ROBUSTNESS VIA LEARNED BREGMAN DIVERGENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

We exploit the Bregman divergence to generate functions that are trained to measure the semantic similarity between images under corruptions and use these functions as alternatives to the L^p norms to define robustness threat models. Then we replace the projected gradient descent (PGD) by semantic attacks, which are instantiations of the mirror descent, the optimization framework associated with the Bregman divergence. Adversarial training under these settings yield classification models that are more robust to common image corruptions. Particularly, for the contrast corruption that was found problematic in prior work we achieve an accuracy that exceeds the L^p- and the LPIPS-based adversarially trained neural networks by a margin of 29% on the CIFAR-10-C corruption dataset.

1 INTRODUCTION

Neural networks for image classification are sensitive to input variations the way the human vision system is not. This means they can misclassify when images are subjected to either small maliciously-crafted perturbations (so-called adversarial examples) (Biggio et al., 2013; Szegedy et al., 2014; Papernot et al., 2016), or to the more realistic distribution shifts associated with common, realistic image corruptions like blur or contrast changes (Dodge & Karam, 2017; Hendrycks & Dietterich, 2019). Thus both adversarial robustness and corruption robustness are active research topics.

The common approach to achieve corruption robustness is the augmentation of the training set with synthetically modified images, e.g., mixing images by blending, cutting and pasting parts, fusion through spectral analysis, and others (Zhang et al., 2018a; Yun et al., 2019; Harris et al., 2021; Walawalkar et al., 2020; Park et al., 2022; Yin et al., 2022; Liang et al., 2023).

The most successful method for achieving adversarial robustness is adversarial training (AT) (Madry et al., 2018) and its many follow-up variants, e.g., (Uesato et al., 2019; Zhang et al., 2019; Carmon et al., 2019; Wu et al., 2020; Chen et al., 2021; Rebuffi et al., 2021; Jiang et al., 2023). It consists of first performing an attack to find adversarial examples images within an ϵ-ball w.r.t. an L^p norm, and then including these in the training set. The basic machinery for most white-box attacks is projected gradient descent (PGD) (Madry et al., 2018; Goodfellow et al., 2015; Wong et al., 2020; Croce & Hein, 2020). Even though L^p norms carry no semantic meaning (they are oblivious to the nature of images relevant to humans), AT was found to also improve corruption robustness when done with carefully chosen hyperparameters (Hendrycks & Dietterich, 2019; Ford et al., 2019; Xie et al., 2020; Kang et al., 2019; Kireev et al., 2022). Conversely, Ford et al. (2019) proved that adversarial examples exist due to a nonzero test error under random noise, a particular distribution shift. Thus, the hope is to further improve corruption robustness by using the AT machinery with more meaningful similarity measures. A prominent example is the work by Kireev et al. (2022); Laidlaw et al. (2021); Wang et al. (2021). It uses the so-called learned perceptual image patch similarity (LPIPS) (Zhang et al., 2018b), which computes the L^2 form over the features extracted by a convolutional neural net, and ports PGD accordingly.

In this work, we take a different route by first learning a *Bregman divergence* (Bregman, 1967) for image corruptions and then using the associated *mirror descent* optimization framework (Nemirovskij & Yudin, 1983), which generalizes PGD, for AT.

Bregman divergence and mirror descent. Bregman divergence (Bregman, 1967) is a generalization of the Kullback–Leibler divergence (Kullback & Leibler, 1951), and is widely used in statistics and information theory to define distances in spaces where the Euclidean geometry is not appropriate

such as probability distributions, covariance descriptors, random processes and others (Chowdhury et al., 2023; Csiszar & Matus, 2008; Bauschke & Borwein, 1997; Stummer & Vajda, 2009; Frigyik et al., 2008; Harandi et al., 2014). Once defined, the associated mirror descent (Nemirovskij & Yudin, 1983), which generalizes PGD, allows solving constrained optimization problems including, as we will show, the kind needed in adversarial attacks and training.

Contributions. In this paper we offer progress in the quest for corruption robustness through a theoretically principled approach that uses a learned Bregman divergence with a suitably designed AT. A Bregman divergence is defined by a so-called base function that is convex and has an invertible gradient. As example, the KL divergence is defined by the Shannon entropy.

For a given corruption type we show how to learn an eligible base function as a neural net using a proposed self-supervised algorithm. The associated Bregman divergence is semantic in that it assesses corrupted images as close and randomly perturbed ones as far from the clean image, even if in Euclidean distance it is clearly the opposite.

We instantiate mirror descent, which generalizes PGD, to perform semantic adversarial attacks using the learned divergence instead of using an L^p norm.

We adopt this attack for AT and show that it yields classification models with high corruption robustness on the CIFAR-10-C for the contrast and fog corruptions that are known to be problematic (e.g., Ford et al. (2019) and Kireev et al. (2022)).

2 BACKGROUND

We first recall standard adversarial training (AT) with projected gradient descent (PGD). Then we provide background on Bregman divergence (Bregman, 1967) and the associated mirror descent framework, which generalizes PGD (Nemirovskij & Yudin, 1983). Our work will then port AT with PGD using an L^p norm to one with mirror descent and a learned Bregman divergence as similarity measure.

Adversarial training. Let $l(\boldsymbol{x}, y; \theta)$ be a loss of a classifier parameterized by θ where the input image $\boldsymbol{x}$ and the label y are sampled from the data distribution $\mathcal{D}$. As formalized by Madry et al. (2018), training an adversarially robust model amounts to solving the following min-max optimization problem:

$$\min_{\theta} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}} \left[\max_{\boldsymbol{x}' \in \mathbb{S}(\boldsymbol{x})} l(\boldsymbol{x}', y; \theta) \right] \tag{1}$$

where $\mathbb{S}(\boldsymbol{x})$ is the set of images that are considered similar to $\boldsymbol{x}$. Under the common L^p threat model, $\mathbb{S}(\boldsymbol{x})$ is defined as an L^p ball centered on $\boldsymbol{x}$ of chosen radius ϵ: $\mathbb{S}(\boldsymbol{x}) = \mathbb{B}(\boldsymbol{x}, \epsilon)$[1]. In this case, the inner maximization problem is solved by PGD, which consists of iterating over two steps: a gradient-based update followed by a projection into $\mathbb{B}(\boldsymbol{x}, \epsilon)$.

Bregman divergence. For a strictly convex function $h : \mathcal{X} \to \mathbb{R}$ on a given space $\mathcal{X}$ (called the primal space) with (thus strictly monotonous) gradient $\nabla h : \mathcal{X} \to \mathcal{Z}$ ($\mathcal{Z}$ is called the dual space), the Bregman divergence (Bregman, 1967) $D_h : \mathcal{X} \times \mathcal{X} \to [0, \infty)$ from $\boldsymbol{x}$ to $\boldsymbol{x}'$ with respect to h is defined as

$$D_h(\boldsymbol{x}' \parallel \boldsymbol{x}) = h(\boldsymbol{x}') - h(\boldsymbol{x}) - \langle \nabla h(\boldsymbol{x}), \boldsymbol{x}' - \boldsymbol{x} \rangle \tag{2}$$

The Bregman divergence is similar to a metric or distance (nonnegative, zero iff $\boldsymbol{x} = \boldsymbol{x}'$), except that in general it is not symmetric in its arguments and only satisfies a weaker version of the triangle inequality (whose exact form is not relevant here). D_h is convex in its first argument but not necessarily in the second (Edelsbrunner & Wagner, 2017). The projection of an $\boldsymbol{x} \in \mathcal{X}$ on a closed and convex set $\mathbb{K} \subseteq \mathcal{X}$ w.r.t. to D_h exists and is unique:

$$\Pi_{\mathbb{K}}(\boldsymbol{x}) = \min_{\boldsymbol{x}' \in \mathbb{K}} D_h(\boldsymbol{x}' \parallel \boldsymbol{x}). \tag{3}$$

The generic concepts are shown in the first column in Table 1; the other columns are examples. The squared Euclidean distance is a Bregman divergence if h is chosen as the squared L^2 norm. More in-

[1] All threat models add another condition to ensure that the adversarial example $\boldsymbol{x}'$ does not exceed its natural range of pixels.

Table 1: Notation and context of our approach. The first column shows the generic concepts associated with the Bregman divergence and mirror descent. The second and third columns are known instantiations. The last column is our contribution and basis for a novel approach to robustness.

Generic	Euclidean norm	KL divergence	Ours
Some space $\mathcal{X}$	Euclidean space	Discrete distributions	Images
Base function $h : \mathcal{X} \to \mathbb{R}$ *(strictly convex)*	$h(\boldsymbol{x}) = \frac{1}{2}\|\|\boldsymbol{x}\|\|_2^2$	$h(\boldsymbol{p}) = \sum_i \boldsymbol{p}_i \log(\boldsymbol{p}_i)$ *(Shannon entropy)*	$h =$ learned ϕ (an ICNN)
Mirror map $\nabla h : \mathcal{X} \to \mathcal{Z}$ *(strictly monotone)*	$\nabla h(\boldsymbol{x}) = \boldsymbol{x}$	$\nabla h(\boldsymbol{p})_i = \log(\boldsymbol{p}_i)$	$\Psi \approx \nabla h$ (approximate gradient)
Inverse map $(\nabla h)^{-1} : \mathcal{Z} \to \mathcal{X}$	$(\nabla h)^{-1}(\boldsymbol{z}) = \boldsymbol{z}$	$(\nabla h)^{-1}(\boldsymbol{z})_i = e^{\boldsymbol{z}_i}$	Fenchel conjugate $\overline{\Psi}$
Bregman Divergence $D_h(\boldsymbol{x}' \parallel \boldsymbol{x})$	$\frac{1}{2}\|\|\boldsymbol{x}' - \boldsymbol{x}\|\|_2^2$	$\sum_i \boldsymbol{q}_i \log \frac{\boldsymbol{q}_i}{\boldsymbol{p}_i}$	D_ϕ (learned divergence)
Mirror descent	PGD	Hedge algorithm	Ours
$\boldsymbol{z}^t = \nabla h(\boldsymbol{x}^t)$ $\boldsymbol{z}^{t+1} = \boldsymbol{z}^t - \eta \nabla f(\boldsymbol{x}^t)$ $\boldsymbol{x}^* = (\nabla h)^{-1}\big(\boldsymbol{z}^{t+1}\big)$ $\boldsymbol{x}^{t+1} = \Pi_{\mathbb{K}}(\boldsymbol{x}^*)$	$\boldsymbol{x}^* = \boldsymbol{x}^t - \eta \nabla f(\boldsymbol{x}^t)$ $\boldsymbol{x}^{t+1} = \Pi_{\mathbb{B}}(\boldsymbol{x}^*)$	$\boldsymbol{p}_i^* = \boldsymbol{p}_i^t e^{-\eta \boldsymbol{l}_i}$ $\boldsymbol{p}^{t+1} = \Pi_{\Delta}(\boldsymbol{p}^*)$	$\boldsymbol{z}^t = \Psi(\boldsymbol{x}^t)$ $\boldsymbol{z}^{t+1} = \boldsymbol{z}^t + \eta \nabla l(\boldsymbol{x}^t)$ $\boldsymbol{x}^* = \overline{\Psi}\big(\boldsymbol{z}^{t+1}\big)$ $\boldsymbol{x}^{t+1} = \Pi_{\mathbb{S}}(\boldsymbol{x}^*)$

terestingly, if h is the negative Shannon entropy, the associated Bregman divergence is the Kullback-Leibler (KL) divergence. Other examples of Bregman divergence include the Itakura–Saito distance, LeCam divergence, Brug divergence, Jeffreys Divergence, or Stein Divergence (Bauschke & Borwein, 1997; Stummer & Vajda, 2009; Harandi et al., 2014). The Bregman divergence is used if no suitable choice of a metric is available.

L^p balls generalize to the Bregman divergence: the *Bregman ball* centered on $\boldsymbol{x}$ with radius ϵ is given by

$$\mathbb{B}_h(\boldsymbol{x}, \epsilon) = \big\{\boldsymbol{x}' \in \mathcal{X} \mid D_h(\boldsymbol{x}' \parallel \boldsymbol{x}) \leq \epsilon\big\}. \tag{4}$$

$\mathbb{B}_h$ is bounded and compact if $\mathcal{X}$ is closed but not necessarily convex (Edelsbrunner & Wagner, 2017).

Mirror descent. Mirror descent (Nemirovskij & Yudin, 1983) is a framework for optimizing functions $f : \mathcal{X} \to \mathbb{R}$ possibly constrained to a feasible convex set $\mathbb{K}$, $\min_{\boldsymbol{x} \in \mathbb{K}} f(\boldsymbol{x})$ given a suitable base function h that defines a Bregman divergence. Mirror descent requires the gradient ∇h (called the *mirror map*) and the existence of $(\nabla h)^{-1}$ (called the *the inverse map*). The algorithm is iterative as shown in the first column in Table 1. After initializing $\boldsymbol{x}^0$ at any point in $\mathbb{K}$, each iteration t consists of four steps: *(i)* mapping the current point $\boldsymbol{x}^t$ to a point in the dual space $\boldsymbol{z}^t = \nabla h(\boldsymbol{x}^t)$ through the mirror map, *(ii)* taking a gradient step of size η: $\boldsymbol{z}^{t+1} = \boldsymbol{z}^t - \eta \nabla f(\boldsymbol{x}^t)$, *(iii)* mapping $\boldsymbol{z}^{t+1}$ back to the primal space using the inverse map: $\boldsymbol{x}^* = (\nabla h)^{-1}\big(\boldsymbol{z}^{t+1}\big)$, *(iv)* projecting $\boldsymbol{x}^*$ into the feasible set $\mathbb{K}$ w.r.t. D_h: $\boldsymbol{x}^{t+1} = \Pi_{\mathbb{K}}(\boldsymbol{x}^*)$ with (3).

As shown in Table 1, for the Euclidean divergence, mirror descent is exactly PGD. For the KL divergence it becomes the so-called hedge algorithm (Freund & Schapire, 1997). In this paper, as sketched in the fourth column, we will learn base functions h that we call ϕ and associated divergences for common image corruptions and use them for AT as explained next.

3 Generating a Bregman divergence for images

Motivated by the KL divergence in information theory we exploit the theory of Bregman divergence to derive new tools for robustness that match the geometry of image corruptions. Our high-level approach is outlined in the fourth column of Table 1. For a given type of corruption, we learn a base function $h = \phi$ that satisfies the properties to make D_ϕ a divergence. Mathematically, this ϕ will

play the same role as the Shannon entropy for KL divergence. We then instantiate mirror descent to solve the inner maximization problem in (1) and thus enable AT with D_ϕ. Formally, the challenge is to learn a ϕ with the following properties:

(i) ϕ is convex and differentiable, and thus D_ϕ a divergence;

(ii) $\nabla\phi(\boldsymbol{x})$ and $(\nabla\phi)^{-1}(\boldsymbol{x})$ have to be (approximately) computable to execute mirror descent.

Hand-engineering an eligible and performant ϕ (e.g., by stacking feature extractors) is likely to be a daunting task. We propose to model ϕ as a deep neural network with a particular architecture: the *input convex neural network (ICNN)* (Amos et al., 2017) for which we design a self-supervised learning algorithm. The details are explained next.

3.1 Convex architecture

Based on the work by Amos et al. (2017), we define ϕ as an ICNN. The architecture is an L-layered deep neural network with activations $\boldsymbol{z}^l$ given by:

$$\begin{aligned} \boldsymbol{z}^1 &= g^0(\boldsymbol{W}^0\boldsymbol{x} + \boldsymbol{b}^0), \\ \boldsymbol{z}^l &= g^{l-1}(\boldsymbol{W}^{l-1}\boldsymbol{x} + \boldsymbol{V}^{l-1}\boldsymbol{z}^{l-1} + \boldsymbol{b}^{l-1}) \text{ for } l = 2, \dots, L. \end{aligned} \tag{5}$$

The output is $\phi(\boldsymbol{x}) = \boldsymbol{z}^L$. All the weights $\boldsymbol{W}^l$ and $\boldsymbol{V}^l$ and the biases $\boldsymbol{b}^l$ are learnable parameters. The function ϕ is convex provided that all $\boldsymbol{V}^l$ are non-negative and all the activation functions g^l are convex and non-decreasing (Amos et al., 2017, Proposition 1). We set all the activation functions g^l to be the continuously differentiable exponential linear unit (CELU) (Barron, 2017) and the linear layers as convolutions. Once we have ϕ, we numerically approximate the evaluation of the mirror map $\Psi(\boldsymbol{x}) \approx \nabla\phi(\boldsymbol{x})$ using automatic differentiation (Paszke et al., 2017).

3.2 Inverse map

Since Ψ is a gradient of a neural network, its inverse Ψ^{-1} is not readily available. Our solution leverages the Fenchel conjugate (Fenchel, 1949) $\overline{\phi} : \mathcal{Z} \to \mathbb{R}$ of ϕ, which exists for convex ϕ, is again convex, and defined as:

$$\overline{\phi}(\boldsymbol{z}) = \max_{\boldsymbol{x}} \langle \boldsymbol{x}, \boldsymbol{z} \rangle - \phi(\boldsymbol{x}). \tag{6}$$

If ϕ is of so-called *Legendre type* (i.e., proper closed, essentially smooth and essentially strictly convex (Rockafellar, 1970)), then Fenchel (1949) states that $(\nabla\phi)^{-1} = \nabla\overline{\phi}$. Inspired by this equation, we define the conjugate $\overline{\phi}$ again as an ICNN with the exact same architecture as ϕ in (5). Given Ψ, it is trained by minimizing:[2]

$$\min_{\overline{\phi}, \overline{\Psi}} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}} \left[||\overline{\Psi}(\Psi(\boldsymbol{x})) - \boldsymbol{x}||_2 \right]. \tag{7}$$

Now $\overline{\Psi}(\boldsymbol{x}) \approx \nabla\overline{\phi}(\boldsymbol{x})$ is again computed using automatic differentiation and approximates $(\nabla\phi)^{-1}(\boldsymbol{x})$ as desired.

3.3 Training algorithm

A real-world corruption $\tau(\boldsymbol{x})$ (like blurred or with changed contrast) typically lies at a large L^2 distance ϵ (say 10) of the clean image $\boldsymbol{x}$ and thus an L^2-based attack with this ϵ would not find it but instead an extremely noisy one $\tilde{\boldsymbol{x}}$ at similar distance which would likely not be recognizable by a human (Fig. 1, left). Also, AT does not converge for large ϵ and typically very small ϵ around 0.1 are used (Hendrycks & Dietterich, 2019; Ford et al., 2019; Xie et al., 2020; Kang et al., 2019; Kireev et al., 2022).

Our basic idea is to train ϕ such that, with respect to the induced Bregman divergence, a suitable Bregman ball $\mathbb{B}_\phi$ includes $\tau(\boldsymbol{x})$ while excluding noisy images at much closer L^2 distance (Fig. 1, right). To do so, we first need a way to sample random images at a given mean distance from a clean image:

[2] In this expression, $\overline{\Psi}$ is not an explicit neural network but rather a gradient of a neural network ($\overline{\phi}$) computed w.r.t. the input.

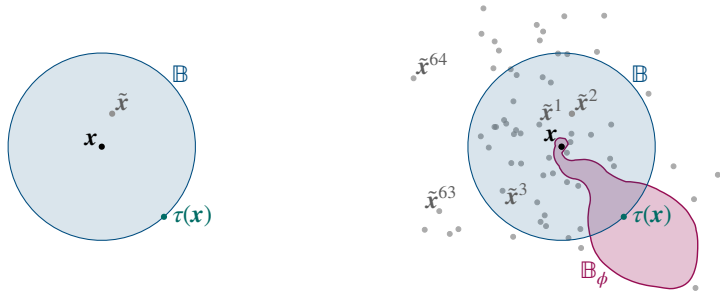


Figure 1: A two-dimensional cartoon visualizing our approach. Left: a clean image $\boldsymbol{x}$ corrupted to $\tau(\boldsymbol{x})$ whose inclusion in an L^2 ball $\mathbb{B}$ requires a large radius. An attack with this distance would yield an extremely noisy $\tilde{\boldsymbol{x}}$. Right: our learned Bregman distance yields balls that can include $\tau(\boldsymbol{x})$ while excluding noisy images $\tilde{\boldsymbol{x}}$ (here 64 many) at much closer L^2 distance.

Lemma 1 *When sampling the isotropic Gaussian random variable*

$$\tilde{\mathrm{x}} = \boldsymbol{x} + d\,\frac{\Gamma\left(\frac{n}{2}\right)}{\sqrt{2}\,\Gamma\left(\frac{n+1}{2}\right)}\,||\tau(\boldsymbol{x}) - \boldsymbol{x}||_2\,\boldsymbol{\delta}\,,\;\; \boldsymbol{\delta} \sim \mathcal{N}(0, \boldsymbol{I}_n), \tag{8}$$

where $\boldsymbol{x}$ is a fixed clean image, $d \in (0,1)$ is a hyperparameter and Γ is the gamma function, we have

$$\mathbb{E}\left[||\tilde{\mathrm{x}} - \boldsymbol{x}||_2\right] = d||\tau(\boldsymbol{x}) - \boldsymbol{x}||_2. \tag{9}$$

The proof is given in Appendix A.1.

Fig. 1 (right) shows $m = 64$ such samples $\{\tilde{\boldsymbol{x}}^i\}_{i=1}^m$ using $d = 1$. Next, we force each of their divergences $D_\phi(\tilde{\boldsymbol{x}}^i \parallel \boldsymbol{x})$ to be larger than $D_\phi(\tau(\boldsymbol{x}) \parallel \boldsymbol{x})$ or equivalently $-D_\phi(\tau(\boldsymbol{x}) \parallel \boldsymbol{x}) > -D_\phi(\tilde{\boldsymbol{x}}^i \parallel \boldsymbol{x})$. We propose the following *Bregman loss* $l_B(\boldsymbol{x}; \phi, \Psi)$ to enforce all these m inequalities at once:

$$l_B(\boldsymbol{x}; \phi, \Psi) = -\log \frac{e^{-D_\phi(\tau(\boldsymbol{x})\|\boldsymbol{x})}}{e^{-D_\phi(\tau(\boldsymbol{x})\|\boldsymbol{x})} + \sum_i e^{-D_\phi(\tilde{\boldsymbol{x}}^i\|\boldsymbol{x})}}. \tag{10}$$

$l_B(\boldsymbol{x}; \phi, \Psi)$ can be interpreted as a cross entropy where the logits vector is the negative of Bregman divergences $\left[-D_\phi(\tau(\boldsymbol{x}) \parallel \boldsymbol{x}), -D_\phi(\tilde{\boldsymbol{x}}^1 \parallel \boldsymbol{x}), ..., -D_\phi(\tilde{\boldsymbol{x}}^m \parallel \boldsymbol{x})\right]$ and the ground truth class always corresponds the first entry. Then, we learn ϕ by minimizing:

$$\min_{\phi,\Psi} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[l_B(\boldsymbol{x}; \phi, \Psi)\right]. \tag{11}$$

After a successful training where $D_\phi(\tau(\boldsymbol{x}) \parallel \boldsymbol{x}) < D_\phi(\tilde{\boldsymbol{x}}^i \parallel \boldsymbol{x})$ for all $i = 1, ..., m$, the Bregman ball $\mathbb{B}_\phi(\boldsymbol{x}, D_\phi(\tau(\boldsymbol{x}) \parallel \boldsymbol{x}))$ contains the transformed image $\tau(\boldsymbol{x})$ by definition but does not contain any of the noisy images $\{\tilde{\boldsymbol{x}}^i\}_{i=1}^m$ (Fig. 1, right).

3.4 Bregman-based semantic attack

Given a learned Bregman divergence, we define the neighborhood of a clean image $\boldsymbol{x}$ as the intersection of a Bregman ball and fixed L^2 ball:

$$\mathbb{S}(\boldsymbol{x}) = \mathbb{B}_\phi(\boldsymbol{x}, \epsilon) \cap \mathbb{B}(\boldsymbol{x}, \epsilon_{max}). \tag{12}$$

Here, as expected, ϵ is a parameter used in the adversarial attacks; ϵ_{max} is an empirical large value, typically chosen a hundred times bigger than the usual box radii used for the L^2-based ATs. We empirically found this L^2 bound to be necessary. The reason is that most random samples used for training with (11) are near $\boldsymbol{x}$ w.r.t L^2 and thus D_ϕ might assign also small values in "under-explored" regions that are far from $\boldsymbol{x}$. As a result, a Bregman ball of radius $D_\phi(\tau_s(\boldsymbol{x}) \parallel \boldsymbol{x})$ may include these regions. Experimentally, we found that imposing a large L^2-bound on $\mathbb{B}_\phi$ eliminates this problem.

Algorithm 1 Bregman-based semantic attack

1: $\boldsymbol{x}' \leftarrow \boldsymbol{x}$ ▷ Initialization
2: **for** $t = 1, ..., T$ **do**
3: $\quad \eta \leftarrow \epsilon 10^{-4t/T}$
4: $\quad \boldsymbol{x}' \leftarrow \overline{\Psi}\left(\Psi(\boldsymbol{x}') + \eta \nabla_{\boldsymbol{x}} l(\boldsymbol{x}', y; \theta)\right)$ ▷ The mirror descent update explained in Sec. 2
5: $\quad \boldsymbol{x}' \leftarrow \boldsymbol{x} - \epsilon_{max}(\boldsymbol{x} - \boldsymbol{x}')/||\boldsymbol{x}' - \boldsymbol{x}||_2$ ▷ Projecting $\boldsymbol{x}'$ into $\mathbb{B}(\boldsymbol{x}, \epsilon_{max})$
6: $\quad a, b \leftarrow 0, 1$
7: $\quad$ **while** $D_\phi(\boldsymbol{x}' \parallel \boldsymbol{x}) > \epsilon$ **do** ▷ Projecting $\boldsymbol{x}'$ into $\mathbb{B}_\phi(\boldsymbol{x}, \epsilon)$
8: $\quad\quad m \leftarrow (a + b)/2$
9: $\quad\quad \boldsymbol{x}' \leftarrow \boldsymbol{x} + m(\boldsymbol{x}' - \boldsymbol{x})$
10: $\quad\quad a \leftarrow m$ **if** $D_\phi(\boldsymbol{x}' \parallel \boldsymbol{x}) > \epsilon$ **else** $b \leftarrow m$
11: $\quad \boldsymbol{x}' \leftarrow \texttt{clip}(\boldsymbol{x}', 0, 1)$ ▷ Projecting $\boldsymbol{x}'$ into $[0, 1]^n$
12: **return** $\boldsymbol{x}'$ ▷ $\boldsymbol{x}'$ is potentially misclassified

Our *Bregman-based semantic attack* follows the mirror descent as outlined in the fourth column of Table 1 and detailed in Algorithm 1. The projection of $\boldsymbol{x}'$ into $\mathbb{S}$ is done as a projection into $\mathbb{B}(\boldsymbol{x}, \epsilon_{max})$ followed by a projection into $\mathbb{B}_\phi(\boldsymbol{x}, \epsilon)$. Since the latter has no closed-from expression, we approximate it by a binary search over the segment having $\boldsymbol{x}$ and $\boldsymbol{x}'$ as endpoints (lines 7–11 in Algorithm 1).

4 Related work

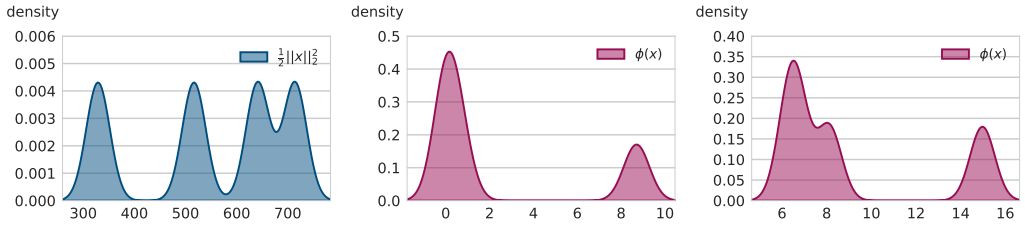
Corruption robustness via data augmentation. Much of the prior literature on corruption robustness aims to improve out-of-distribution generalization by using simulated and augmented images for training. Many such data augmentation techniques are based on creating synthetic training examples through mixing pairs of training images and their labels. This is achieved for example by linear weighted blending of images (Zhang et al., 2018a) or by cutting and pasting parts of an image onto another (Yun et al., 2019). Researchers also fused images based on masks computed through frequency spectrum analysis (Harris et al., 2021), based on adaptive masks (Liu et al., 2022) or based on model-generated features (Walawalkar et al., 2020). Other works considered a hybrid version of these mixing methods (Park et al., 2022), a stochastic version of them (Park et al., 2022), an ensemble of them (Yin et al., 2022) or a concurrent combination of them (Liang et al., 2023).

Adversarial attacks without L^p norms. Another line of related work focuses on adversarial image perturbations that are not constrained by L^p norms. Hsiung et al. (2023) introduces semantic adversarial attacks that target image transformation parameters instead of image pixels. Similarly, Engstrom et al. (2019) targets spatial transformations. Hosseini & Poovendran (2018) manipulates the hue and saturation components in the hue saturation value (HSV) color space to create adversarial examples. In addition to colorization, Bhattad et al. (2019) also tweaked texture of objects within images. (Shamsabadi et al., 2020) modified colors within the invisible range. Some works altered the semantic features of images through conditional generative models (Joshi et al., 2019) or conditional image editing(Qiu et al., 2020).

Robustness via learned similarity metric. The closest related work adopts the so-called learned perceptual image patch similarity (LPIPS) to study robustness. LPIPS is a weighted sum of the L^2 of the feature maps taken from the activation layers of a trained convolutional network:

$$\text{LPIPS}(\boldsymbol{x}, \boldsymbol{x}') = \sum_l w_l ||\omega_l(\boldsymbol{x}) - \omega_l(\boldsymbol{x}')||_2 \tag{13}$$

where ω_l is the feature map up to the l-layer and w_l weighs the contribution of the layer l. Wang et al. (2021) and Luo et al. (2022) propose an attack similar to (Carlini & Wagner, 2017) by adding the LPIPS along with the L^p norm. Differently, Kireev et al. (2022) and Laidlaw et al. (2021) used LPIPS as a function to define the set of similar images (refer to Sec. 2 for notation context): $\mathbb{S}(\boldsymbol{x}) = \{\boldsymbol{x}' \in \mathbb{R}^n | \text{ LPIPS}(\boldsymbol{x}, \boldsymbol{x}') \leq \epsilon\}$. Since the projection into this LPIPS-based set does not admit a closed-from expression, solving the inner maximization problem of (1) (i.e., performing the adversarial attack) requires approximation (Laidlaw et al., 2021) or relaxation (Kireev et al., 2022).



(a) Euclidean Space (b) Contrast (c) Fog

Figure 2: Distribution of Bregman divergence's base functions over 10,000 CIFAR-10 test set images. Two trained base functions ϕ, (b) for contrast and (c) for fog, are compared against (a) their counterpart in the PGD setting, the half of norm L^2 squared (see Table 1).

The resulting attacks and their associated AT have been proven effective to train robust models against common image corruptions. We compare against LPIPS in our experiments.

5 Evaluation

We perform experiments on CIFAR-10 (Krizhevsky et al., 2009) and the common corruption dataset CIFAR-10-C (Hendrycks & Dietterich, 2019). For the classification model, we use the PreAct ResNet-18 architecture (He et al., 2016) the same used by Kireev et al. (2022). The image corruptions are from (Hendrycks & Dietterich, 2019) and come with severities from 1 to 5. Our focus is on the corruptions of contrast and fog which have been found the most challenging in (Ford et al., 2019; Kireev et al., 2022).

Training the Bregman divergence. Both ϕ and $\overline{\phi}$ have the same architecture, an ICNN with 12 convolutional layers followed by 4 fully connected layers. The mirror map and the inverse mirror are numerically approximated using `autograd.grad` from PyTorch's automatic differentiation engine (Paszke et al., 2017). As an initialization phase, we first train ϕ and $\overline{\phi}$ such that Ψ and $\overline{\Psi}$ approximate the identity function (so initially $\overline{\Psi} = \Psi^{-1}$ holds) on uniformly drawn samples from the usual range of pixels $[0, 1]^n$:

$$\min_{\phi,\Psi} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{U}([0,1]^n)} \left[||\Psi(\boldsymbol{x}) - \boldsymbol{x}||_2\right], \quad \min_{\overline{\phi},\overline{\Psi}} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{U}([0,1]^n)} \left[||\overline{\Psi}(\boldsymbol{x}) - \boldsymbol{x}||_2\right]. \tag{14}$$

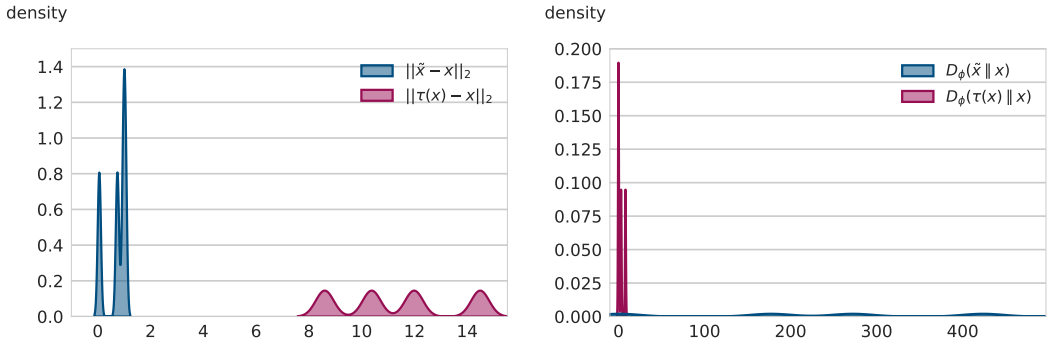
This training is performed for 10,000 steps using the Adam optimizer (Kingma & Ba, 2014) with a learning rate of 0.001 and a weight decay of 10^{-10}.

Next, for a given corruption τ, we train ϕ with (11) while randomly sampling its severity for each image at each epoch. The hyperparameter d for sampling noisy images defined in (8) is uniformly sampled in $[10^{-7}, 10^{-1}]$. Since the fraction of Γ functions in (8) is difficult to compute, we replace it empirically by $1/\sqrt{n}$. Doing so yields $\sqrt{\mathbb{E}\left[||\tilde{\mathrm{x}} - \boldsymbol{x}||_2^2\right]} = d||\tau_s(\boldsymbol{x}) - \boldsymbol{x}||_2$ instead of Lemma 1 (Appendix A.1). We use $m = 31$ samples per clean image while a batch consists of 16 clean images.

The training is performed for 20 epochs using the AdamW optimizer (Loshchilov & Hutter, 2017) with a learning rate of 0.0001 and a weight decay of 10^{-11}. After each update of ϕ (according to Equ. 11), we also update $\overline{\phi}$ (according to Equ. 7). Finally, we freeze the parameters of ϕ and continue training $\overline{\phi}$ for additional 20 epochs.

Results of the Bregman divergence. We first inspect the learned base functions ϕ, i.e., our trained "entropy" for an image corruption. The distribution of its outputs on the test set is shown in Fig. 2. We notice that the trained based functions have modalities with different shapes and heights compared to the L^2 norm, the base function in the L^2-based threat models (see Table 1).

Next we show in Fig. 3 that the learned divergence D_ϕ agrees with Fig. 1. To do so we consider, for the entire test set of 10,000 clean images, noisy images (blue, one per clean image) and the set of contrast-corrupted images (red). We compute the distribution of their L^2 distances to the clean image in Fig. 3a. It is small for the noisy images (by construction) and large for the corrupted



(a) Euclidean distance

(b) Trained Bregman divergence

Figure 3: Distribution of (a) the Euclidean distance and (b) our trained Bregman divergence for noisy images $\tilde{\boldsymbol{x}}$ (blue) and contrast-corrupted images $\tau(\boldsymbol{x})$ (red) over 10,000 CIFAR-10 test set images.

ones. Fig. 3b shows the distribution of their learned divergences to the clean image.[3] Here, the corrupted images are considered close but the noisy ones far, which shows that the learned Bregman divergence is semantically meaningful and works as expected.

Adversarial training. After training a Bregman divergence to be semantically meaningful for a corruption τ and using it in the definition of the threat model, we can run an AT using our Bregman-based semantic attack. We call this procedure the *Bregman-based adversarial training (BAT)*. For a corruption type τ we write BAT(τ).

We compare against the relaxed LPIPS AT (RLAT) (Kireev et al., 2022) without any further fine-tuning of the training parameters. For a fair comparison, we set the number of iteration of our attack to $T = 1$ to match the one-step attack used in RLAT. Fig. 4 shows a sample of the resulting images for contrast corruptions. The training is performed using the SGD optimizer for 150 epochs with a learning rate of 0.1 that decays by a factor of 10 each 50 epochs, a batch size of 128, and a weight decay of 0.0005. These are the same hyperparameters for which RLAT performs the best. The RLAT radius is taken to be 0.08. We also compare against the L^2 PGD AT with the radius of 0.1, which Kireev et al. (2022) found the most effective for corruption robustness. We emphasize that there is no meaning in the relation of the magnitudes between L^2 radius values and Bregman ball radius values.

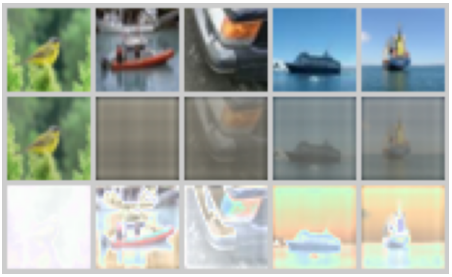


Figure 4: Samples from training images (first row), adversarial examples for contrast corruption found by our Bregman-based attack (second row) and the pixel-wise difference.

We run BAT for different values of Bregman ball radius ϵ from 0.025 to 20 while fixing $\epsilon_{max} = 10$ in (12). We notice that the accuracy of the resulting classifier is not very sensitive to the choice of ϵ unlike in L^p- and LPIPS-based threat models (Kireev et al., 2022).

For contrast corruptions, the best accuracy is achieved with $\epsilon = 0.2$ which we use in the comparison in Table 2. The standard accuracy is on the clean data set. The columns show the corruption accuracy for different severities and the last is their average. BAT preserves the standard accuracy (a slight drop of 0.2%) while maintains high accuracies even under high corruption severities where the other methods start to fail.

Additionally, we trained for fog and zoom blur corruptions and report the average results in Table 3 together with the prior contrast corruption. Fog is again best by a margin, whereas for zoom blur we perform about 5% worse. Surprisingly our training for contrast also performs best on fog and well

[3]The mean, std, min and max of the distributions are reported in Appendix B.

Table 2: Comparison of corruption robustness of models trained under different regimes.

	Standard accuracy	Contrast					Average
		$s=1$	$s=2$	$s=3$	$s=4$	$s=5$	
Standard training	94.82	94.25	90.57	86.35	76.42	34.72	76.46
L^2-based AT	93.52	91.68	82.96	72.31	51.43	21.26	63.92
RLAT	93.27	91.47	82.32	70.65	48.35	21.58	62.87
BAT(contrast)	94.62	94.38	93.74	93.07	91.58	84.04	91.36

Table 3: Corruption robustness of the standard-trained model against adversarially trained models under L^2, RLAT, and BAT for different corruptions.

	Standard	Contrast	Fog	Zoom Blur
Standard training	94.82	76.46	87.04	77.22
L^2-based AT	93.52	63.92	77.47	85.87
RLAT	93.27	62.87	77.00	85.88
BAT(contrast)	94.62	91.36	91.47	80.11
BAT(fog)	94.66	79.53	89.20	80.42
BAT(zoom blur)	93.86	89.60	90.13	80.62

for zoom blur. One hint is that the three corruptions are similar in nature but this behavior requires further investigation.

Limitation and discussion. The current main limitation of our approach is that it is specific to the corruption type, and, as shown by zoom blur, does not always perform best. Further, there is an overhead in first training for a valid divergence before starting the AT. On the other hand we could significantly improve robustness on contrast and fog where the other methods fail to improve over standard training. Also, our method produces adversarial examples unlike the work by Kireev et al. (2022) that perturbs the feature space with no mechanism to produce an associated image. Finally, we see value in the theoretical underpinning, which yields desirable properties (e.g., the Bregman ball is compact unlike the LPIPS-based sets) and the well-established mirror descent. In fact, the learned base function and divergence may be interesting for other applications. We hope that scaling to larger convex architectures (the ones presented in this work are tiny; contain only 2M parameters) would yield base functions ϕ that generalize to large sets of corruption types.

6 Conclusion

We presented a novel approach and tool set to improve corruption robustness of neural networks for image classification. The key idea was to first learn a similarity measure that semantically captures a considered corruption and that is also theoretically sound by instantiating a Bregman divergence. Doing so gave access to executing mirror descent and thus adversarial attacks and training. The main, and significant, challenge in our work was to ensure that the learned base functions underlying the divergence satisfy the properties required by the theory.

Our results are prototypical and can only be considered a first step but we consider them in strong support of our novel contribution. Specifically, we demonstrated that the learned divergence measures similarity as intended and we could demonstrate a significantly improved corruption robustness for two corruptions on which prior work failed. Thus we see great potential when using large network models and training sets. Finally, due to the underlying theory, our work is not specific to images and learned divergences may find applications outside the scope of this paper.

REFERENCES

Brandon Amos, Lei Xu, and J. Zico Kolter. Input convex neural networks. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 146–155. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/amos17b.html.

Jonathan T. Barron. Continuously differentiable exponential linear units. *CoRR*, abs/1704.07483, 2017. URL http://arxiv.org/abs/1704.07483.

Heinz H. Bauschke and Jonathan Michael Borwein. Legendre functions and the method of random bregman projections. 1997. URL https://api.semanticscholar.org/CorpusID:43970672.

Anand Bhattad, Min Jin Chong, Kaizhao Liang, B. Li, and David Alexander Forsyth. Unrestricted adversarial examples via semantic manipulation. In *International Conference on Learning Representations*, 2019. URL https://api.semanticscholar.org/CorpusID:213304602.

Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pp. 387–402. Springer, 2013.

L.M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7(3):200 – 217, 1967. ISSN 0041-5553. doi: https://doi.org/10.1016/0041-5553(67)90040-7. URL http://www.sciencedirect.com/science/article/pii/0041555367900407.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57, 2017. doi: 10.1109/SP.2017.49.

Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, Percy Liang, and John C. Duchi. *Unlabeled Data Improves Adversarial Robustness*. Curran Associates Inc., Red Hook, NY, USA, 2019.

Jinghui Chen, Yu Cheng, Zhe Gan, Quanquan Gu, and Jingjing Liu. Efficient robust training via backward smoothing, 2021. URL https://openreview.net/forum?id=49V11oUejQ.

Sayak Ray Chowdhury, Patrick Saux, Odalric Maillard, and Aditya Gopalan. Bregman deviations of generic exponential families. In Gergely Neu and Lorenzo Rosasco (eds.), *Proceedings of Thirty Sixth Conference on Learning Theory*, volume 195 of *Proceedings of Machine Learning Research*, pp. 394–449. PMLR, 12–15 Jul 2023. URL https://proceedings.mlr.press/v195/chowdhury23a.html.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, 2020.

Imre Csiszar and Frantisek Matus. On minimization of entropy functionals under moment constraints. In *2008 IEEE International Symposium on Information Theory*, pp. 2101–2105, 2008. doi: 10.1109/ISIT.2008.4595360.

Samuel Dodge and Lina Karam. A study and comparison of human and deep learning recognition performance under visual distortions. In *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–7, 2017. doi: 10.1109/ICCCN.2017.8038465.

Herbert Edelsbrunner and Hubert Wagner. Topological Data Analysis with Bregman Divergences. In Boris Aronov and Matthew J. Katz (eds.), *33rd International Symposium on Computational Geometry (SoCG 2017)*, volume 77 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 39:1–39:16, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. ISBN 978-3-95977-038-5. doi: 10.4230/LIPIcs.SoCG.2017.39. URL http://drops.dagstuhl.de/opus/volltexte/2017/7198.

Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. Exploring the landscape of spatial robustness. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1802–1811. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/engstrom19a.html.

W. Fenchel. On conjugate convex functions. *Canadian Journal of Mathematics*, 1(1):73–77, 1949. doi: 10.4153/CJM-1949-007-x.

Nicolas Ford, Justin Gilmer, and Ekin D. Cubuk. Adversarial examples are a natural consequence of test error in noise, 2019. URL https://openreview.net/forum?id=S1xoy3CcYX.

Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997. ISSN 0022-0000. doi: https://doi.org/10.1006/jcss.1997.1504. URL https://www.sciencedirect.com/science/article/pii/S002200009791504X.

BÉla A. Frigyik, Santosh Srivastava, and Maya R. Gupta. Functional bregman divergence and bayesian estimation of distributions. *IEEE Transactions on Information Theory*, 54(11):5130–5139, 2008. doi: 10.1109/TIT.2008.929943.

Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. URL http://arxiv.org/abs/1412.6572.

Mehrtash Harandi, Mathieu Salzmann, and Fatih Porikli. Bregman divergences for infinite dimensional covariance matrices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1003–1010, 2014.

Ethan Harris, Antonia Marcu, Matthew Painter, Mahesan Niranjan, Adam Prugel-Bennett, and Jonathon Hare. {FM}ix: Enhancing mixed sample data augmentation, 2021. URL https://openreview.net/forum?id=oev4KdikGjy.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pp. 630–645. Springer, 2016.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HJz6tiCqYm.

Hossein Hosseini and Radha Poovendran. Semantic adversarial examples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1614–1619, 2018.

Lei Hsiung, Yun-Yun Tsai, Pin-Yu Chen, and Tsung-Yi Ho. Towards compositional adversarial robustness: Generalizing adversarial training to composite semantic perturbations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24658–24667, June 2023.

Yulun Jiang, Chen Liu, Zhichao Huang, Mathieu Salzmann, and Sabine Susstrunk. Towards stable and efficient adversarial training against l_1 bounded adversarial attacks. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 15089–15104. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/jiang23f.html.

Ameya Joshi, Amitangshu Mukherjee, Soumik Sarkar, and Chinmay Hegde. Semantic adversarial attacks: Parametric transformations that fool deep classifiers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4773–4783, 2019.

Daniel Kang, Yi Sun, Tom Brown, Dan Hendrycks, and Jacob Steinhardt. Transfer of adversarial robustness between perturbation types. *arXiv preprint arXiv:1905.01034*, 2019.

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

Klim Kireev, Maksym Andriushchenko, and Nicolas Flammarion. On the effectiveness of adversarial training against common corruptions. In *The 38th Conference on Uncertainty in Artificial Intelligence*, 2022. URL https://openreview.net/forum?id=BcU_UIIjqg9.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

S. Kullback and R. A. Leibler. On information and sufficiency. *Ann. Math. Statist.*, 22(1):79–86, 1951.

Cassidy Laidlaw, Sahil Singla, and Soheil Feizi. Perceptual adversarial robustness: Defense against unseen threat models. In *ICLR*, 2021.

Wen Liang, Youzhi Liang, and Jianguo Jia. Miamix: Enhancing image classification through a multi-stage augmented mixed sample data augmentation method, 2023.

Zicheng Liu, Siyuan Li, Di Wu, Zhiyuan Chen, Lirong Wu, Jianzhu Guo, and Stan Z. Li. Automix: Unveiling the power of mixup for stronger classifiers. In *European Conference on Computer Vision*, pp. 441–458, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Cheng Luo, Qinliang Lin, Weicheng Xie, Bizhu Wu, Jinheng Xie, and Linlin Shen. Frequency-driven imperceptible adversarial attack on semantic similarity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

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. URL https://openreview.net/forum?id=rJzIBfZAb.

Arkadij Semenovič Nemirovskij and David Borisovich Yudin. Problem complexity and method efficiency in optimization. 1983.

Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pp. 372–387. IEEE, 2016.

Chanwoo Park, Sangdoo Yun, and Sanghyuk Chun. A unified analysis of mixed sample data augmentation: A loss function perspective. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=SLdfxFdIFeN.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

Haonan Qiu, Chaowei Xiao, Lei Yang, Xinchen Yan, Honglak Lee, and Bo Li. Semanticadv: Generating adversarial examples via attribute-conditioned image editing. In *ECCV*, 2020.

Sylvestre-Alvise Rebuffi, Sven Gowal, Dan Andrei Calian, Florian Stimberg, Olivia Wiles, and Timothy Mann. Data augmentation can improve robustness. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=kgVJBBThdSZ.

R. Tyrrell Rockafellar. *Convex analysis*. Princeton Mathematical Series. Princeton University Press, Princeton, N. J., 1970.

Ali Shahin Shamsabadi, Ricardo Sanchez-Matilla, and Andrea Cavallaro. Colorfool: Semantic adversarial colorization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, Washington, USA, 2020.

Wolfgang Stummer and Igor Vajda. On bregman distances and divergences of probability measures. *IEEE Transactions on Information Theory*, 58:1277–1288, 2009. URL https://api.semanticscholar.org/CorpusID:14442250.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. January 2014. 2nd International Conference on Learning Representations, ICLR 2014 ; Conference date: 14-04-2014 Through 16-04-2014.

Jonathan Uesato, Jean-Baptiste Alayrac, Po-Sen Huang, Robert Stanforth, Alhussein Fawzi, and Pushmeet Kohli. *Are Labels Required for Improving Adversarial Robustness?* Curran Associates Inc., Red Hook, NY, USA, 2019.

Devesh Walawalkar, Zhiqiang Shen, Zechun Liu, and Marios Savvides. Attentive cutmix: An enhanced data augmentation approach for deep learning based image classification. *CoRR*, abs/2003.13048, 2020. URL https://arxiv.org/abs/2003.13048.

Yajie Wang, Shangbo Wu, Wenyi Jiang, Shengang Hao, Yu-an Tan, and Quanxin Zhang. Demiguise attack: Crafting invisible semantic adversarial perturbations with perceptual similarity. *CoRR*, abs/2107.01396, 2021. URL https://arxiv.org/abs/2107.01396.

Eric Wong, Leslie Rice, and J. Zico Kolter. Fast is better than free: Revisiting adversarial training. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BJx040EFvH.

Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan L. Yuille, and Quoc V. Le. Adversarial examples improve image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Hao Yin, Dongyu Cao, and Ying Zhou. Randommix: An effective framework to protect user privacy information on ethereum. In *2022 IEEE 22nd International Conference on Software Quality, Reliability, and Security Companion (QRS-C)*, pp. 764–765, 2022. doi: 10.1109/QRS-C57518.2022.00124.

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 *International Conference on Computer Vision (ICCV)*, 2019.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pp. 7472–7482. PMLR, 2019.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018a. URL https://openreview.net/forum?id=r1Ddp1-Rb.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018b.

A PROOFS

A.1 LEMMA 1

Let $\boldsymbol{x} \in \mathbb{R}^n$ a fixed image and $\tilde{\mathrm{x}}$ a random variable defined as follows with $\mu > 0$:

$$\tilde{\mathrm{x}} = \boldsymbol{x} + \mu\boldsymbol{\delta}\ ,\ \boldsymbol{\delta} \sim \mathcal{N}(0, \boldsymbol{I}_n), \tag{15}$$

The random variable $\tilde{\mathrm{x}}$ is a gaussian because it is a linear combination of gaussians ($\boldsymbol{x}$ is fixed).

$$\begin{aligned} ||\tilde{\mathrm{x}} - \boldsymbol{x}||_2 &= \sqrt{\sum_i (\tilde{\mathrm{x}}_i - \boldsymbol{x}_i)^2} \\ &= \mu\sqrt{\sum_i \boldsymbol{\delta}_i^2} \\ &= \mu\sqrt{\mathrm{z}} \end{aligned} \tag{16}$$

z is a chi square distribution of degree n with density:

$$p_{\mathrm{z}}(z) = \frac{z^{n/2-1}e^{-z/2}}{2^{n/2}\Gamma(\frac{n}{2})} \tag{17}$$

We defined the variable $\mathrm{u} = f(\mathrm{z}) = \sqrt{\mathrm{z}}$, $\mathrm{u} \geq 0$. The density of u can be computed by the change or variable formula:

$$\begin{aligned} p_{\mathrm{u}}(u) &= p_{\mathrm{z}}(f^{-1}(u)) \left|\frac{dz}{du}\right| \\ &= \frac{u^{n-1}e^{-u^2/2}}{2^{n/2-1}\Gamma(\frac{n}{2})} \end{aligned} \tag{18}$$

Next, we compute the expectation of u:

$$\begin{aligned} \mathbb{E}(\mathrm{u}) &= \int_0^\infty u p_{\mathrm{u}}(u) du \\ &= \frac{1}{2^{n/2-1}\Gamma(\frac{n}{2})} \int_0^\infty u^n e^{-u^2/2} du \\ &= \frac{\sqrt{2}}{\Gamma(\frac{n}{2})} \int_0^\infty t^{(n-1)/2} e^{-t} dt \qquad \text{(by substituting } \quad u = \sqrt{2t}) \\ &= \frac{\sqrt{2}}{\Gamma(\frac{n}{2})} \Gamma(\frac{n+1}{2}) \qquad \text{(by definition of } \Gamma) \end{aligned} \tag{19}$$

So we have:

$$\mathbb{E}(||\tilde{\mathrm{x}} - \boldsymbol{x}||_2) = \mathbb{E}(\mu\mathrm{u}) = \mu\frac{\sqrt{2}\Gamma(\frac{n+1}{2})}{\Gamma(\frac{n}{2})} \tag{20}$$

In Equ. 15, we set:

$$\mu = d\ \frac{\Gamma\left(\frac{n}{2}\right)}{\sqrt{2}\,\Gamma\left(\frac{n+1}{2}\right)}\ ||\tau(\boldsymbol{x}) - \boldsymbol{x}||_2 \tag{21}$$

to obtain the same expression as in Lemma 1

A.2 AN ALTERNATIVE TO LEMMA 1

For large values of n, the constants $\Gamma\left(\frac{n}{2}\right)$ and $\Gamma\left(\frac{n+1}{2}\right)$ are intractable. Here we propose an alternative. With the same notation as above :

$$\mathbb{E}(||\tilde{\mathrm{x}} - \boldsymbol{x}||_2^2) = \mu^2 \mathbb{E}(\mathrm{z}) = \mu^2 n. \tag{22}$$

In Equ. 15, we set:

$$\mu = d\ \frac{1}{\sqrt{n}}\ ||\tau(\boldsymbol{x}) - \boldsymbol{x}||_2 \tag{23}$$

and obtain:

$$\sqrt{\mathbb{E}(||\tilde{\mathrm{x}} - \boldsymbol{x}||_2^2)} = d||\tau(\boldsymbol{x}) - \boldsymbol{x}||_2. \tag{24}$$

B DETAILS ABOUT THE DIVERGENCE EVALUATION

In Sec. 5, we have shown that our trained Bregman divergence is semantically meaningful as it considers noisy image far off clean images and the corrupted images closer even when the L^2 says otherwise. The distribution of these Bregman and L^2 values that are shown in Fig. 3 are further described in Tab. 4.

Table 4: Description of the distributions from Fig. 3.

	$\|\|\tau(\boldsymbol{x}) - \boldsymbol{x}\|\|_2$	$\|\|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|\|_2$	$D_\phi(\tau(\boldsymbol{x}) \parallel \boldsymbol{x})$	$D_\phi(\tilde{\boldsymbol{x}} \parallel \boldsymbol{x})$
mean	11.36	0.70	2.79	218.69
std	2.16	0.39	3.31	153.34
min	8.60	0.06	10^{-6}	1.11
max	14.49	1.04	8.11	423.90

C THE EVOLUTION OF THE TRAINING/VALIDATION LOSS

Figure 5 illustrates the convergence of the Bragman training under the settings of Sec. 5.

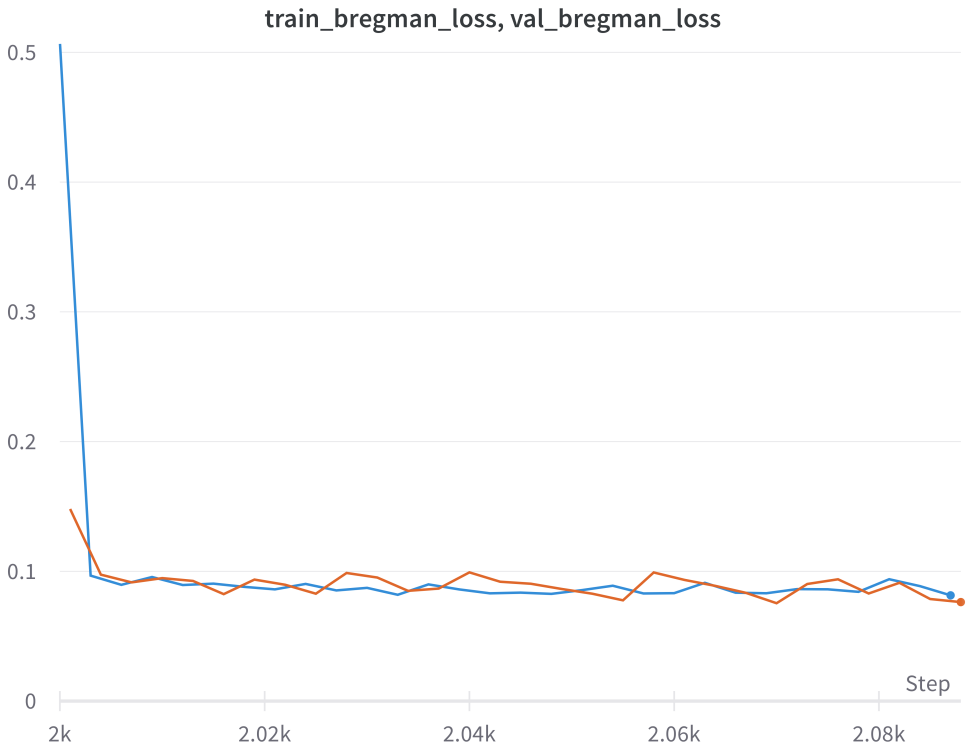


Figure 5: The evolution of the Bregman loss $l_B(\boldsymbol{x}; \phi, \Psi)$ for the training and the validation set across optimization steps.