A. Derivations

Proof of Lemma 6.3. Recall that our goal is to solve the constrained optimization problem

$$\min\{|\delta|_p \,|\, \delta \in \mathbb{R}^d, w^T(x + U\delta) + b = 0\} \tag{A.1}$$

Using the method of Lagrange multipliers, we know that a minimizer $\delta \in \mathbb{R}^d$ must satisfy the critical point condition

$$\lambda U^T w \in \partial|\delta|_p \tag{A.2}$$

where $\partial|\delta|_p$ is the subdifferential of the ℓ^p-norm at $\delta \in \mathbb{R}^d$. It is a classical fact (see e.g. [1, Prop. 1.2]) that

$$\partial|\delta|_p = \begin{cases} \{v \in \mathbb{R}^d \,|\, |v|_q \leq 1\} & \text{if } \delta = 0 \\ \{v \in \mathbb{R}^d \,|\, |v|_q = 1 \text{ and } v^T\delta = |\delta|_p\} & \text{otherwise.} \end{cases} \tag{A.3}$$

If the first case occurs, clearly the minimum of Eq. (A.1) is 0, and since $\delta = 0$ we obtain

$$0 = w^T(x + U\delta) + b = w^T x + b, \tag{A.4}$$

hence in this case $\frac{|w^T x + b|}{|w^T U|_q} = 0$ as well and the lemma holds.

In the case where $\delta \neq 0$, combining Eqs. (A.2) and (A.3) we see that for some λ

$$|\lambda U^T w|_q = 1 \text{ and } \lambda w^T U\delta = |\delta|_p; \tag{A.5}$$

from the first equation we immediately identify $|\lambda| = |U^T w|_q^{-1}$, and taking absolute values on both sides of the second then gives

$$\begin{aligned} |\delta|_p &= |\lambda||w^T U\delta| \\ &= \frac{|w^T U\delta|}{|U^T w|_q}. \end{aligned} \tag{A.6}$$

Finally, recalling $w^T(x + U\delta) + b = 0$ gives $|w^T U\delta| = |w^T x + b|$, completing the proof. □

B. Experimental details

B.1. Model architectures and training details

For our MNIST experiments we use the simple 2 layer convolutional network of [23] — we ported the TensorFlow code available at https://github.com/MadryLab/mnist_challenge to PyTorch [27]. We train it using SGD with momentum 0.9, batch size 1024 and weight decay 10^{-4} for 100 epochs, with initial learning rate 10^{-3} and learning rate drops by a factor of 0.1 whenever validation accuracy doesn't improve by 1% for 10 epochs. We save the weights with the best validation accuracy ($\approx 98.95\%$).

For our CIFAR10 experiments we use the ResNet9 from MosaicML's Composer library [34]. We train it with SGD with momentum 0.9, batch size 512 and weight decay 10^{-4} for 160 epochs, with initial learning rate 10^{-1} and learning rate drops by a factor of 0.1 whenever validation accuracy doesn't improve by 1% for 10 epochs. We save the weights with the best validation accuracy ($\approx 91.72\%$).

For our ImageNet experiments we use the ResNet50 from TorchVision [24]. We train it with SGD with momentum 0.9 and weight decay 10^{-4} for 100 epochs, with initial learning rate 1.0 and learning rate drops by a factor of 0.1 whenever validation accuracy doesn't improve by 1% for 10 epochs. Due to distributed data parallel training with batches of size 512 on each of 8 GPUs, our effective batch size is $8 \cdot 512 = 4096$. We save the weights with the best validation accuracy ($\approx 72.84\%$).

B.2. Tuning PGD step sizes

In our experiments we generate a large number of PGD adversarial examples for a wide range of perturbation constraints ϵ and in subspaces of varying dimension. In order for our numerical experiments to address our questions about the behavior of $\text{success}(d, \epsilon)$, it is *crucial* that our PGD algorithm for optimizing δ has the capacity to achieve the boundary case $|\delta|_p = \epsilon$. We found that with some standard choices of step size, this did not occur, resulting in an unpleasant situation where the effective budget was significantly lower than ϵ simply due to a too-small PGD step size. Here we briefly discuss a principled choice of PGD step size that accounts for the dimension d of the subspace to which δ is constrained. First we must specify the PGD algorithm being used.

Our basic PGD implementation (adapted from [23]) iterates the following: we constrain δ to a d-dimensional subspace $V \subseteq \mathcal{X}$ using an isometry $U : \mathbb{R}^d \to V$ as in Sec. 6, and initialize $\delta_0 = 0$. Then, for $t = 1, \ldots, T$ where T is the maximum number of steps, we let $g_t = \nabla_\delta \ell(f(x + U\delta), y)$ where ℓ is cross entropy, and replace it with the "normalized" gradient

$$\tilde{g}_t := \begin{cases} \frac{g_t}{|g_t|_p}, & p \in \{1, 2\} \\ \operatorname{sign} g_t, & p = \infty. \end{cases} \tag{B.1}$$

We then project $\delta_{t-1} + \eta\tilde{g}_t$, where η is a learning rate, onto the ℓ^p ϵ-ball centered at 0 to obtain in the case $p \in \{1, 2\}$

$$\delta_t = \begin{cases} \epsilon\frac{\delta_{t-1} + \eta\tilde{g}_t}{|\delta_{t-1} + \eta\tilde{g}_t|_p}, & |\delta_{t-1} + \eta\tilde{g}_t|_p > \epsilon \\ \delta_{t-1} + \eta\tilde{g}_t, & \text{otherwise} \end{cases} \tag{B.2}$$

and in the case $p = \infty$

$$\delta_t = \begin{cases} \epsilon\frac{\delta_{t-1} + \eta\tilde{g}_t}{\max\{\delta_{t-1} + \eta\tilde{g}_t\}}, & \max\{\delta_{t-1} + \eta\tilde{g}_t\} > \epsilon \\ \delta_{t-1} + \eta\tilde{g}_t, & \text{otherwise.} \end{cases} \tag{B.3}$$

Finally, we must ensure that $x + U\delta$ lies in the image hypercube $[0, 1]^{C \times H \times W}$ (in our implementation pixel values lie

in $[0,1]$). To do this, we let $\text{clip} : \mathbb{R} \to [0,1]$ be the clipping function, i.e. $\text{clip}(x) = \max\{0, \min\{x, 1\}\}$, and replace δ with

$$U^T(\text{clip}(x + U\delta) - x) \tag{B.4}$$

(here we use the fact that U is orthogonal and so U^T is a left inverse for U).

To set the step size η, we can adopt the heuristic that the δ_t behave like a random walk, i.e. that the normalized gradients $\tilde{g}_t$ are sampled IID from some distribution (this almost certainly quite false, but we found it to be useful in a back-of-the-envelope sort of way). We will even further assume that for each t the coordinates of $\tilde{g}_t$ are IID. Ignoring projection and clipping, we have $\delta_T = \eta \sum_{t=1}^T \tilde{g}_t$. In the case $p = 2$, using the supposed IID-ness we see that

$$E[|\delta|_2^2] = \eta^2 \sum_{t=1}^T |\tilde{g}_t|_2^2 = \eta^2 T. \tag{B.5}$$

Since we want to ensure the left hand side is at least ϵ, we obtain the step size

$$\eta = \frac{\epsilon}{\sqrt{T}}. \tag{B.6}$$

In practice we multiply the above by 2 for good measure.[8] In the case $p = \infty$, by our IID-ness assumptions and the fact that by definition $\tilde{g}_t = \text{sign}\, g_t$, the individual coordinates $\tilde{g}_{t,j}$ for $j = 1, \ldots, d$ are IID samples from $\{\pm 1\}$ and so

$$|\delta_T|_\infty = \eta\sqrt{T} \max_{j=1,\ldots,d} \{|\frac{1}{\sqrt{T}} \sum_{t=1}^T \tilde{g}_{t,j}|\} \tag{B.7}$$

The distribution of each term $\frac{1}{\sqrt{T}} \sum_{t=1}^T \tilde{g}_{t,j}$ tends towards a Gaussian distribution with mean 0 and variance 1 by the central limit theorem. — letting $\Phi(x)$ be the standard normal CDF, the CDF of each $|\frac{1}{\sqrt{T}} \sum_{t=1}^T \tilde{g}_{t,j}|$ is approximated by

$$F(x) := \Phi(x) - \Phi(-x). \tag{B.8}$$

By [2], the distribution of the max occuring in Eq. (B.7) is concentrated around the $\frac{d-1}{d}$-th quantile of the CDF Eq. (B.8), i.e. $F^{-1}(1 - \frac{1}{d})$. Assuming d is relatively large, so that $1 - \frac{1}{d}$ is near 1, we ignore the $\Phi(-x)$ term in Eq. (B.8) for the purposes of inversion and get

$$F^{-1}(1 - \frac{1}{d}) \approx \Phi^{-1}(1 - \frac{1}{d}) \tag{B.9}$$

Recall that our objective is to ensure that $|\delta_T|_\infty \geq \epsilon$. By the above arguments, this translates to

$$\begin{aligned} &\epsilon \leq \eta\sqrt{T}\Phi^{-1}(1 - \frac{1}{d}), \text{ i.e.} \\ &\eta \approx \frac{\epsilon}{\sqrt{T}\Phi^{-1}(1 - \frac{1}{d})}. \end{aligned} \tag{B.10}$$

[8] Note that this is *larger* than what is suggested in [23, p. 12, section "Resistance for different values ..."], which divides by T. By "for good measure" we mean that our primary concern is using *too small* of a step size.

Again, in practice we multiply by 2 for good measure. Observe that while our ℓ^2 step size is independent of d, Eq. (B.10) does depend on d. In fact, as $d \to \infty$ the step size goes to 0, but very slowly (for $d = 3 \cdot 224^2$, the dimension of ImageNet images, $\Phi^{-1}(1 - \frac{1}{d}) \approx 4.36$).

For the $p = 1$ case, we use a heuristic similar to that of $p = 2$ above; explicitly, we set

$$\eta = \sqrt{2\pi}\frac{\epsilon}{\sqrt{T}}$$

B.3. Adversarial example generation

For each dataset and model, we select a range of perturbation budgets ϵ and subspace dimensions d, in both cases logarithmically spaced between minimum and maximal values of ϵ and d, with as many grid points as we can afford (for MNIST and CIFAR, 32 different values of each, for ImageNet only 8 of each).

For each pair (ϵ, d) we loop over the entire validation set of the relevant dataset, with the exception of ImageNet where we randomly sample 10,000 of the 50,000 images. We randomly sample a distinct subspace $V_i \subset \mathcal{X}$ for each validation datapoint (x_i, y_i) (as above, by randomly generating a matrix U whose columns span V). We then loop through validation datapoints (x_i, y_i) and corresponding matrices U_i and compute PGD adversarial examples as described above, with $T = 16$ steps. We compute the error over the validation set (subsampled in the case of ImageNet), i.e.

$$1 - \frac{1}{N} \sum_{i=1}^N \mathbf{1}(f(x_i + U\delta_i) = y_i).$$

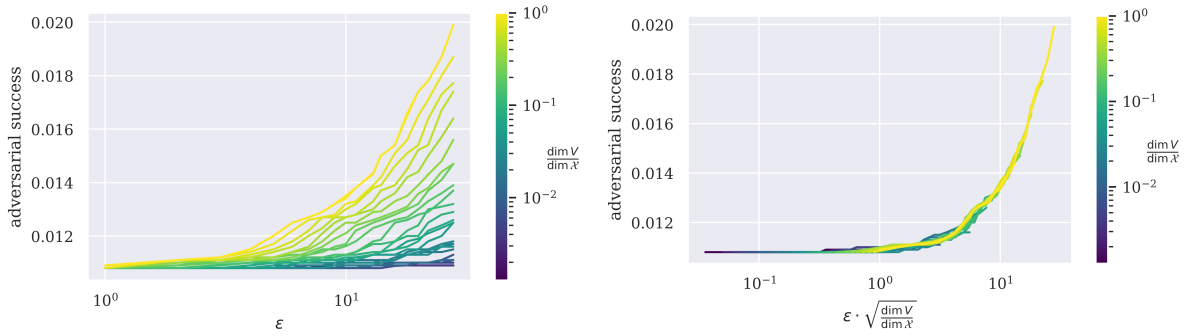
Remark B.11. We intended to permute the matrices U_i across several runs through the validation dataset $\{(x_i, y_i)\}$ to provide error bars with respect to the distribution of subspaces. Due to a coding oversight, they didn't actually get permuted. For this reason our plots lack confidence intervals, however we hope that the tight agreement of all curves after our x-axis reparametrization illustrates that such error bars would be quite small.

B.4. Subspaces sampled uniformly from the Grassmannian

In the case of MNIST and for $p = 2$, we can also sample subspaces uniformly from the Grassmannian $\text{Gr}(d, \dim \mathcal{X})$ by sampling matrices U of shape $n \times d$ with orthonormal columns using the QR decomposition as in used in the method `scipy.stats.ortho_group` of [36]. The results, shown in Fig. 3, are similar to those in Figs. 5a and 5b.

B.5. Analysis of the 1-norm case

When $p = 1$, $q = \infty$ and so the arguments used in Eqs. (6.6) and (6.7) do not make sense as written; more-



(a) (b)

Figure 3. **(a)** Projected gradient descent (PGD) adversarial examples for the 2-layer convolutional neural network of [23] on the MNIST validation set. In these experiments we generated the subspaces V by sampling uniformly from the Grassmannian, and constrain perturbations using the ℓ^2 norm. Error bars represent standard deviations of 5 passes over the MNIST validation set. **(b)** The same data, reparametrized by plotting $\epsilon \cdot \sqrt{\frac{\dim V}{\dim \mathcal{X}}}$ along the x-axis. *Note*: x-axes are log-scale.

over, while we have not explicitly verified this, it seems that attempting to take a limit of those equations as $q \to \infty$ one will encounter an "∞/∞" case, and it's not clear that e.g. L'Hospital's rule helps at all.

Instead, we propose a different estimate of the quotient

$$\frac{|w^T U|_\infty}{|w|_\infty}, \tag{B.12}$$

proceeding as follows. We will again assume, as is the case in our experiments, that U is obtained by subsampling basis vectors, say $\{e_{i_1}, \ldots, e_{i_d}\}$. Then

$$\frac{|w^T U|_\infty}{|w|_\infty} = \frac{\max_j\{|w_{i_j}|\}}{\max_i\{|w_i|\}} \tag{B.13}$$

The question, then, is how much smaller the $\max$ over a random $\dim V$-element subset of the absolute values $|w_i|$ is than the $\max$ over all $\dim \mathcal{X}$ of them. The need to make some assumption on the distribution the $|w_i|$ are drawn from seems unavoidable at this point: we suppose the coefficients w_i come from a standard normal distribution, so that their absolute values come from a "half-normal" (equivalently χ_1) distribution: if Φ is the standard normal cumulative distribution function, with this assumption the cumulative distribution function of the $|w_i|$ is

$$F(x) := \Phi(x) - \Phi(-x). \tag{B.14}$$

We make a further crude estimate that the numerator and denominator are $\max$s of *independent* samples of size $\dim V$ and $\dim \mathcal{X}$ respectively;[9] then the theory of quantiles in large samples [2] suggests the estimates

$$\begin{aligned} \max_j\{|w_{i_j}|\} &\approx F^{-1}(1 - \frac{1}{\dim V}) \text{ and} \\ \max_i\{|w_i|\} &\approx F^{-1}(1 - \frac{1}{\dim \mathcal{X}}) \end{aligned} \tag{B.15}$$

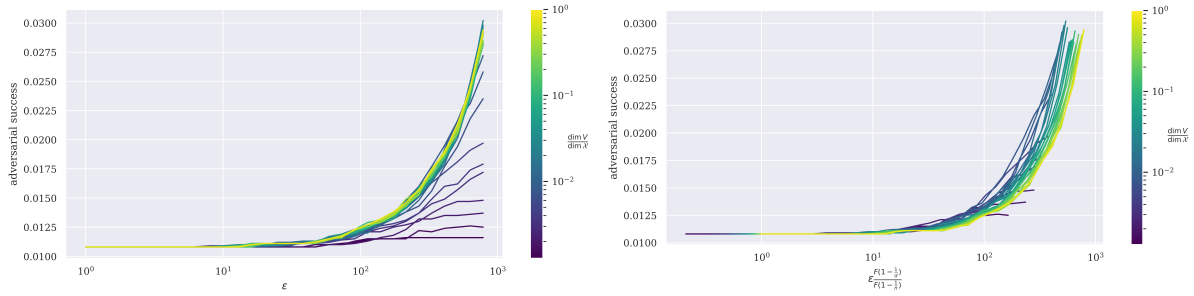
leading to the overall estimate

$$\frac{|w^T U|_\infty}{|w|_\infty} \approx \frac{F^{-1}(1 - \frac{1}{\dim V})}{F^{-1}(1 - \frac{1}{\dim \mathcal{X}})} \tag{B.16}$$

Figure 4 shows the result of using Eq. (B.16) as a stand-in for $(\frac{\dim V}{\dim \mathcal{X}})^{\frac{1}{q}}$ in the case where $p = 1$, on the MNIST dataset. One immediate observation is that at least for $\dim V$ a significant fraction of $\dim \mathcal{X}$ (e.g. $\frac{\dim V}{\dim \mathcal{X}} \geq 0.1$) it does appear that the curves $\text{success}(\dim V, \epsilon)$ converge to a common limit, as one would expect from a naïve application of a factorization $\text{success}(\dim V, \epsilon) = g(\epsilon(\frac{\dim V}{\dim \mathcal{X}})^{\frac{1}{\infty}}) = g(\epsilon)$. The reparameterization of Eq. (B.16) seems to do okay at accounting for behavior in the lowest dimensions, at the expense of over-compensating and pushing the curves corresponding to low-to-medium values of $\dim V$ to the left of the curve corresponding to $\dim V = \dim \mathcal{X}$. There are various potential causes of this undesirable effect (roughly one per crude oversimplification in the above analysis). Results for CIFAR10 and ImageNet can be found in Fig. 9 and Fig. 10 respectively. One concerning aspect of those two results is we see downturns in the curves $\text{success}(\dim V, \epsilon)$ for the highest values of ϵ, suggesting there may have been issues with our PGD optimizer in the $p = 1$ case.

One question we had was whether these results were impacted by sub-optimal PGD optimization. A reason for asking this is that the $p = 1$ case of Eq. (B.1) is arguably incorrect: the "correct" way of deriving these generalized Fast Gradient Sign Method (FGSM) steps is through the analysis of Appendix A. Assuming $U = I$ for simplicity, one

[9] This is of course quite false, as the numerator differs from the denominator by taking the max over a subsample. How can one deal with this step more realistically?



(a) (b)

Figure 4. **(a)** Success of PGD adversarial attacks on an MNIST trained small convolutional network, with ℓ^1-norm constraints on perturbation budget, constrained to subspaces $V \subseteq \mathcal{X}$ spanned by $\dim V$ randomly selected standard basis vectors. Adversarial examples are computed for all datapoints in the MNIST validation set. The x-axis is the ϵ-bound used during example generation and the different colored curves indicate the dimension $\dim V$ of the subspace to which the examples were constrained to, relative to the dimension $\dim \mathcal{X}$ ($= 28^2$) of the ambient space. **(b)** These curves become *more* aligned when we reparameterize the x-axis by scaling by $\frac{F(1-\frac{1}{\dim V})}{F(1-\frac{1}{\dim \mathcal{X}})}$, where F is the cummulative distribution function of the absolute value of a standard normal random variable.

sees that $w^T\delta = |w|_\infty |\delta|_1$, and one can show this occurs if and only if:

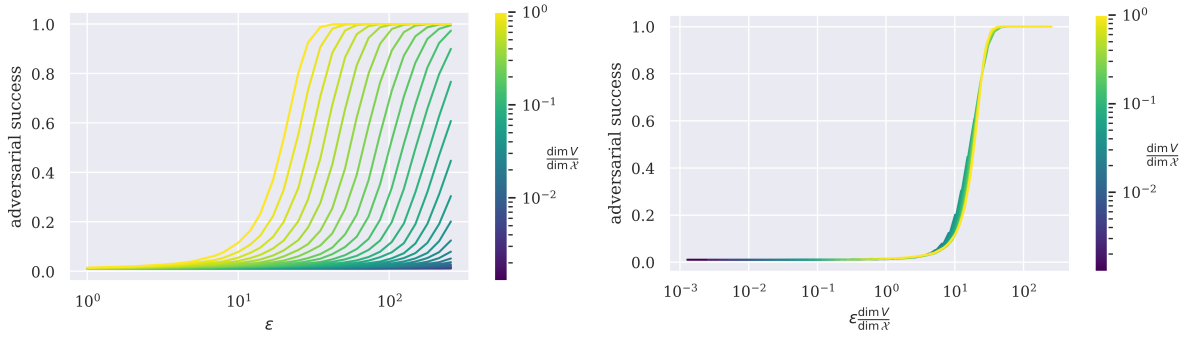
- letting $A = \arg\max_i\{|w_i|\} \subseteq \{1, \dots, n\}$ (the argmax of $|w_i|$, which is a set in general although a single index with probability 1), $\delta_i = 0$ if $i \notin A$.
- $\operatorname{sign} \delta_i = \operatorname{sign} w_i$ for all $i \in A$.

In the case where $A = \{a\}$ (i.e. $|w_i|$ has a unique maximum) we obtain the simplified solution $\delta = (c \operatorname{sign} w_i)e_a$ (where e_a is the a-th standard basis vector. Hence for $p = 1$, one can argue that we should use

$$\tilde{g}_t := e_{\arg\max_i\{|w_i|\}} \tag{B.17}$$

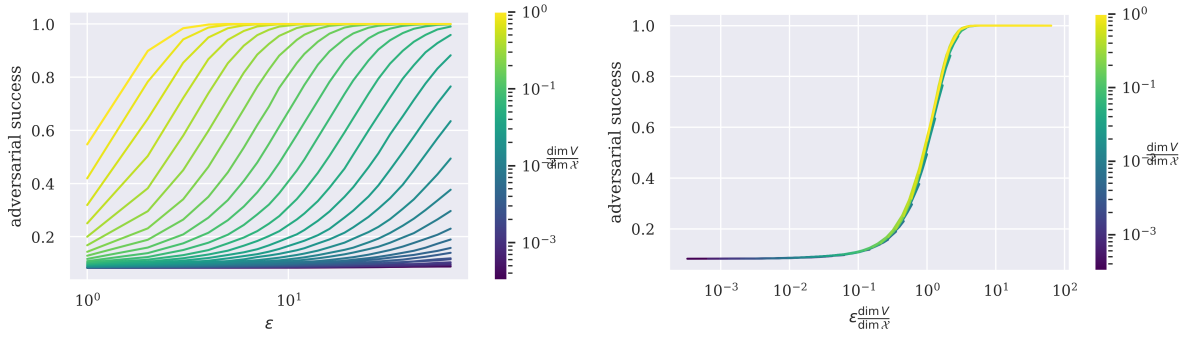
We found that while this method performed similarly to that of Eq. (B.1) for small values of ϵ, it struggled for large ϵ and failed at the scale of ImageNet input space (see). A reasonable suspicion is that the number of basis directions selected by Eq. (B.17) is bounded by the number of PGD iterations, and that when this number of iterations is far is smaller than the input dimension Eq. (B.17) underexplores. However we leave further analysis to future work.

C. Additional experimental results



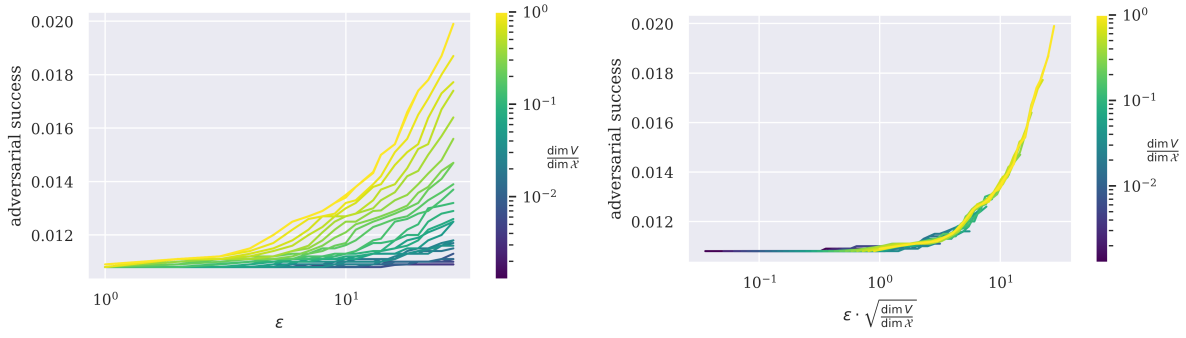
(a) (b)

Figure 5. Plot for experiments analogous to those found in Figure 1 but run with a 2-layer CNN trained and evaluated on MNIST.



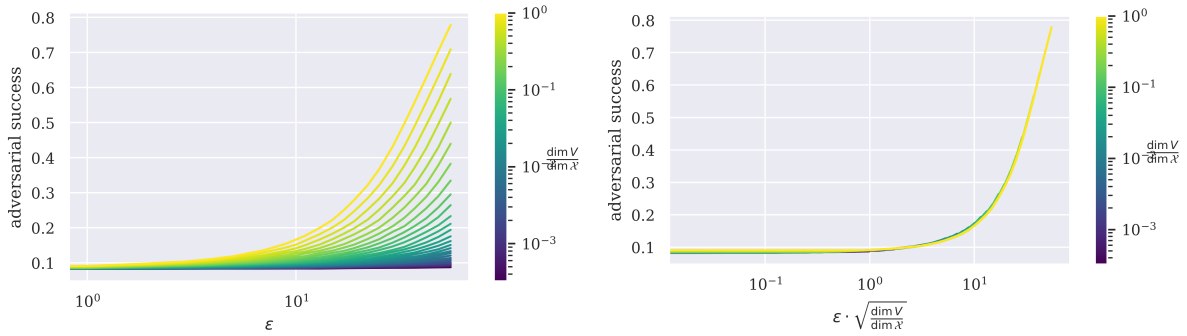
(a) (b)

Figure 6. Plot for experiments analogous to those found in Figure 1 but run with a ResNet9 trained and evaluated on CIFAR10.



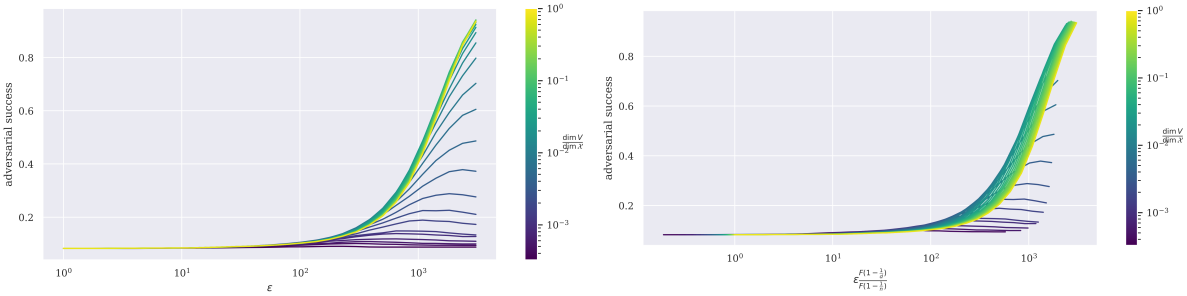
(a) (b)

Figure 7. Plot for experiments analogous to those found in Figure 2 but run with a 2-layer CNN trained and evaluated on MNIST.



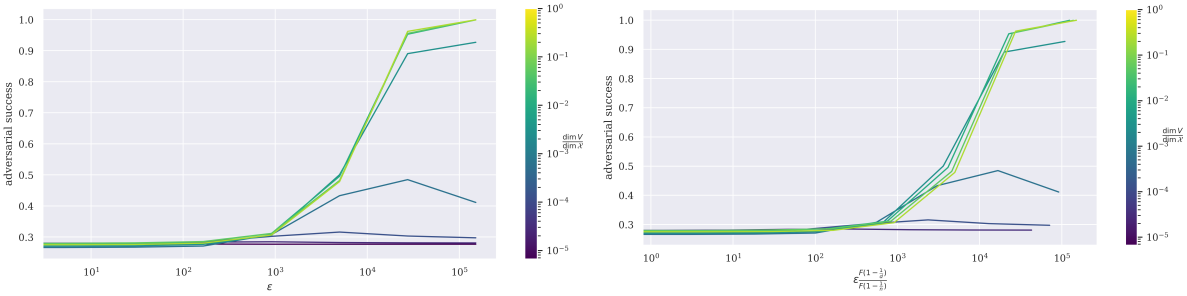
(a) (b)

Figure 8. Plot for experiments analogous to those found in Figure 2 but run with a ResNet9 trained and evaluated on CIFAR10.



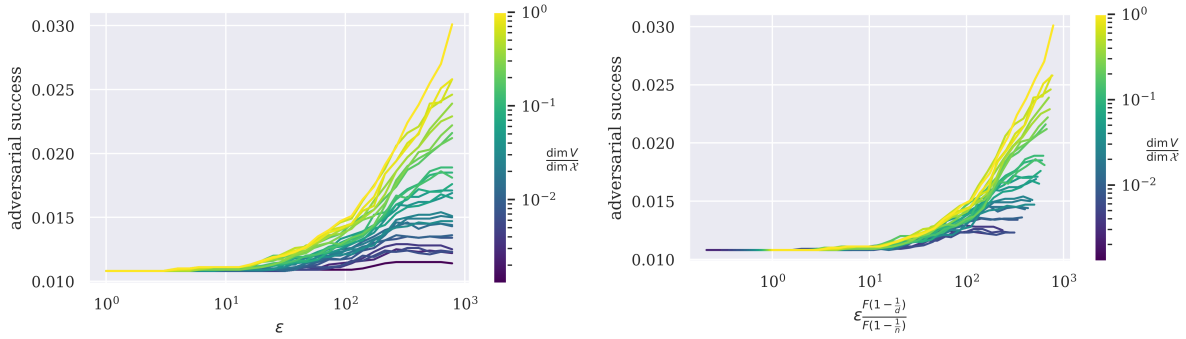
(a) (b)

Figure 9. Plot for experiments analogous to those found in Figure 4 but run with a ResNet9 trained and evaluated on CIFAR10.



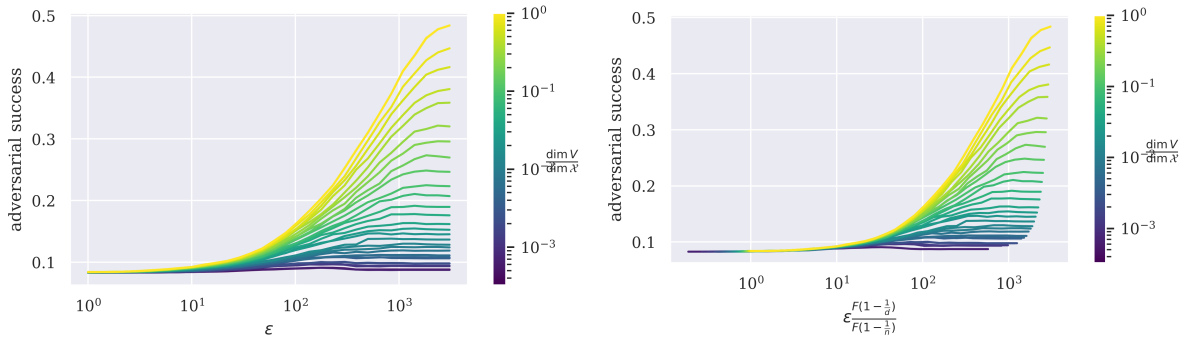
(a) (b)

Figure 10. Plot for experiments analogous to those found in Fig. 4 but run with a ResNet50 trained and evaluated on ImageNet.



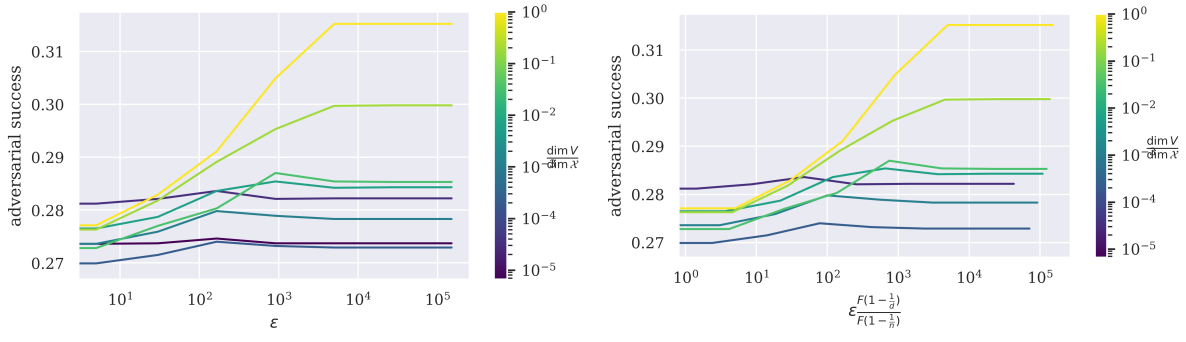
(a) (b)

Figure 11. Plot for experiments analogous to those found in Figure 4, the only difference being that we use the FGSM step of Eq. (B.17).



(a) (b)

Figure 12. Plot for experiments analogous to those found in Figure 4 but run with a ResNet9 trained and evaluated on CIFAR10, using the FGSM step of Eq. (B.17).



(a) (b)

Figure 13. Plot for experiments analogous to those found in Fig. 4 but run with a ResNet50 trained and evaluated on ImageNet, using the FGSM step of Eq. (B.17).