Shedding More Light on Robust Classifiers under the lens of Energy-based Models — Supplementary material —

Mujtaba Hussain Mirza[1], Maria Rosaria Briglia[1], Senad Beadini[2], and Iacopo Masi[1]

[1] OmnAI Lab, CS Department, Sapienza University of Rome, Italy
[2] Eustema S.p.A. Italy
{mirza,briglia,masi}@di.uniroma1.it
s.beadini@eustema.it

A Appendix

A.1 Energy in function of PGD Steps

Similar to Fig. 1(a) in the paper, Fig. A shows the dependency using three different architectures with diverse depths for CIFAR 100. In particular, Fig. A reveals that increasing the number of classes by an order of magnitude—from 10 to 100—reduces the gap of the energies across different model depths. In Fig. A the energies are all collapsing to -50 while in Fig. 1(a) in the paper there are more variations.

In Fig. B, utilizing the WideResnet-28-10 [4], we observed the same intriguing trend where the energy $E_{\boldsymbol{\theta}}(\mathbf{x})$ associated with adversarial inputs reduces as the intensity of the attack amplifies. Notice that we quantify the attack's intensity by the discrete count of steps undertaken in a PGD

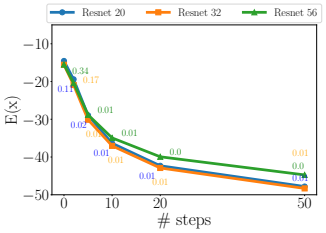
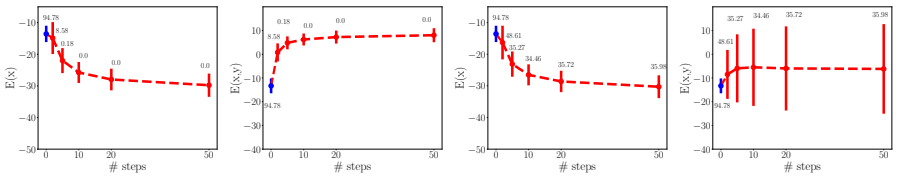


Fig. A: $E_{\boldsymbol{\theta}}(\mathbf{x})$ w.r.t. to PGD on CIFAR 100. For each point we report the robust accuracy.



(1) PGD [9] **(2)** PGD [9] **(3)** TRADES [17] **(4)** TRADES [17]

Fig. B: Dependency of $E_{\boldsymbol{\theta}}(\mathbf{x})$ and $E_{\boldsymbol{\theta}}(\mathbf{x}, y)$ w.r.t. number of steps of PGD. We show classic PGD using CE loss and TRADES using KL divergence on a *non-robust* WideResnet-28-10 [4]. Each point of the plot also reports the robust accuracy and the standard deviation of the energy values. Note how TRADES has higher std. dev. for $E_{\boldsymbol{\theta}}(\mathbf{x}, y)$ given that the distribution is bimodal.

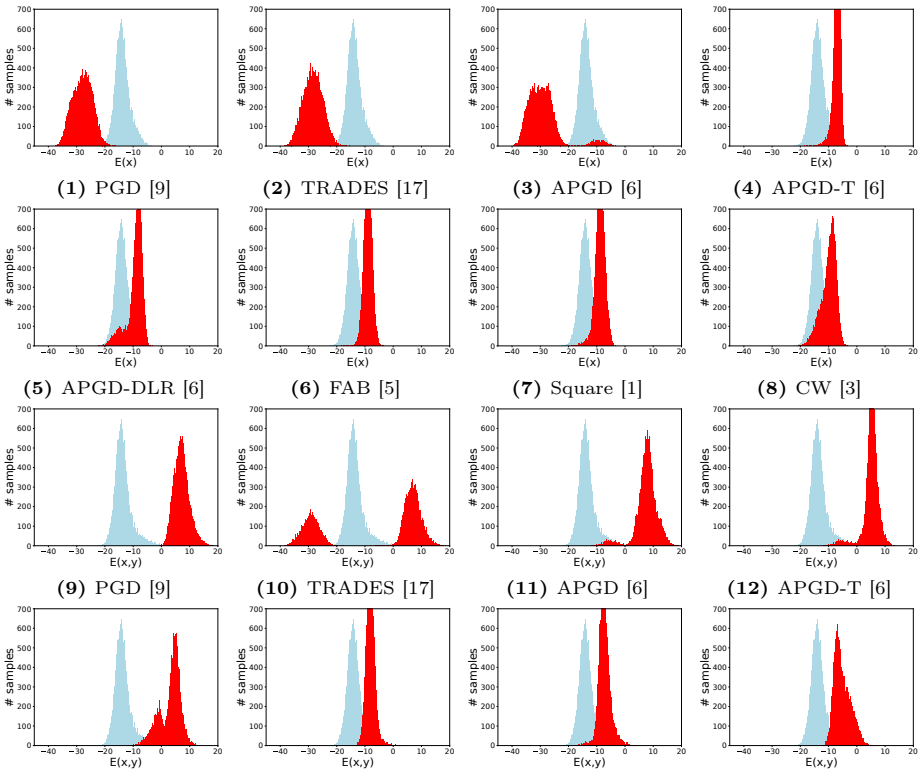
attack. In this plot, in addition to what we show in the paper, we have also added the trend for $E_{\boldsymbol{\theta}}(\mathbf{x}, y)$ that goes up. Notably, while PGD and TRADES[3] have the same trends in terms of average energies, their spread is very different with TRADES having a much larger standard deviation than PGD, given that TRADES show a bimodal $E_{\boldsymbol{\theta}}(\mathbf{x}, y)$ distribution—see Fig. 2(b) in the paper.

In Fig. C, we present a high-resolution version of the Fig. 2 in the paper, where we show the conditional and marginal energy distribution for a diverse set of state-of-the-art adversarial attacks. All the attacks except for CW are produced with a deformation of input given by $\ell_\infty \leq \epsilon = 8/255$ and a step size of $2/255$. The CW attack operates under an ℓ_2 perturbation constraint. For PGD, APGD, TRADES, and FAB we operate with 20 steps, while for Square and CW we used 1000 queries and 200 steps, respectively. All these observations, when reevaluated through the Energy-Based Model perspective, lead to an insightful deduction. Moving beyond the traditional notion that adversarial attacks merely cross the decision boundary, our research suggests that DNNs are predisposed to consider adversarial examples as extremely probable according to the hidden generative model.

A.2 Energy Dynamics during Adversarial Training

We explored the dynamics of energy values throughout the adversarial training process when employing SAT [9]. While training, we track both marginal energy $E_{\boldsymbol{\theta}}(\mathbf{x})$ and joint energies $E_{\boldsymbol{\theta}}(\mathbf{x}, y)$ associated with the ground truth label for both original samples and adversarial points — shown in Figs. D and E. These figures extend Fig. 1 in the paper. More precisely in Fig. D, we show a similar plot that we have in the paper but without the vector fields, thereby showing original points and adversarial points separately. In addition, to better show $E_{\boldsymbol{\theta}}(\mathbf{x})$ decreasing, in this plot, we fixed the axis to have the same numerical range that we attain at the end of the training, to notice how $E_{\boldsymbol{\theta}}(\mathbf{x})$ elongates along the diagonal component. Fig. E instead is the same Fig. 1 in the paper but with higher resolution, in addition, we offer also the same plot but color-coded with class labels rather than the drift intensity. Initially, as training commences, energy values for all data points typically initialize around zero. However, as the model progresses through successive training epochs and refines its understanding of the data, the energy values start to decline. Moreover, we observe a convergence between the values of marginal $E_{\boldsymbol{\theta}}(\mathbf{x})$ and joint energies $E_{\boldsymbol{\theta}}(\mathbf{x}, y)$, where y is the ground truth label, indicating that the model has successfully fitted these points. This means that for points around the black dashed line the CE loss is almost zero, i.e. the model pushed $p(y|x) \approx 1$ or in terms of energy $E_{\boldsymbol{\theta}}(\mathbf{x}, y) \approx E_{\boldsymbol{\theta}}(\mathbf{x})$. However, an interesting observation is that even as the model fits certain points, their energy values continue to decrease. These trends persist

[3] We refer to PGD attack maximizing Cross-Entropy loss introduced by [9] as simply PGD, while the PGD attack maximizing the KL divergence between the conditional probability distributions given original sample $\mathbf{x}$ and adversarial sample $\mathbf{x}^\star$, denoted as $p(y|\mathbf{x})$ and $p(y|\mathbf{x}^\star)$ respectively, employed by [17] as TRADES.



(1) PGD [9] **(2)** TRADES [17] **(3)** APGD [6] **(4)** APGD-T [6]

(5) APGD-DLR [6] **(6)** FAB [5] **(7)** Square [1] **(8)** CW [3]

(9) PGD [9] **(10)** TRADES [17] **(11)** APGD [6] **(12)** APGD-T [6]

(13) APGD-DLR [6] **(14)** FAB [5] **(15)** Square [1] **(16)** CW [3]

Fig. C: Top two rows (1-8). Marginal Energy distribution $E_{\boldsymbol{\theta}}(\mathbf{x})$. **(1)** PGD energy moves on the left, notice how the distributions are almost separated, the robust accuracy is 0% **(2)** TRADES performs similarly though robust accuracy is 30%; **(3)** APGD is more subtle; a tiny fraction of test points share similar values than natural data. **(4-5)** Targeted attacks such as APGD-T move energy on the right **(6)** FAB (Fast Adaptive Boundary) behaves similarly to a targeted attack. **(7-8)** Square and CW are very difficult attack since the energies overlap more, it is even visible how CW attack logic in finding the minimal deformation to flip the classification is visible in the highest overlap between energies. **Bottom two rows (9-16)** Conditional Energy distribution $E_{\boldsymbol{\theta}}(\mathbf{x}, y)$. **(9)** PGD drastically increases the $E_{\boldsymbol{\theta}}(\mathbf{x}, y)$ of the ground-truth class, thereby reducing the GT logit; **(10)** TRADES does the same but shows 2 modes, the mode on the left corresponds to points that are *not* attacked **(11-12-13)** APGD series of attacks move too $E_{\boldsymbol{\theta}}(\mathbf{x}, y)$ on the right yet making an effort to create overlap with natural distribution **(14-15-16)** FAB, Square and CW share a similar distribution that overlaps the natural ones making these attacks harder to detect. We show our analysis for a diverse set of state-of-the-art adversarial perturbations for both untargeted and targeted (-T) attacks on CIFAR-10 test set, using a non-robust model with 94.78% clean accuracy. All the attacks except for CW are produced with a deformation of input given by $\ell_\infty \leq \epsilon = 8/255$ and a step size of 2/255. The CW attack operates under an ℓ_2 perturbation constraint. ■ indicates adversarial while ■ natural data.

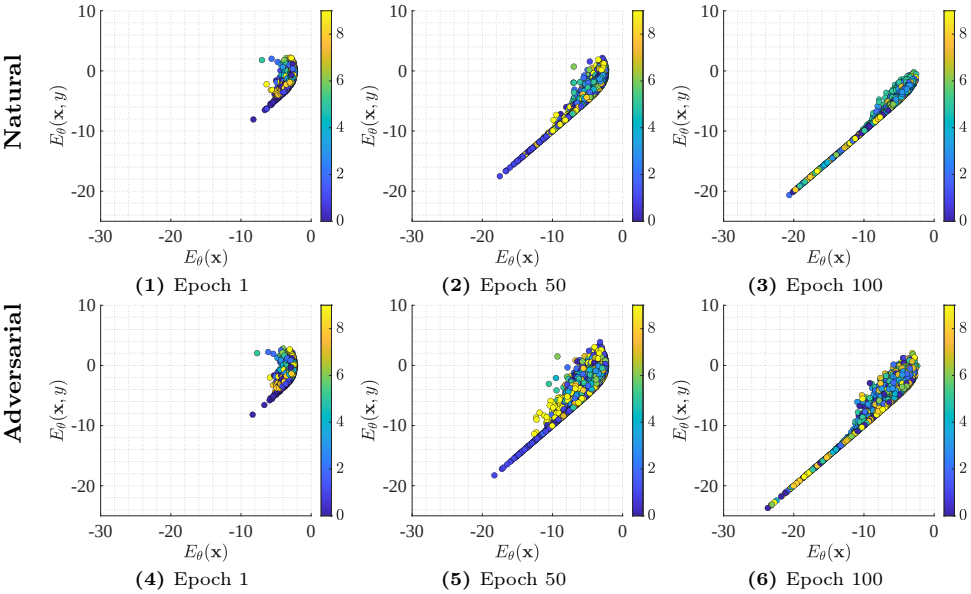


Fig. D: Scatter plots of $E_{\boldsymbol{\theta}}(\mathbf{x}, y)$ and $E_{\boldsymbol{\theta}}(\mathbf{x})$ with axis in the same range, on the CIFAR-10 dataset at various stages during training the model. **Top row (1,2,3) natural images (1)** illustrates the plots at the early stage of training and as expected, for most of the samples $E_{\boldsymbol{\theta}}(\mathbf{x}, y) > E_{\boldsymbol{\theta}}(\mathbf{x})$, indicating high loss. **(2)** showcases the plot after 50 training epochs where we can notice both $E_{\boldsymbol{\theta}}(\mathbf{x}, y)$ and $E_{\boldsymbol{\theta}}(\mathbf{x})$ have started to decrease. Finally, **(3)** shows at the 100th epoch of training, for most of the samples the $E_{\boldsymbol{\theta}}(\mathbf{x}, y)$ and $E_{\boldsymbol{\theta}}(\mathbf{x})$ have same values, indicating lower loss. From the plots, we also observe that the values for $E_{\boldsymbol{\theta}}(\mathbf{x}, y)$ and $E_{\boldsymbol{\theta}}(\mathbf{x})$ keep decreasing as we move into the later stages of the training process. **Bottom row (4,5,6) adversarial images** The trend of adversarial points is similar to what depicted in the top row yet adversarial points tend to bend the energy more and incur higher loss values.

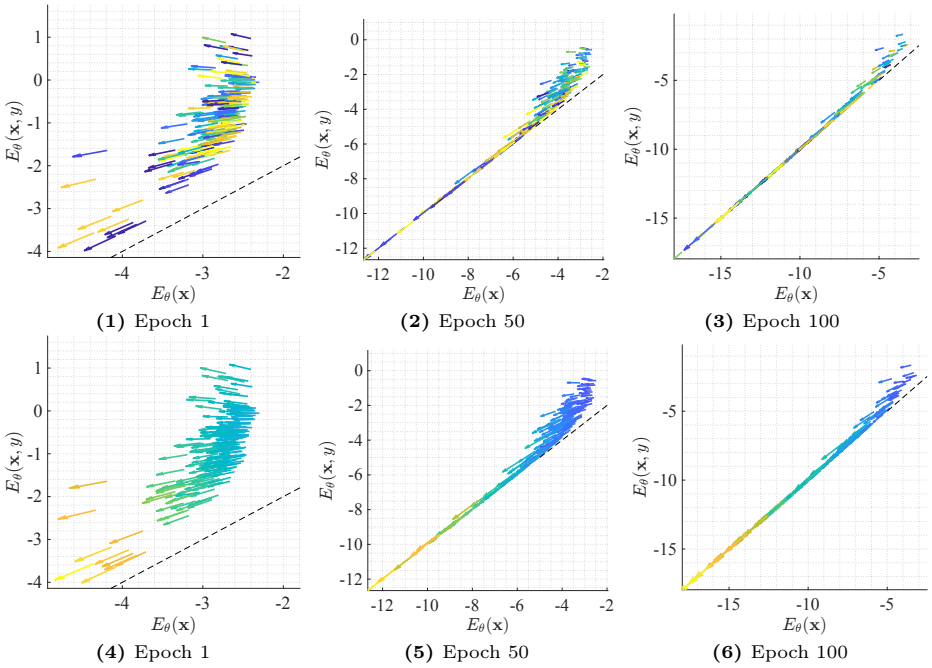


Fig. E: We scatter plot $E_{\boldsymbol{\theta}}(\mathbf{x}, y)$ in function of $E_{\boldsymbol{\theta}}(\mathbf{x})$ for a sub sample of training data of the CIFAR-10 dataset at various stages during standard AT with PGD 5 iterations at epoch 1, 50, 100. Note that the axes across figures are not in the same range for clarity. Each arrow represents the "drift" induces in the energy by the attacks: the base of the arrow is the natural data while the tip is after the attack. The dashed black, the identity line, corresponds to cross-entropy loss zero when $E_{\boldsymbol{\theta}}(\mathbf{x}, y) = E_{\boldsymbol{\theta}}(\mathbf{x})$. The plot can takes values only above that line. *Top row:* each arrow is color-coded w/ class labels: ■ airplane ■ automobile ■ bird ■ cat ■ deer ■ dog ■ frog ■ horse ■ ship ■ truck *Bottom row:* color-coded by intensity of the drift ■ less ■ more intense.

across both original and adversarial points. However, with adversarial points, we notice that the model struggles to fit a significant portion of them, and all of them being high-energy samples, located in the upper right part of the plot.

A.3 Implementation Details for Experimental Section

We train on the entire training set and select the model with the best robust accuracy under PGD on validation set, created by sampling from the synthesized images [15]. CIFAR-10/100 and Tiny-ImageNet are trained for 100 epochs while SVHN is trained for 30 epochs. We used SGD optimizer with momentum and weight decay set to 0.9 and 5×10^{-4} respectively, cyclic learning rate [11] with a maximum learning rate of 0.1. We use the ℓ_∞ threat model with $\epsilon = 8/255$, with step size α set to $2/255$ for CIFAR, Tiny-ImageNet and $1/255$ for SVHN as per standard practices. With WEAT_{ADV}, β is 6 for CIFAR-10 and SVHN, and 7 for CIFAR-100. Whereas, WEAT_{NAT} has $\beta = 6$, matching TRADES [17] for fair comparison. For MAIL-TRADES [8] using PM_{adv}, $\beta = 5$ and burn-in period is 75 epochs. In image generation, we preserve 99% of data variance, effectively guaranteeing a certain amount of starting information while minimizing high-frequency noise. Parameters such as number of SGLD steps (N), friction ζ, noise variance γ, and step size η are set to 150, 0.8, 0.001, and 0.05 respectively, with an exception of SAT [10] with $N = 20$ and $\zeta = 0.5$. With these choices, energy descent stays smooth over the generation steps, where images are projected to the range $[0, 1]$ at each iteration.

A.4 Additional Details on Experiment in Fig. 5(a)

As discussed in Section 3.2, in **"AT in function of High vs Low Energy Samples"**, we conducted a proof-of-concept experiment to better investigate the finding of MART [13], suggesting that the natural samples that are incorrectly classified contribute significantly to final robustness. Our findings revealed instead that are the high-energy samples that significantly contribute to robustness. In this section, we provide additional details on this experiment. Notably, most misclassified samples also fall into the category of high-energy samples as shown in Fig. F1. To start, we trained a robust model using SAT [9] which we used to identify correct and incorrect classifications among our training samples. We isolated 3317 (6.6% of the total samples) incorrectly classified samples and randomly sampled an equivalent number of correctly classified ones, creating two distinct datasets without these subsets, which we denote as $\mathcal{I}$ and $\mathcal{C}$, respectively.

Subsequently, we created two additional subsets, $\mathcal{L}$ and $\mathcal{H}$, this time utilizing energy values. Given that energy values are unnormalized, we found it more convenient to sort the samples based on these values and remove the 6500 samples (13% of the total samples) with the lowest energy values from the original dataset to create $\mathcal{L}$. Similarly, an equal number of samples with the highest energy values, with the condition that all samples are correctly classified, were removed from the original dataset to create $\mathcal{H}$. The thresholds for defining high and low energy

Dataset	# Correct Classified	# Incorrect Classified
High Energy Samples — $E_{\boldsymbol{\theta}}(\mathbf{x}) > -3.8744$	6500	2724
Low Energy Samples — $E_{\boldsymbol{\theta}}(\mathbf{x}) \leq -11.4755$	6500	0
Samples — $E_{\boldsymbol{\theta}}(\mathbf{x}) < -3.8744 \ \cup \ E_{\boldsymbol{\theta}}(\mathbf{x}) > -11.4755$	33683	593

Table A: It is important to clarify that the thresholds used here to classify samples as either high or low energy were automatically determined based on sizes of the selected subsets. Any sample with an energy value above -3.8744 was categorized as high energy, while those with an energy value below -11.4755 were classified as low energy.

samples were automatically determined based on the selected subset sizes. The statistics related to the original dataset with these thresholds can be seen in Tab. A. This process allowed us to generate two more datasets based on energy values. For a visual representation of how these datasets were created, please refer to Fig. F2. With four distinct datasets ($\mathcal{C}$, $\mathcal{I}$, $\mathcal{L}$, and $\mathcal{H}$) at our disposal, we trained four different models using each of these datasets. This approach facilitated a systematic examination of the influence of various sample subsets on the model's performance and robustness. The statistics of the four datasets are shown in Tab. B.

Dataset	# Correct Classified	# Incorrect Classified
$\mathcal{I}$ (w/o Incorrect)	46683	0
$\mathcal{C}$ (w/o Correct)	43366	3317
$\mathcal{H}$ (w/o High En. & Correct)	40183	3317
$\mathcal{L}$ (w/o Low Energy)	40183	3317

Table B: Summary of Datasets ($\mathcal{C}$, $\mathcal{I}$, $\mathcal{L}$, and $\mathcal{H}$) displaying the number of correctly and incorrectly classified samples within each dataset.

As shown in Fig. F3 and Fig. F4, we observe that removing incorrect samples has a significant effect on both robust and clean accuracy. They decrease robust accuracy and increase clean accuracy, whereas removing correct samples does not have much effect on either accuracy, consistent with prior knowledge. Surprisingly, we find that similar effects on accuracy can be achieved by removing just the correct samples, provided they are all high energy. Additionally, we notice that removing low energy samples has a lesser impact on both clean and robust accuracy, similar to when we randomly remove correct samples from the dataset. From this, we can deduce that the influence on accuracy is not solely determined by whether the samples are classified correctly or incorrectly, but rather by their energy levels—high energy and low energy.

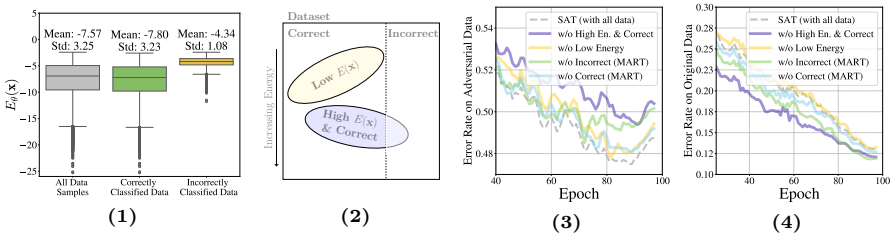


Fig. F: **(1)** Boxplots illustrating energy value distributions for all samples in the dataset, correctly classified samples, and misclassified samples. **(2)** A visual representation showing the removed subsets of data from the entire dataset. **(3)** Plots illustrating the error rates of the robust models on the adversarial **(4)** and original test samples. These models were trained on derived datasets $\mathcal{C}$, $\mathcal{I}$, $\mathcal{L}$, and $\mathcal{H}$.

A.5 Interpreting TRADES as Energy-based Model

Going beyond prior work [2, 7, 12, 18], we reinterpret TRADES objective [17] as an EBM. TRADES stands for "TRadeoff-inspired Adversarial DEfense via Surrogate-loss minimization". Given an input image $\mathbf{x}$ and Δ a feasible set of in the ℓ_p ball round $\mathbf{x}$ that is $\forall\ \mathbf{x}^\star : \mathbf{x} + \boldsymbol{\delta},\ \left\|\boldsymbol{\delta}\right\|_p < \epsilon$, a classification problem with K classes, TRADES loss is as follows:

$$\min_{\boldsymbol{\theta}} \left[\mathcal{L}_{\text{CE}}\big(\boldsymbol{\theta}(\mathbf{x}), y\big) + \beta \max_{\boldsymbol{\delta} \in \Delta} \text{KL}\left(p(y|\mathbf{x}) \middle\| p(y|\mathbf{x}^\star) \right) \right], \tag{1}$$

where $\text{KL}(\cdot, \cdot)$ is the KL divergence between the conditional probability over classes $p(y|\mathbf{x})$ that acts as reference distribution and probability over classes for generated points $p(y|\mathbf{x}^\star)$, the loss $\mathcal{L}$ is CE loss and $p(y|\mathbf{x})$ is given by the softmax applied to the logits $\boldsymbol{\theta}(\mathbf{x})$.

Proposition 1. *The KL divergence between two discrete distributions $p(y|\mathbf{x})$ and $p(y|\mathbf{x}^\star)$ can be interpreted as EBM as:*

$$\underbrace{\mathbb{E}_{k \sim p(y|\mathbf{x})}\Big[E_{\boldsymbol{\theta}}(\mathbf{x}^\star, k) - E_{\boldsymbol{\theta}}(\mathbf{x}, k) \Big]}_{\text{conditional term weighted by classifier prob.}} + \underbrace{E_{\boldsymbol{\theta}}(\mathbf{x}) - E_{\boldsymbol{\theta}}(\mathbf{x}^\star)}_{\text{marginal term}} \tag{2}$$

Proof. KL divergence is defined as:

$$\begin{aligned} KL(P||Q) = \sum_{k \in K} p(k|\mathbf{x}) \log \left(\frac{p(k|\mathbf{x})}{p(k|\mathbf{x}^\star)} \right) = \\ = \sum_{k \in K} p(k|\mathbf{x}) \log \big(p(k|\mathbf{x}) \big) - \sum_{k \in K} p(k|\mathbf{x}) \log \big(p(k|\mathbf{x}^\star) \big). \end{aligned} \tag{3}$$

Now recalling that the $\log\big(p(k|\mathbf{x})\big)$ can be written in terms of energies as $\log\big(p(k|\mathbf{x})\big) = -E_{\boldsymbol{\theta}}(\mathbf{x}, k) + E_{\boldsymbol{\theta}}(\mathbf{x})$, noting that $\sum_{k\in K} p(k|\mathbf{x})$ is one and $E_{\boldsymbol{\theta}}(\mathbf{x})$ does not depend on k, then we have that:

$$\sum_{k\in K} p(k|\mathbf{x}) \log\big(p(k|\mathbf{x})\big) = \sum_{k\in K} p(k|\mathbf{x})\Big[- E_{\boldsymbol{\theta}}(\mathbf{x}, k) + E_{\boldsymbol{\theta}}(\mathbf{x})\Big] = \\ = E_{\boldsymbol{\theta}}(\mathbf{x}) + \sum_{k\in K} p(k|\mathbf{x})\Big[- E_{\boldsymbol{\theta}}(\mathbf{x}, k)\Big].$$

Thus Eq. (3) can be written shortly as:

$$\mathrm{KL}\Big(p(y|\mathbf{x})\Big\|p(y|\mathbf{x}^\star)\Big) \doteq E_{\boldsymbol{\theta}}(\mathbf{x}) - E_{\boldsymbol{\theta}}(\mathbf{x}^\star) + \sum_{k\in K} p(k|\mathbf{x})\Big[E_{\boldsymbol{\theta}}(\mathbf{x}^\star, k) - E_{\boldsymbol{\theta}}(\mathbf{x}, k)\Big].$$

So the KL loss minimizes two terms:

$$\underbrace{\mathbb{E}_{k\sim p(y|\mathbf{x})}\Big[E_{\boldsymbol{\theta}}(\mathbf{x}^\star, k) - E_{\boldsymbol{\theta}}(\mathbf{x}, k)\Big]}_{\text{conditional term weighted by classifier prob.}} + \underbrace{E_{\boldsymbol{\theta}}(\mathbf{x}) - E_{\boldsymbol{\theta}}(\mathbf{x}^\star)}_{\text{marginal term}} \ \square \tag{4}$$

Corollary 1. *TRADES object can be written as EBM as:*

$$E_{\boldsymbol{\theta}}(\mathbf{x}, y) + (\beta - 1)E_{\boldsymbol{\theta}}(\mathbf{x}) - \beta\Big\{E_{\boldsymbol{\theta}}(\mathbf{x}^\star) + \mathbb{E}_{p(y|\mathbf{x})}\Big[E_{\boldsymbol{\theta}}(\mathbf{x}, k) - E_{\boldsymbol{\theta}}(\mathbf{x}^\star, k)\Big]\Big\}. \tag{5}$$

Proof. It follows from combining Proposition 1 and CE loss applied to natural data but written as EBM. It follows from just rearranging the terms and combining the $E_{\boldsymbol{\theta}}(\mathbf{x})$ part from KL divergence w.r.t. to the CE loss.

$$\mathcal{L}_{\mathrm{CE}}\big(\boldsymbol{\theta}(\mathbf{x}), y\big) + \beta\,\mathrm{KL}\Big(p(y|\mathbf{x})\Big\|p(y|\mathbf{x}^\star)\Big),$$

$$E_{\boldsymbol{\theta}}(\mathbf{x}, y) - E_{\boldsymbol{\theta}}(\mathbf{x}) + \beta\Big\{E_{\boldsymbol{\theta}}(\mathbf{x}) - E_{\boldsymbol{\theta}}(\mathbf{x}^\star) + \mathbb{E}_{p(y|\mathbf{x})}\Big[E_{\boldsymbol{\theta}}(\mathbf{x}^\star, k) - E_{\boldsymbol{\theta}}(\mathbf{x}, k)\Big]\Big\},$$

$$E_{\boldsymbol{\theta}}(\mathbf{x}, y) + (\beta - 1)E_{\boldsymbol{\theta}}(\mathbf{x}) - \beta\Big\{E_{\boldsymbol{\theta}}(\mathbf{x}^\star) + \mathbb{E}_{p(y|\mathbf{x})}\Big[E_{\boldsymbol{\theta}}(\mathbf{x}, k) - E_{\boldsymbol{\theta}}(\mathbf{x}^\star, k)\Big]\Big\}\ \square.$$

Our formulation can also better explain why the samples that the model fit well, referred to low-loss data lead to robust overfitting [16]. Usually $\beta = \{1, 6\}$, following Eq. (5), when $\beta = 1$, then we have:

$$E_{\boldsymbol{\theta}}(\mathbf{x}, y) - E_{\boldsymbol{\theta}}(\mathbf{x}^\star) - \mathbb{E}_{p(y|\mathbf{x})}\Big[E_{\boldsymbol{\theta}}(\mathbf{x}, k) - E_{\boldsymbol{\theta}}(\mathbf{x}^\star, k)\Big]$$

which means we do not consider the marginal energy of the natural data. Moreover, in the later phase of training, TRADES resembles more SAT, assuming k is the index of most likely class with high confidence and k matches the ground-truth label y, then Eq. (4) approximately becomes:

$$\cancel{E_{\boldsymbol{\theta}}(\mathbf{x}, y)} - E_{\boldsymbol{\theta}}(\mathbf{x}^\star) - \Big[\cancel{E_{\boldsymbol{\theta}}(\mathbf{x}, y)} - E_{\boldsymbol{\theta}}(\mathbf{x}^\star, y)\Big], \tag{6}$$

given that when the model is well trained the expectation acts more like a one-hot encoding thereby selecting the ground-truth class. By rearranging the terms, Eq. (6) becomes:

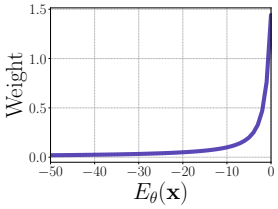
$$E_{\boldsymbol{\theta}}(\mathbf{x}^\star, y) - E_{\boldsymbol{\theta}}(\mathbf{x}^\star) = \mathcal{L}_{\mathrm{CE}}(\mathbf{x}^\star, y; \boldsymbol{\theta}).$$

and hence with $\beta = 1$ and under the assumptions stated before, towards the end of the training, TRADES, e.g. Eq. (5), precisely resembles the outer minimization objective of SAT [9] which has been seen to exhibit severe overfitting.

A.6 Weighted Energy Adversarial Training (WEAT) algorithm

Based on our several observations from Section 3.2, **"How Adversarial Training Impacts the Energy of Samples"**, we propose a novel weighting scheme, Weighted Energy Adversarial Training (WEAT). The core of the WEAT lies in its weighting function, which assigns higher weights to samples with higher energy and lower weights to the samples with low energy.

Since energy is unnormalized, finding an appropriate weighting function can be challenging. Throughout our preliminary experimentation, it became evident that the marginal energy values for all samples predominantly reside in the negative range, with the highest values observed not surpassing zero. Therefore, we found that a function shown here on the right yielded the most favorable results: it assigns higher weights to samples around zero and non-linearly decreases the weights as it moves away from zero. Our weighting function $w(\mathbf{x})$ is defined as:



$$w(\mathbf{x}) = \frac{1}{\log\left(1 + \exp(|E_{\boldsymbol{\theta}}(\mathbf{x})|)\right)}. \tag{7}$$

Finally, we present the WEAT method in Algorithm 1.

A.7 Additional Details on the Generative Capabilities

Hyperparameters Choice. In the section outlining our approaches, we presented two distinct models, both of which emerged as our best performers, employing the same inference method but built on different architectures. The first model is rooted in SAT [10], while the second one is constructed based on the principles outlined in Better DM [15]. Better DM uses TRADES for training and employs millions of synthetic images generated by diffusion models. Despite utilizing the same inference method, the primary distinction lies in the choice of hyperparameters, which are determined based on their respective capabilities in terms of generation intensity.

As asserted in the section discussing model's generation capabilities, we observed that SAT's generative intensity is more pronounced. In the process of generating images, each iteration contributes with a significantly informative

Algorithm 1: Weighted Energy Adversarial Training (WEAT)

Input and parameters: Dataset $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, Batch size m, Number of epochs T, Number of steps for perturbation method s, Learning rate η, perturbation function P [17], KL-Divergence function KL.
Output: Adversarially Robust Network $\boldsymbol{\theta}$
Initialize model parameters $\boldsymbol{\theta}$
for $t = 1$ *to* T **do**
  for *each mini-batch* $(\mathbf{x}_b, y_b)$ *in* D **do**
    Generate perturbed examples: $\mathbf{x}_b^\star = P(\mathbf{x}_b, s, \theta)$
    Compute Energy: $E_\theta(\mathbf{x}_b)$, and detach it from computational graph
    Compute weights vector as Eq. (7): $w(\mathbf{x}_b) = 1/\log(1 + \exp(|E_\theta(\mathbf{x}_b)|))$
    `Note that the` $w(\mathbf{x}_b)$ `is computed on original points.`
    if $WEAT_{adv}$ **then**
      $\mathcal{L}_{\text{CE}} = \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}_{\text{CE}}(\boldsymbol{\theta}(\mathbf{x}_b^\star), y_b) \odot w(\mathbf{x}_b)$
    end
    else if $WEAT_{nat}$ **then**
      $\mathcal{L}_{\text{CE}} = \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}_{\text{CE}}(\boldsymbol{\theta}(\mathbf{x}_b), y_b) \odot w(\mathbf{x}_b)$
    end
    $\mathcal{L}_{\text{KL}} = \frac{1}{m}\sum_{i=1}^{m} (KL(\boldsymbol{\theta}(\mathbf{x}_b), \boldsymbol{\theta}(\mathbf{x}_b^\star)) \odot w(\mathbf{x}_b))$
    Compute total loss: $\mathcal{L}_{\text{total}} : \mathcal{L}_{\text{CE}} + \beta \cdot \mathcal{L}_{\text{KL}}$
    Update model parameters: $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{\text{total}}$
  end
end

content, reducing the necessity for multiple iterations. However, the robust model incorporates image components distinguished by sharply defined contours and vibrant colors. If these features are added for too many iterations, they can lead to the generation of unrealistic images that deviate from the underlying manifold and amplifies significant traits of the class. For this reason, the number of SGLD iterations is well calibrated as well as the momentum friction—see Tab. C—which is set to a smaller constant to prevent excessive speed in the SGLD dynamics, avoiding the generation of excessively bright, sharp and unrealistic images. An example of generations from the model is given in Fig. G.

Parameter	**Better DM** [15]	**SAT** [10]
SGLD steps (N)	150	20
Friction (ζ)	0.8	0.5
Step size (η)	0.05	0.05
Noise variance (γ)	0.001	0.001

Table C: Parameters for SAT's and Better DM's Model Generation

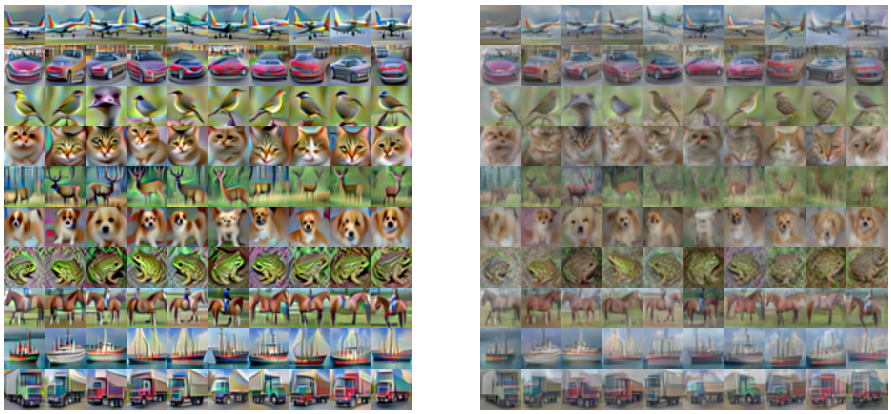


Fig. G: (Left) Generated images using SAT [10] and with parameters chosen for BetterDM [15]: images have saturated colors and class features are exaggerated. **(Right)** Inference from SAT [10] with parameters tuning: the colors and subject contours better match the distribution of natural images.

On the contrary, the intensity of the generation of other models trained with TRADES, e.g. Better DM [15], is less pronounced. These models do have generative capabilities but the generation is less intense and more "smooth". Their contributions at each step are more subdued and less sharp, both in terms of color and shape. Consequently, the generation procedure for these models was calibrated differently, employing more steps and introducing more friction in the momentum. The inference configuration of hyperparameters for our best, Better DM [15], is reported in Tab. C. In particular, we display synthesized samples for our best performing model in the following sections, giving an extensive qualitative evaluation of its generation capability considering it is only a classifier.

Additional Generated Samples. In Fig. H and Fig. I, we present 100 generated samples for each class from the top-performing model [15]. This section provides an expanded set of images for a more in-depth qualitative analysis.

We additionally employ the Structural Similarity Index [14] to assess the comparison between the generated images and samples extracted from the CIFAR-10 test set. This comparison involves evaluating the similarity between the synthesized images and the in-distribution samples, which are real images not included in the training set, for a better qualitative evaluation. The results of this comparison are depicted in Fig. J.

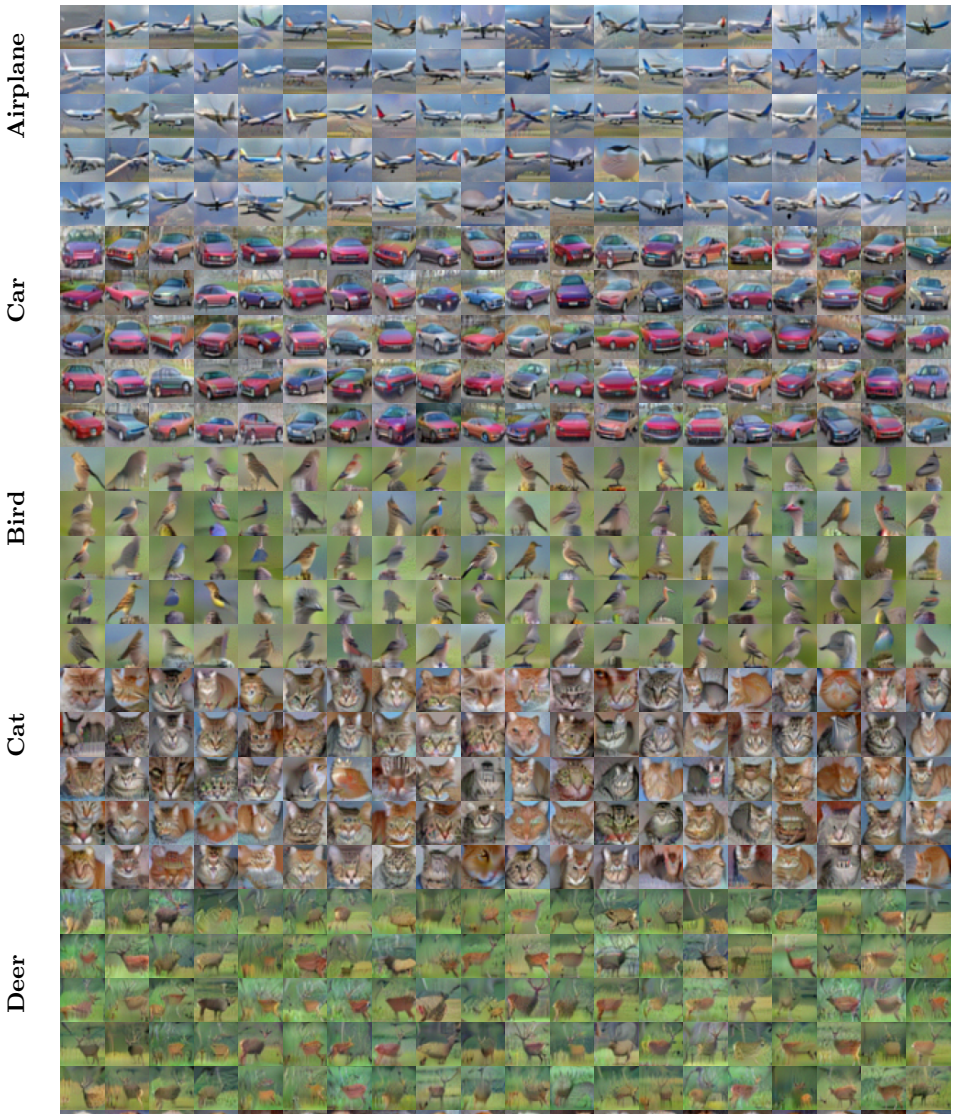


Fig. H: Generated class-conditional samples of CIFAR-10. Each subfigure corresponds to samples belonging to a specific class.

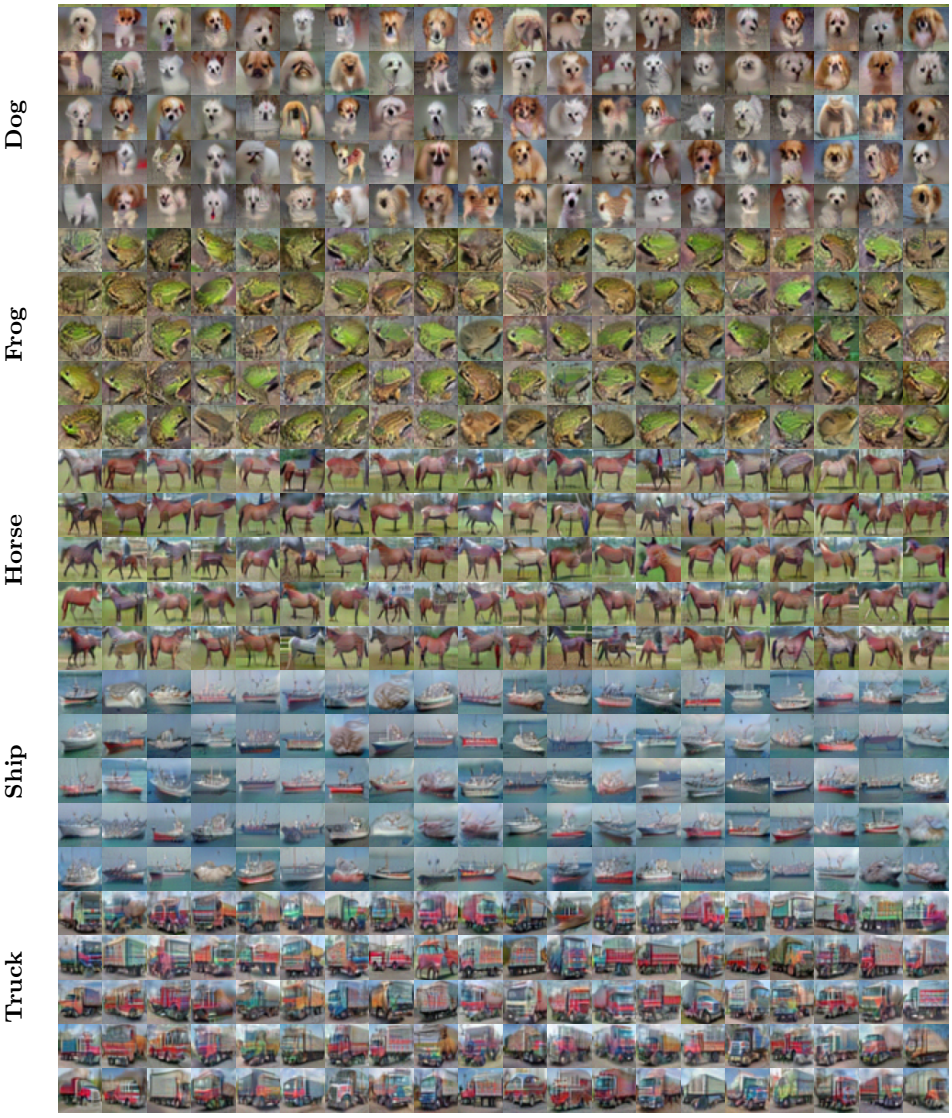


Fig. I: Generated class-conditional samples of CIFAR-10. Each subfigure corresponds to samples belonging to a specific class.

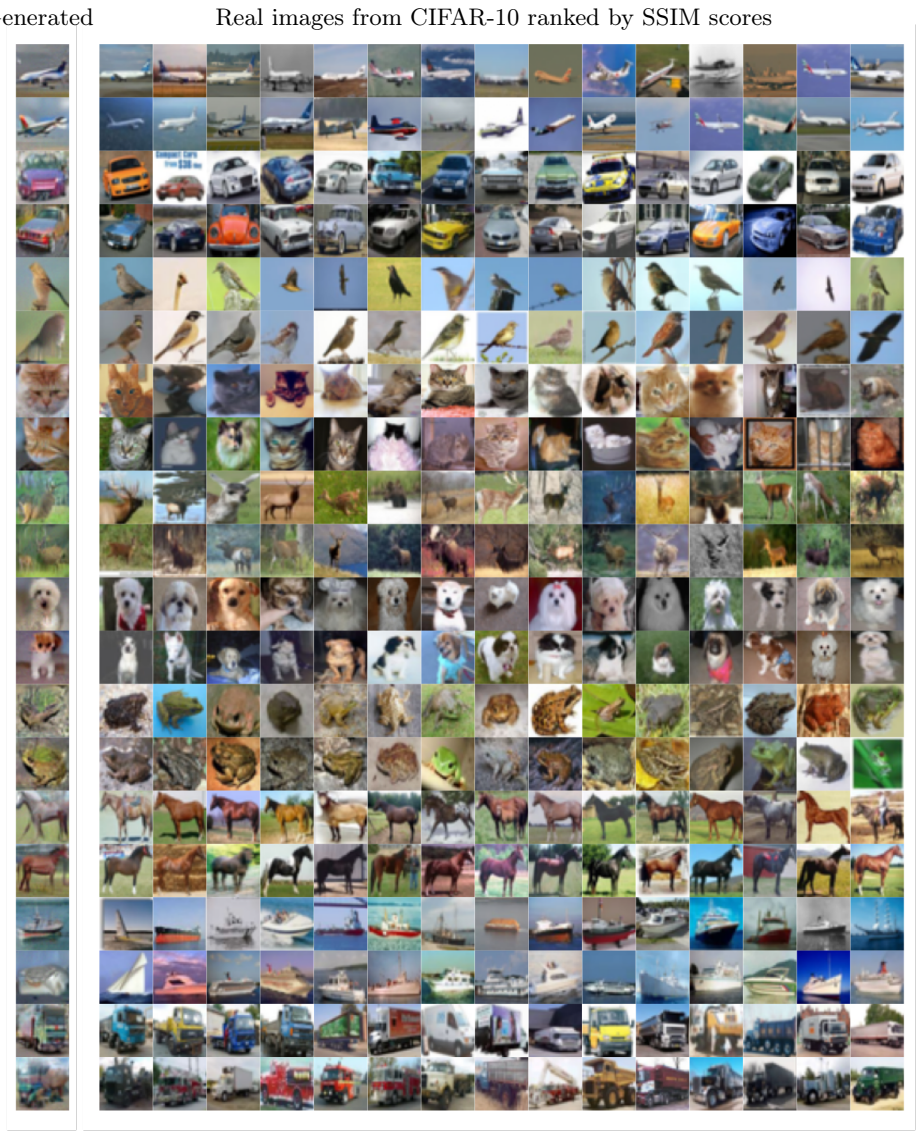


Fig. J: In this plot we show a qualitative comparison between some generated samples, shown in the left column, and fifteen images belonging to CIFAR-10 test set that showed the fifteen greatest SSIM scores.

References

1. Andriushchenko, M., Croce, F., Flammarion, N., Hein, M.: Square attack: a query-efficient black-box adversarial attack via random search. In: ECCV (2020)
2. Beadini, S., Masi, I.: Exploring the connection between robust and generative models. In: Italian Conference on AI - Ital-IA - Workshop on AI for Cybersecurity (2023)
3. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: IEEE Symposium on Security and Privacy (SP) (2017)
4. Croce, F., Andriushchenko, M., Sehwag, V., Debenedetti, E., Flammarion, N., Chiang, M., Mittal, P., Hein, M.: Robustbench: a standardized adversarial robustness benchmark. In: ICLR Workshops 2021 - Workshop on Security and Safety in Machine Learning Systems (2021)
5. Croce, F., Hein, M.: Minimally distorted adversarial examples with a fast adaptive boundary attack. In: ICML (2020)
6. Croce, F., Hein, M.: Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In: ICML (2020)
7. Grathwohl, W., Wang, K.C., Jacobsen, J.H., Duvenaud, D., Norouzi, M., Swersky, K.: Your classifier is secretly an energy based model and you should treat it like one. In: ICLR (2020)
8. Liu, F., Han, B., Liu, T., Gong, C., Niu, G., Zhou, M., Sugiyama, M., et al.: Probabilistic margins for instance reweighting in adversarial training. In: NeurIPS (2021)
9. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: ICLR (2018)
10. Santurkar, S., Tsipras, D., Tran, B., Ilyas, A., Engstrom, L., Madry, A.: Image synthesis with a single (robust) classifier. In: NeurIPS (2019)
11. Smith, L.N., Topin, N.: Super-convergence: Very fast training of neural networks using large learning rates. In: Artificial intelligence and machine learning for multi-domain operations applications. vol. 11006, pp. 369–386. SPIE (2019)
12. Wang, Y., Wang, Y., Yang, J., Lin, Z.: A unified contrastive energy-based model for understanding the generative ability of adversarial training. In: ICLR (2022)
13. Wang, Y., Zou, D., Yi, J., Bailey, J., Ma, X., Gu, Q.: Improving adversarial robustness requires revisiting misclassified examples. In: ICLR (2020)
14. Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.: Image quality assessment: From error visibility to structural similarity. IEEE Transactions on Image Processing (2004)
15. Wang, Z., Pang, T., Du, C., Lin, M., Liu, W., Yan, S.: Better diffusion models further improve adversarial training. In: ICML (2023)
16. Yu, C., Han, B., Shen, L., Yu, J., Gong, C., Gong, M., Liu, T.: Understanding robust overfitting of adversarial training and beyond. In: ICML (2022)
17. Zhang, H., Yu, Y., Jiao, J., Xing, E.P., Ghaoui, L.E., Jordan, M.I.: Theoretically principled trade-off between robustness and accuracy. In: ICML (2019)
18. Zhu, Y., Ma, J., Sun, J., Chen, Z., Jiang, R., Chen, Y., Li, Z.: Towards understanding the generative capability of adversarially robust classifiers. In: ICCV (2021)