizing flow, as well as (vii) **Adversarial-LA (ours)**–adversarial latent attacks using normalizing flow, where *ours* are described in §4 and the rest of the methods in §2. For brevity, PGD, Randomized-LA and Adversarial-LA are sometimes denoted with $\mathcal{P}_{pgd}$, $\mathcal{P}_{rand}$ and $\mathcal{P}_{adv}$, respectively.

Method	Low-data	Full-set
Standard (no DA)	49.8	89.7
Standard + common DA	64.1	95.2
VAE-GAN [53]	58.9	94.2
Cutout [9]	66.8	96.0
Mixup [66]	73.4	95.9
Randomized-LA	70.1	96.3
Adversarial-LA	**80.4**	**96.6**

Table 1. Test accuracy (%) on **CIFAR-10**, in the *low-data regime* compared to the *full train set*. For the former, we use 5% and 100% of the training and test set, respectively. In addition to standard training, we consider standard training with commonly used data augmentations (DA) in the image space, which includes rotation and horizontal flips [65], as well as more recent *Cutout* [9] and *Mixup* [66] methods. See §5.1 for a discussion.

5.1. Generalization on CIFAR-10

We are primarily interested in the performance of our perturbations in the low-data regime when using only a small subset of CIFAR-10 as the training set. We train ResNet-18 classifiers on only 5% of the full training set and evaluate models on the full test set. We compare our methods with some of the most commonly used data augmentations methods such as *Cutout* [9] and *Mixup* [66], as well as with the VAE-GAN based approach [53]. For [53], we use the authors' implementation, and their default parameters for CelebA dataset, see Appendix A for details. For [9], we report the best test accuracy observed among a grid search on the learning rate $\eta \in \{0.1, 0.01\}$. Similarly, for [66], we report the best accuracy among grid search on learning rate $\eta \in \{0.1, 0.01\}$ and mixup coefficient $\lambda \in \{.1, .2, .3, .4, 1.0\}$. For Randomized-LA, we use $\ell = \ell_\infty, \epsilon = 0.25$, and for Adversarial-LA, we use $\ell = \ell_2, \epsilon = 1.0, \alpha = 0.5, k = 3$.

Table 1 summarizes our generalization experiments in the low data regime—using only 5% of CIFAR-10 for training, compared to the full CIFAR-10 training set. Figure 4 depicts the train and test accuracy throughout the training. Both Randomized-LA and Adversarial-LA notably outperform the standard training baseline. In particular, we observe that (i) our simplistic Randomized-LA method already outperforms some recent strong data augmentation methods, and (ii) Adversarial-LA achieves *best* test accuracy for both low-data and full-set regimes. See §5.3 below for additional benchmarks with VAE-GAN [53].

5.2. Transfer Learning Experiments

To further analyze potential applications of our normalizing flow based latent attacks to real-world use cases, we study if a normalizing flow pre-trained on a large dataset can be used for training classifiers on a different, smaller dataset. In particular, we use CIFAR-10 to train the normalizing flows and then our latent attacks to train a classifier on 10% and 5% of the CIFAR-100 and SVHN training datasets, respectively.

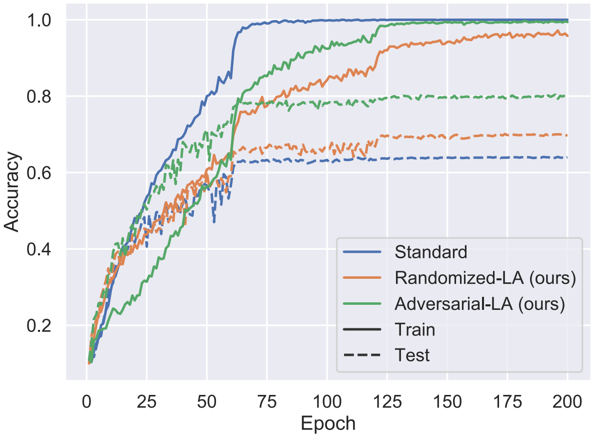

Figure 4. Accuracy when training on 5% of the **CIFAR-10** dataset, and testing on its full test set. See §5.1 for a discussion.

Perturbation	Accuracy
Standard	36.4
Randomized-LA, $\ell{=}\ell_\infty, \epsilon{=}.2$	39.7
Randomized-LA, $\ell{=}\ell_\infty, \epsilon = .3$	41.0
Randomized-LA, $\ell{=}\ell_2, \epsilon = 10$	40.4
Randomized-LA, $\ell{=}\ell_2, \epsilon = 20$	**42.3**
Adversarial-LA, $\ell{=}\ell_2, \alpha{=}.5, k{=}3$	**45.0**

Table 2. Test accuracy (%) on **CIFAR-100**, in the *low-data regime*, where we use 10% of the training set and the full test set. The normalizing flow used is trained on **CIFAR-10**.

Table 2 shows our results for CIFAR-100 using a selection of latent attacks. Randomized-LA and Adversarial-LA achieve 16% and 24% improvements over the standard baseline. The results indicate that normalizing flows are capable of transferring useful augmentations learned from CIFAR-10 to CIFAR-100.

Table 3 shows our results for SVHN. To provide a baseline on the effect of using different datasets for normalizing flows and classifiers, we also provide results with pre-training on SVHN. Latent attacks transferred from CIFAR-10 achieve superior performance to direct pre-training on SVHN, indicating that transferring augmentations across datasets is indeed a promising direction.

5.3. Additional Comparison with VAE-GAN

Following Stutz et al. [53], we study the performance of our latent perturbation-based training strategies in varying settings, starting from low-data regime to full-set. For the VAE-GAN results, we use the source code provided by the authors, while using their default hyperparameters for the same dataset. For our methods, we reproduce the same

NF	Perturbation	Accuracy
–	Standard	81.2
SVHN	$\mathcal{P}^{\ell_2}_{rand}$, ϵ=15. $\mathcal{P}^{\ell_2}_{adv}$, ϵ=.5, α=.25, k=2	84.9 86.9
CIFAR-10	$\mathcal{P}^{\ell_2}_{rand}$, ϵ=15. $\mathcal{P}^{\ell_2}_{adv}$, ϵ=.3, α=.15, k=2	**90.0** **90.5**

Table 3. Test accuracy (%) on **SVHN**, in the *low-data regime*, where we use 5% of the training set and the full test set. Comparison of normalizing flows trained on **CIFAR-10**, versus **SVHN**.

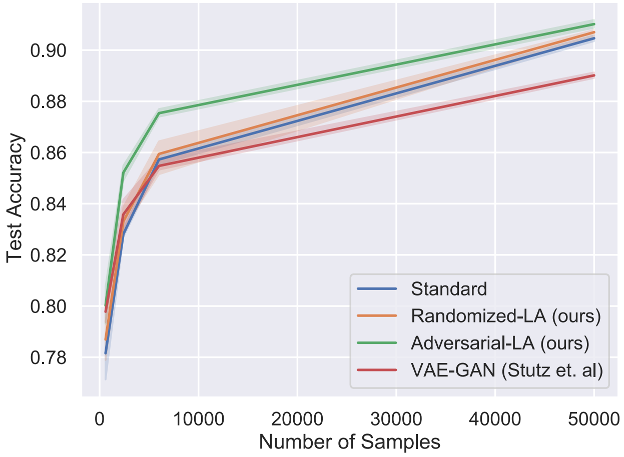

Figure 5. Test accuracy (%) on full test set, for a varying number of *training* samples, on the **FashionMNIST** dataset. To replicate the setup of VAE-GAN [53], only a portion of the dataset (x-axis) is used to train the classifier, while the corresponding generative model is trained on the full dataset. We run each experiment with three different random seeds and report the mean and standard deviation of the test accuracy. See §5.3.

classifier and hyperparameter setup. For Randomized-LA, we use $\ell = \ell_\infty, \epsilon = 0.15$, and for Adversarial-LA, we use $\ell = \ell_\infty, \epsilon = 0.05, \alpha = 0.01, k = 10$.

Figure 5 shows our average results for 3 runs with training sizes in $\{600, 2400, 6000, 50000\}$. We observe that Randomized-LA performs comparatively to the standard training baseline, whereas Adversarial-LA outperforms the standard baseline across all train set sizes. Note that the difference to the standard baselines shrinks as we increase the number of samples available to the classifiers.

In line with our results, Stutz et al. [53] report diminishing performance gains for increasingly challenging datasets such as FashionMNIST to CelebA, when using therein VAE-GAN based approach. One potential cause could be the *approximate* encoding and decoding mappings or sensitivity to hyperparameter tuning. Relative to VAE-GAN, normalizing flows have significantly fewer hyperparameters, see Appendix A.2. Indeed, our results support the numerous appealing advantages of normalizing flows for latent-space perturbations and indicate that they have a better capacity to produce useful augmented training samples.

Model or Perturbation	FID
Baseline: GANs	
DCGAN [20]	36.9
WGAN-GP [20]	24.8
BigGAN [5]	14.73
StyleGAN [23]	2.92
Baseline: image-space	
PGD [37], ℓ=ℓ_∞, ϵ=.03, α=.008, k=10	23.61
Ours: latent-space	
Randomized-LA, ℓ=ℓ_∞, ϵ=.25	3.71
Adversarial-LA, ℓ=ℓ_2, ϵ=1., α=.5, k=3	3.65

Table 4. FID scores (lower is better) of generated samples of GANs, image-space PGD perturbations, and our Randomized-LA and Adversarial-LA methods. For PGD and Adversarial-LA perturbations which use a classifier, we use the same standardly trained ResNet-18. See §5.4.

Perturbation	ℓ_2 in $\mathcal{X}$	ℓ_∞ in $\mathcal{X}$
Baseline: image-space		
$\mathcal{P}^{\ell_\infty}_{pgd}$, $\epsilon = .03, \alpha = .008, k = 10$	1.13	0.03
$\mathcal{P}^{\ell_2}_{pgd}$, $\epsilon = 2., \alpha = .5, k = 10$	1.98	0.15
Ours: latent-space		
$\mathcal{P}^{\ell_\infty}_{rand}$, $\epsilon = .25$	4.18	0.41
$\mathcal{P}^{\ell_2}_{adv}$, $\epsilon = 1., \alpha = .5, k = 3$	4.61	0.44

Table 5. Average ℓ_2 and ℓ_∞ size of perturbations computed in image space $\mathcal{X}$ using CIFAR-10 test samples. For PGD and Adversarial-LA perturbations which use a classifier, we use the same standardly trained ResNet-18.

5.4. Analysis of Generated Images

Figure 3 depicts samples of our Randomized-LA and Adversarial-LA methods. Primarily, in contrast to random image-space perturbations, we observe that both Randomized-LA and Adversarial-LA yield perturbations dependent on the semantic content of the input image. Interestingly, one could argue that Adversarial-LA further masks potential *shortcuts* that the classifier may learn, for *e.g.*, by masking the windows, it forces the classifier to, in fact, learn the shape of a car. Moreover, we observe that relative to image-space perturbations, latent attacks produce samples that are semantically closer to the CIFAR-10 training set—see Table 4 for FID scores, and at the same time, more distinct in image space—see Table 5.

5.5. Robustness against Latent Attacks

In Table 6, we evaluate the robustness of classifiers against our latent attacks and observe that both standard

Attack	Trained Perturbation	Acc.	Drop
$\mathcal{P}^{\ell_\infty}_{rand}$	Standard	90.5	4.8
	$\mathcal{P}^{\ell_\infty}_{pgd}, \epsilon = .03, \alpha = .008, k = 10$	76.9	9.4
	$\mathcal{P}^{\ell_\infty}_{rand}, \epsilon = .25$	94.1	2.1
	$\mathcal{P}^{\ell_2}_{adv}, \epsilon = 1., \alpha = .5, k = 3$	94.6	2.0
$\mathcal{P}^{\ell_2}_{adv}$	Standard	58.8	38.2
	$\mathcal{P}^{\ell_\infty}_{pgd}, \epsilon = .03, \alpha = .008, k = 10$	36.2	57.3
	$\mathcal{P}^{\ell_\infty}_{rand}, \epsilon = .25$	71.2	25.9
	$\mathcal{P}^{\ell_2}_{adv}, \epsilon = 1., \alpha = .5, k = 3$	76.4	20.8

Table 6. Robustness against our perturbations $\mathcal{P}^{\ell_\infty}_{rand}, \epsilon = .25$ and $P^{\ell_2}_{adv}, \epsilon = 1., \alpha = .5, k = 3$ on the **CIFAR-10** dataset. **Trained Perturbation**: the *training-time* perturbation used to train the model; **Drop**: the drop in test accuracy with latent perturbations *relative* to the accuracy on CIFAR-10 test samples.

and image-space adversarial training suffer from a significant loss of performance against Adversarial-LA. Combined with observations from §5.4, this indicates that our adversarial latent attack is a novel approach to generate *realistic* adversarial samples. Interestingly, classifiers trained with image-space adversarial perturbations are more prone to large accuracy drops than standardly trained classifiers. Additionally, although the classifiers trained with our perturbations are robust to Randomized-LA, they are not fully robust to Adversarial-LA, suggesting the possibility of further improving generalization using latent attacks.

6. Discussion

Exact Coding. As formalized in §3, normalizing flows can perform exact encoding and decoding by their construction. That is, the decoding operation is exactly the reverse of the encoding operation. Any continuous encoder maps a neighborhood of a sample to some neighborhood of its latent representation. However, the invertibility of normalizing flows also maps *any neighborhood of latent code to a neighborhood of the original sample*. In principle, this property also holds for off-manifold samples and may explain the effectiveness of our methods in transferring augmentations.

Increasing Dataset Size. The primary advantage of exact coding is that the generated samples via latent perturbations improve the generalization of classifiers, as shown in §5.1. To understand why this occurs, consider the limit case $\epsilon \to 0$ for a latent perturbation. Assuming a numerically stable normalizing flow, we recover the original data samples, hence the training distribution. As we increase ϵ, this distribution grows around each data point. Thus, by increasing ϵ, we add further plausible data points to our training set, as long the learned latent representation is a good approximation of the underlying data manifold. This does not necessarily hold for approximate methods due to inherent *decoder noise*.

Controllability. In §4, we introduced two variants of latent perturbations that define different procedures around the latent code of the original sample. Each variant employs a normalizing flow to efficiently map a complex on-manifold objective to a local objective in the latent space. The randomized latent attack defines a sampling operation on the data manifold, and the adversarial latent attack, a stochastic search procedure to find on-manifold samples attaining high classifier losses. In principle, any other on-manifold objectives may also utilize such mappings to the latent space and, potentially, use the density provided by the normalizing flow to enforce strict checks for on-manifold data points. Moreover, conditional normalizing flows may achieve more expressive, class-specific augmentations and control mechanisms.

Compatibility with Data Augmentations. It is important to note that our method is orthogonal to image-space data augmentation methods. In other words, we can train normalizing flows with commonly used data augmentations. As observed in Figure 3, trained models can apply some of the training-time augmentations to CIFAR-10 test samples. This allows us to encode and decode *augmented* samples as well as original samples of CIFAR-10. Additionally, we can use DeVries and Taylor [9], Zhang et al. [66] concurrently with our latent perturbations to train classifiers.

7. Conclusion

Motivated by the numerous advantages of normalizing flows, we propose flow-based latent perturbation methods to augment the training datasets to train classifiers. Our extensive empirical results on several real-world datasets demonstrate the efficacy of these perturbations for improving generalization both in full and low-data regimes. In particular, these perturbations can increase sample efficiency in low-data regimes and, in practice, reduce labeling efforts.

Further directions include (i) decoupling the effects of exact coding from any modeling gains through ablation studies, as well as (ii) combining image and latent-space augmentations.

Acknowledgments

TC was funded in part by the grant P2ELP2_199740 from the Swiss National Science Foundation. The authors would like to thank Maksym Andriushchenko and Suzan Üsküdarlı for insightful feedback and discussions.

References

[1] Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017. 2

[2] Lynton Ardizzone, Carsten Lüth, Jakob Kruse, Carsten Rother, and Ullrich Köthe. Guided image generation with conditional invertible neural networks. *arXiv preprint arXiv:1907.02392*, 2019. 12

[3] Shumeet Baluja and Ian Fischer. Adversarial transformation networks: Learning to generate adversarial examples. *arXiv preprint arXiv:1703.09387*, 2017. 2

[4] Battista Biggio and Fabio Roli. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84:317–331, 2018. 1

[5] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 7

[6] Tatjana Chavdarova, Matteo Pagliardini, Sebastian U Stich, François Fleuret, and Martin Jaggi. Taming GANs with Lookahead-Minmax. In *International Conference on Learning Representations*, 2021. 3

[7] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 113–123, 2019. 2

[8] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020. 2

[9] Terrance DeVries and Graham W. Taylor. Improved Regularization of Convolutional Neural Networks with Cutout. *arXiv:1708.04552*, 2017. arXiv: 1708.04552. 1, 2, 5, 6, 8, 12, 13

[10] Laurent Dinh, David Krueger, and Yoshua Bengio. NICE: non-linear independent components estimation. In Yoshua Bengio and Yann LeCun, editors, *International Conference on Learning Representations, ICLR*, 2015. 3

[11] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *International Conference on Learning Representations, ICLR*, 2017. 3

[12] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using Real NVP. In *International Conference on Learning Representations (ICLR)*, 2017. 12

[13] Hadi M Dolatabadi, Sarah Erfani, and Christopher Leckie. Advflow: Inconspicuous black-box adversarial attacks using normalizing flows. *arXiv preprint arXiv:2007.07435*, 2020. 2

[14] A. Fawzi, H. Samulowitz, D. Turaga, and P. Frossard. Adaptive data augmentation for image classification. In *IEEE International Conference on Image Processing (ICIP)*, 2016. 2

[15] Justin Gilmer, Luke Metz, Fartash Faghri, Samuel S. Schoenholz, Maithra Raghu, Martin Wattenberg, and Ian Goodfellow. Adversarial spheres. *arXiv preprint arXiv:1801.02774*, 2018. 2

[16] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 27, pages 2672–2680, 2014. 1, 3

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 5

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 12

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016. 12

[20] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 6626–6637, 2017. 5, 7, 13

[21] Jonathan Ho, Xi Chen, Aravind Srinivas, Yan Duan, and Pieter Abbeel. Flow++: Improving flow-based generative models with variational dequantization and architecture design. In *Proceedings of the 36th International Conference on Machine Learning*, pages 2722–2730, 2019. 3

[22] Ajil Jalal, Andrew Ilyas, Constantinos Daskalakis, and Alexandros G. Dimakis. The robust manifold defense: Adversarial training using generative models. *arXiv preprint arXiv:1712.09196*, 2017. 2

[23] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *arXiv preprint arXiv:2006.06676*, 2020. 7

[24] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8110–8119, 2020. 2

[25] Yannic Kilcher, Aurélien Lucchi, and Thomas Hofmann. Generator reversal, 2017. 3

[26] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 12

[27] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in neural information processing systems*, pages 10215–10224, 2018. 1, 2, 3, 5, 12

[28] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014. 1, 3

[29] Klim Kireev, Maksym Andriushchenko, and Nicolas Flammarion. On the effectiveness of adversarial training against common corruptions. *arXiv preprint arXiv:2103.02325*, 2021. 2

[30] Alex Krizhevsky and Geoffrey Hinton. Learning Multiple Layers of Features from Tiny Images. page 60, 2009. 12

[31] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. 2

[32] Cassidy Laidlaw, Sahil Singla, and Soheil Feizi. Perceptual adversarial robustness: Defense against unseen threat mod-

els. In *International Conference on Learning Representations (ICLR)*, 2021. 2

[33] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In *ICML*, 2016. 1, 3

[34] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov. 1998. Conference Name: Proceedings of the IEEE. 12

[35] 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. 12

[36] Calvin Luo, Hossein Mobahi, and Samy Bengio. Data augmentation via structured adversarial perturbations. *arXiv preprint arXiv:2011.03010*, 2020. 2

[37] 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 (ICLR)*, 2018. 5, 7

[38] Shashank Manjunath, Aitzaz Nathaniel, Jeff Druce, and Stan German. Improving the performance of fine-grain image classifiers via generative data augmentation. *arXiv preprint arXiv:2008.05381*, 2020. 2

[39] Agnieszka Mikołajczyk and Michał Grochowski. Style transfer-based image synthesis as an efficient regularization technique in deep learning. In *2019 24th International Conference on Methods and Models in Automation and Robotics (MMAR)*, pages 42–47. IEEE, 2019. 2

[40] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, Feb. 2015. 1

[41] Y. E. Nesterov. A method for solving the convex programming problem with convergence rate O(1/kˆ2). *Dokl. Akad. Nauk SSSR*, 269:543–547, 1983. 13

[42] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. 12

[43] Xi Peng, Zhiqiang Tang, Fei Yang, Rogerio S. Feris, and Dimitris Metaxas. Jointly optimize data augmentation and network training: Adversarial data augmentation in human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[44] Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017. 2

[45] Alexander J Ratner, Henry Ehrenberg, Zeshan Hussain, Jared Dunnmon, and Christopher Ré. Learning to compose domain-specific transformations for data augmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30. Curran Associates, Inc., 2017. 2

[46] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951. 13

[47] Alexander Robey, Hamed Hassani, and George J. Pappas. Model-based robust deep learning: Generalizing to natural, out-of-distribution data. *arXiv preprint arXiv:2005.10247*, 2020. 2

[48] Mihaela Rosca, Balaji Lakshminarayanan, David Warde-Farley, and Shakir Mohamed. Variational approaches for auto-encoding generative adversarial networks, 2017. 1, 3

[49] Andras Rozsa, Manuel Günther, and Terrance E. Boult. Are accuracy and robustness correlated. In *15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 227–232. IEEE Computer Society, 2016. 2

[50] Patrice Y. Simard, Yann A. LeCun, John S. Denker, and Bernard Victorri. *Transformation Invariance in Pattern Recognition — Tangent Distance and Tangent Propagation*, pages 239–274. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998. 2

[51] Yang Song, Rui Shu, Nate Kushman, and Stefano Ermon. Constructing unrestricted adversarial examples with generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31. Curran Associates, Inc., 2018. 2

[52] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research (JMLR)*, 15(1):1929–1958, 2014. 1

[53] 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 (CVPR)*, pages 6976–6987, 2019. 1, 2, 4, 5, 6, 7, 12, 13

[54] Dong Su, Huan Zhang, Hongge Chen, Jinfeng Yi, Pin-Yu Chen, and Yupeng Gao. Is robustness the cost of accuracy? - A comprehensive study on the robustness of 18 deep image classification models. In *ECCV*, 2018. 1

[55] Thomas Tanay and Lewis Griffin. A boundary tilting persepective on the phenomenon of adversarial examples. *arXiv preprint arXiv:1608.07690*, 2016. 2

[56] Zhiqiang Tang, Yunhe Gao, Leonid Karlinsky, Prasanna Sattigeri, Rogerio Feris, and Dimitris Metaxas. Onlineaugment: Online data augmentation with less domain knowledge. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*, pages 313–329. Springer, 2020. 2

[57] Ngoc-Trung Tran, Viet-Hung Tran, Ngoc-Bao Nguyen, Trung-Kien Nguyen, and Ngai-Man Cheung. On data augmentation for GAN training. *arXiv preprint arXiv:2006.05338*, 2020. 2

[58] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2019. 1

[59] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations (ICLR)*, 2019. 2

[60] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C. Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. In *Ad-*

vances in neural information processing systems (NeurIPS), pages 5334–5344, 2018. 2

[61] Eric Wong and J Zico Kolter. Learning perturbation sets for robust machine learning. In *International Conference on Learning Representations (ICLR)*, 2021. 2, 4

[62] Chaowei Xiao, Bo Li, Jun yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating adversarial examples with adversarial networks. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3905–3911, 7 2018. 2

[63] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017. 12

[64] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 6023–6032, 2019. 2

[65] Sergey Zagoruyko and Nikos Komodakis. Wide Residual Networks. In *Procedings of the British Machine Vision Conference 2016*, pages 87.1–87.12, York, UK, 2016. British Machine Vision Association. 6, 13

[66] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond Empirical Risk Minimization. In *International Conference on Learning Representations (ICLR)*, 2018. 1, 2, 5, 6, 8, 12, 13

[67] Linfeng Zhang, Muzhou Yu, Tong Chen, Zuoqiang Shi, Chenglong Bao, and Kaisheng Ma. Auxiliary training: Towards accurate and robust models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[68] Ruixiang Zhang, Tong Che, Zoubin Ghahramani, Yoshua Bengio, and Yangqiu Song. Metagan: An adversarial approach to few-shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31. Curran Associates, Inc., 2018. 2

[69] Xinyu Zhang, Qiang Wang, Jian Zhang, and Zhao Zhong. Adversarial autoaugment. In *International Conference on Learning Representations (ICLR)*, 2020. 2

[70] Zhengli Zhao, Dheeru Dua, and Sameer Singh. Generating natural adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2018. 2