On Lipschitz Explosion in Deep Neural Networks with Normalization: Consequences for Optimization and Adversarial Robustness

Ashkan Soleymani
Massachusetts Institute of Technology
ashkanso@mit.edu

Reyhaneh Hosseinpourkhoshkbari
The University of Texas at Dallas
reyhaneh.hosseinpour@utdallas.edu

Hadi Daneshmand
University of Virginia
dhadi@virginia.edu

Patrick Jaillet
Massachusetts Institute of Technology
jaillet@mit.edu

Abstract

Can a network look stable at every layer and still be exponentially unstable as a whole? The Lipschitz constant of a neural network is a basic stability quantity appearing in optimization guarantees, generalization bounds, and robustness certificates. Deep networks naturally admit two such notions, *input Lipschitzness*, measuring sensitivity to data perturbations, and *parameter Lipschitzness*, measuring sensitivity to weight perturbations. We prove that, for deep networks with normalization layers, with batch normalization as the canonical example, *both* Lipschitz constants can grow exponentially with depth and hence with parameter dimension, even under strong per-layer norm control such as $\|W_\ell\|_2 \leq 1$, and already for linear activations. Thus, a mechanism widely used to stabilize training can create severe worst-case instabilities that are invisible locally, i.e., from layer-wise norm bounds alone. Parameter-Lipschitz explosion turns Lipschitz-dependent nonsmooth optimization guarantees into exponential-in-dimension bounds for deep normalized networks, while input-Lipschitz explosion makes worst-case Lipschitz-based generalization and robustness certificates vacuous. The latter also yields a concrete rank-separation mechanism for adversarial vulnerability. The theory predicts that perturbations introducing new singular directions should be amplified much more strongly than perturbations at the same noise level that remain within the input's original singular subspace. Experiments on MNIST, Fashion-MNIST, and CIFAR-10 support this prediction, showing that rank-creating perturbations cause substantially sharper drops in accuracy, confidence, and margins than same-subspace perturbations.

1 Introduction

Increasing the depth of neural networks is a key driver of modern performance, boosting image classification [He et al., 2016] and text generation in large language models [Radford et al., 2019]. Yet increased depth also makes optimization more challenging, motivating architectural stabilizers such as residual connections [He et al., 2016] and normalization layers [Ioffe and Szegedy, 2015], which are now standard components of modern architectures. In this work, we take a closer look at batch normalization in particular, a seemingly routine component of modern deep neural networks, and study its *unexpected implications* for optimization, generalization, and adversarial robustness.

A recurring quantity linking depth to both optimization and stability is a model's (local) *Lipschitz constant*. Crucially, deep networks come with two different worst-case Lipschitz notions, depending on what is perturbed. Writing F_θ for the network map with parameters θ (and using Frobenius /

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Euclidean norms for concreteness), we consider

$$\mathrm{Lip}_X(F_\theta) := \sup_{X \neq Y} \frac{\|F_\theta(X) - F_\theta(Y)\|_F}{\|X - Y\|_F}, \qquad \mathrm{Lip}_\Theta(F; X_{\mathrm{in}}) := \sup_{\theta \neq \theta'} \frac{\|F_\theta(X_{\mathrm{in}}) - F_{\theta'}(X_{\mathrm{in}})\|_F}{\|\theta - \theta'\|_2}.$$

The *parameter* Lipschitz constant Lip_Θ governs how sharply the network output (and hence the induced training objective) can change as the weights move, and it therefore enters worst-case iteration-complexity guarantees in nonsmooth optimization. The *input* Lipschitz constant Lip_X governs worst-case sensitivity to data perturbations, and it appears (often via upper bounds) in several generalization and robustness analyses based on margins and stability, including spectrally-normalized bounds and related Lipschitz controls [Bartlett et al., 2017, Bubeck and Sellke, 2021]. We use these *doubly Lipschitz* notions as a lens to study unexpected implications of normalization layers in optimization, generalization, and adversarial robustness.

Main result: both Lipschitz constants can be exponentially large. We show that for deep networks with normalization layers (with batch normalization as the canonical example), *both* $\mathrm{Lip}_X(F_\theta)$ and $\mathrm{Lip}_\Theta(F; X_{\mathrm{in}})$ can grow *exponentially* with depth m, and consequently exponentially with the ambient parameter dimension d. This occurs even under strong per-layer norm control (e.g., $\|W_\ell\|_2 \leq 1$, and in the input-Lipschitz construction even W_ℓ orthogonal), and already for linear activations. Put differently, we find that while normalization stabilizes many practical training pipelines [Ioffe and Szegedy, 2015, Salimans and Kingma, 2016, Ba et al., 2016, Ioffe, 2017, Daneshmand et al., 2021, Lange et al., 2022], *it causes worst-case doubly Lipschitz explosion as depth increases*.

1.1 Implications of parameter-Lipschitz explosion for nonsmooth optimization

For smooth objectives, first-order methods admit dimension-independent rates for reaching approximate stationarity [Nesterov, 2013]. Neural-network training objectives, however, are typically *nonsmooth* (e.g., due to ReLU activations, pooling layers or normalizations), so stationarity is captured by generalized-gradient notions. A classical tractable notion is *Goldstein stationarity* [Goldstein, 1977], and recent algorithms can find approximate Goldstein stationary points with oracle complexity polynomial in the Lipschitz constant L of the objective [Zhang et al., 2020, Davis et al., 2022, Jordan et al., 2023]. In our setting, L is not an external constant; instead, it is induced by the network and our lower bound shows it can be $\exp(\Omega(d))$ for deep normalized architectures. Consequently, Lipschitz-based "dimension-free" guarantees of the form $\mathrm{poly}(L, 1/\varepsilon, 1/\delta)$ can unfortunately translate into *exponential dependence on parameter* d for this model class. Moreover, lower bounds in nonsmooth optimization show that some polynomial dependence on L is unavoidable in the worst case [Kornowski and Shamir, 2022], underscoring that one cannot expect a uniform theory for Goldstein stationarity that is both fully general and only polylogarithmic in L. Motivated by the optimization barrier created by exploding Lipschitz constants, we consider a relaxed directional notion (probabilistic Goldstein stationarity) and give a basic algorithm that finds such points with oracle complexity that does *not* depend polynomially on L. This begets a new quest to find more refined solution concepts (with probabilistic Goldstein stationarity as a basic starting point) that are theoretically tractable in $\mathrm{poly}(d, 1/\varepsilon)$ time and hence in $\mathrm{poly}(d, 1/\varepsilon, \log L)$ time.

1.2 Implications of input-Lipschitz explosion for generalization and adversarial robustness

The same phenomenon impacts learning-theoretic uses of Lipschitz control. Since $\mathrm{Lip}_X(F_\theta)$ (or upper bounds on it) enters several margin-based generalization bounds and Lipschitz-based stability arguments [Bartlett et al., 2017, Bubeck and Sellke, 2021, Hou et al., 2023], an exponential lower bound indicates that these worst-case guarantees can deteriorate rapidly with depth, even when each layer is individually norm-controlled. Therefore, we conclude that the Lipschitzness assumption may not be realistic for studying generalization in deep neural networks with normalization layers, since this assumption becomes vacuous exponentially fast with depth.

The input Lipschitzness $\mathrm{Lip}_X(F_\theta)$ lens on batch normalization also sheds new light on recent empirical evidence that batch normalization can have an outsized effect on adversarial robustness [Galloway et al., 2019, Wang et al., 2022, Zeise and Vlaar, 2025]. By showing theoretically that the Lipschitzness of deep neural networks grows exponentially fast with depth as a byproduct of batch normalization, we show that normalization layers significantly increase the sensitivity of neural-network outputs to noise in their inputs. Therefore, we can partially explain this empirical observation.

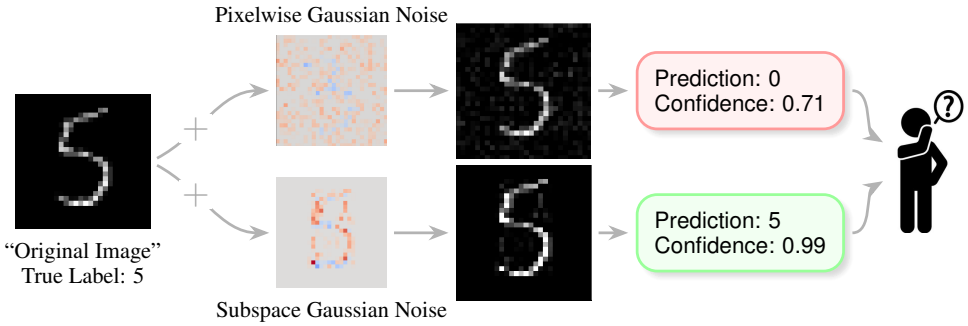


Figure 1: We study the effect of different noise structures on the robustness of a multilayer neural network with normalization layers. Above, we add standard i.i.d. Gaussian noise; below, we instead apply noise restricted to the subspace of the original image. In the noisy images, red indicates positive values and blue indicates negative values. As depicted, even though the structured noise in the same subspace as the original image changes the image much more meaningfully (here, from a human perspective, making the 5 look more like an 8 as also observed in the noise pattern) the model still correctly classifies the image with high confidence. In contrast, under pixel-wise noise, despite the change being barely noticeable from a human perspective, the model is completely fooled and misclassifies the image with relatively high confidence. Our theoretical analysis predicts this phenomenon: the original image is low-rank, while i.i.d. Gaussian pixel noise makes the noisy image full-rank but nearly rank-deficient. Due to this relationship between the original image and the noisy one, normalization layers can amplify their separation across layers, causing neural networks' vulnerability to noise. This separation and vulnerability do not arise when Gaussian noise is added within the same subspace as the original image, even at a much higher magnitude.

At a high level, our hardness construction for exponential growth of $\mathrm{Lip}_X(F_\theta)$ with depth exploits a rank-separation effect. The witness pair consists of a low-rank matrix and a nearby matrix that is full-rank but nearly rank-deficient. Although these two inputs are close in norm, normalization can amplify their separation across layers, leading to an exponential growth in the input Lipschitz constant. This mechanism suggests a corresponding empirical prediction for adversarial vulnerability under batch normalization: If a clean image, or a patch-based representation of it, has approximate low-rank structure (as suggested by classical low-rank patch models for natural images and by low-dimensional structure observed in datasets such as MNIST, Fashion-MNIST, CIFAR-10, and ImageNet [Savas and Eldén, 2007, Zhang et al., 2013, Gu et al., 2014, Xu et al., 2015, Liu et al., 2016]) then adding small i.i.d. noise can generically make the representation full-rank while keeping it close to the low-rank set. Our theory predicts that batch-normalized networks amplify this small perturbation through depth, resulting in a large change in the output. We experimentally validate this prediction (see Figure 1).

Our main contributions are as follows:

- We distinguish two natural Lipschitz notions for deep networks: parameter Lipschitzness and input Lipschitzness. Through this lens, we reveal unexpected implications of normalization layers for optimization, generalization, and adversarial robustness.
- For deep networks with normalization layers, we show that both input and parameter Lipschitz constants can grow exponentially with depth (and hence with parameter dimension) even under strong per-layer norm constraints and even with linear activations.
- We clarify how parameter-Lipschitz explosion propagates to worst-case nonsmooth optimization. Existing guarantees for Goldstein stationarity that are polynomial in the objective Lipschitz constant L become exponential in the parameter dimension for deep normalized networks. This motivates a new search for tractable stationarity notions with complexity polynomial in dimension and accuracy, but only polylogarithmic in L. We initiate this search by introducing probabilistic Goldstein stationarity, a basic directional relaxation of Goldstein stationarity derived from its dual formulation. We give a simple function-value algorithm that finds such points with oracle complexity that avoids polynomial dependence on L.
- We connect input-Lipschitz explosion to generalization and adversarial robustness. In particular, our results show that worst-case Lipschitz-based generalization, margin, and stability arguments

can become vacuous exponentially fast with depth in normalized networks, even when each layer is norm-controlled.

- More broadly, our construction identifies a simple mechanism for adversarial sensitivity in batch-normalized networks. Normalization can amplify small input differences when two nearby inputs have different rank structure. Consequently, when the data are approximately low rank, perturbations that introduce new singular directions can be magnified across layers, while perturbations that remain within the original singular subspace may be much less disruptive. Our experiments support this prediction across MNIST, Fashion-MNIST, and CIFAR-10. Perturbations that change the input's singular structure lead to substantially sharper drops in accuracy, confidence, and classification margins than perturbations at the same noise level that stay within the input's original singular subspace. This suggests a geometry-dependent failure mode of normalized networks and helps explain why batch normalization can have an outsized effect on adversarial robustness.

2 Exponential Explosion of Two Lipschitz Constants with Dimension

In this section, we study the Lipschitz constant of deep neural networks. In particular, we show that the presence of normalization layers, with a focus on batch normalization, causes the Lipschitz constant of the network to grow exponentially with depth, and consequently with the dimension of the parameter space d. We observe this explosion in two notions of Lipschitz continuity: (i) the Lipschitz constant of the deep neural network with fixed weights, viewed as an input–output map with respect to the inputs, and (ii) the Lipschitz constant of the deep neural network with fixed input data, viewed as a function on the parameter space with respect to the weights.

2.1 Preliminaries and Notation

Notation. For a matrix X, $\|X\|_F$ is the Frobenius norm and $\|X\|_2$ the operator norm. For a square matrix A, $\lambda_{\min}(A)$ denotes its smallest eigenvalue. We write $X_{i:}$ for the i-th row of X and $X_{:j}$ for the j-th column of X. We denote the set $\{1, 2, \dots, m\}$ by $[m]$.

Batch normalization (Column-wise normalization on $\mathbb{R}^{n\times k}$). Let $X \in \mathbb{R}^{n\times k}$. Define the diagonal matrix of column-squared norms

$$D(X) \coloneqq \operatorname{diag}(X^\top X) \in \mathbb{R}^{k\times k}, \qquad D(X)_{jj} = \|X_{:j}\|_2^2.$$

When every column of X is nonzero (equivalently $D(X) \succ 0$), the *column-wise batch normalization* map is $\mathrm{BN}(X) \coloneqq X\, D(X)^{-1/2}$. Equivalently, $(\mathrm{BN}(X))_{ij} = X_{ij}/\|X_{:j}\|_2$. We emphasize that the batch normalization operator defined above omits the mean-subtraction step for convenience, in line with prior formulations and theoretical studies [Daneshmand et al., 2021, Joudaki et al., 2023, Meterez et al., 2024]. Our results extend to mean-subtracted batch normalization by an isometric embedding into the mean-zero subspace; Appendix B.8 gives the construction. If $D(X) \succ 0$, then every column of $\mathrm{BN}(X)$ has unit norm, hence

$$\|\,\mathrm{BN}(X)\|_F^2 = \operatorname{tr}(\mathrm{BN}(X)^\top\,\mathrm{BN}(X)) = k. \tag{1}$$

Remark 2.1 (Regularization at zero columns)**.** *If one wants* BN *to be defined on* all *of $\mathbb{R}^{n\times k}$, one may work with the standard ε-regularized variant*

$$\mathrm{BN}_\varepsilon(X) \coloneqq X\,(D(X) + \varepsilon \mathbf{I}_k)^{-1/2}, \qquad \varepsilon > 0,$$

and (when desired) take $\varepsilon \downarrow 0$ at the end. Our lower bound proof below is written for $\mathrm{BN}(X) = X\,D(X)^{-1/2}$ on the open set $\{D(X) \succ 0\}$, and remains valid for BN_ε with the same qualitative conclusion.

A deep network with normalization layers and linear activation. Fix an integer $n \geq 1$, i.e., the batch size. Let $k_0, k_1, \dots, k_m$ be positive integers (layer widths / feature dimensions). For each layer $\ell \in \{1, \dots, m\}$ let $W_\ell \in \mathbb{R}^{k_{\ell-1}\times k_\ell}$ be a weight matrix. Given an input matrix $X^{(0)} \in \mathbb{R}^{n\times k_0}$, define the forward iterates

$$X^{(\ell)} \coloneqq f_{W_\ell}\big(X^{(\ell-1)}\big) \overset{\text{def}}{=} \mathrm{BN}\big(X^{(\ell-1)}W_\ell\big) \in \mathbb{R}^{n\times k_\ell}, \qquad \ell = 1, \dots, m, \tag{2}$$

whenever the normalization is well-defined. The depth-m network map is

$$F_m \coloneqq f_{W_m} \circ \cdots \circ f_{W_1} : \mathbb{R}^{n\times k_0} \to \mathbb{R}^{n\times k_m}.$$

Why linear activations? We focus on linear activations because they provide a conservative setting for a lower bound. By removing nonlinearities, we rule out amplification effects that could come from the activation function itself, so any growth we prove must come from the interaction between the weights and normalization. This choice is also consistent with the mean-field analysis of batch-normalized networks in Yang et al. [2019], where linear activations exhibit the slowest gradient-norm growth among the activation functions considered, even though the growth can still be exponential in depth. Thus, showing exponential Lipschitz growth already in the linear case demonstrates that the phenomenon is not caused by nonlinear activations. We expect the same mechanism to persist for a broader class of standard nonlinearities, we show this result for networks with ReLU activations in Appendix B.9 and leave the rest to future work.

Input and parameter Lipschitzness. We use two Lipschitz notions, depending on whether the perturbation is applied to the input or to the weights. First, fix a sequence of weights $(W_1, \dots, W_m)$ and and an input domain $\mathcal{X}$ on which every normalization step is defined. View the network as a map from inputs to outputs. Its input Lipschitz constant is

$$\mathrm{Lip}_{\mathcal{X}}(F_m) := \sup_{\substack{X,Y\in\mathcal{X}\\ X\neq Y}} \frac{\|F_m(X) - F_m(Y)\|_F}{\|X - Y\|_F}.$$

Second, fix an input batch $X_{\mathrm{in}} \in \mathbb{R}^{n\times k_0}$ and view the network output as a function of the parameters. For a weight tuple $\theta = (W_1, \dots, W_m) \in \Theta \subset \bigtimes_{\ell\in[m]} \mathbb{R}^{k_{\ell-1}\times k_\ell}$, we use the Euclidean norm obtained by vectorizing all weights, $\|\theta\|_2 := \left(\sum_{\ell=1}^m \|W_\ell\|_F^2\right)^{1/2}$. The corresponding parameter-to-output map and its Lipschitz constant are

$$\mathcal{F}_m(\theta) := F_m^\theta(X_{\mathrm{in}}), \qquad \mathrm{Lip}_\Theta(\mathcal{F}_m) := \sup_{\substack{\theta,\theta'\in\Theta\\ \theta\neq\theta'}} \frac{\|\mathcal{F}_m(\theta) - \mathcal{F}_m(\theta')\|_F}{\|\theta - \theta'\|_2}.$$

The first quantity measures sensitivity to input perturbations, while the second measures sensitivity to changes in the weights.

Remark 2.2. *Normalization layer is invariant under positive rescaling since* $\mathrm{BN}(\alpha X) = \mathrm{BN}(X)$ *for every* $\alpha > 0$*. Our exponential lower bounds are established on compact admissible domains with scales bounded away from zero uniformly in depth. Specifically, the input domains satisfy* $1 \le \|X\|_F \le \sqrt{k}$*, while every weight matrix in the parameter witnesses has operator norm one.*

2.2 Explosion of the Input Lipschitz Constant $\mathrm{Lip}_{\mathcal{X}}(F_m)$

We first show that the input Lipschitz constant $\mathrm{Lip}_{\mathcal{X}}(F_m)$ can grow exponentially with depth. This growth is not an artifact of a large or expansive linear map. In fact, it already occurs in the following highly controlled setting with orthogonal weights:

$$n = k_0 = k_1 = \cdots = k_m = k \ge 2, \qquad X^{(0)} \in \mathbb{R}^{k\times k}, \qquad W_\ell \in \mathbb{O}(k). \tag{3}$$

Here each weight matrix is orthogonal $W_\ell \in \mathbb{O}(k)$, so $\|W_\ell\|_2 = 1$ and therefore $\|XW_\ell\|_F \le \|X\|_F$. Thus, the linear part of every layer is non-expansive.

The setting is deliberately restrictive. The linear maps do not increase Frobenius norm, and by (1), every post-normalization activation has fixed norm, $\|X^{(\ell)}\|_F = \sqrt{k}$. In particular, the activations remain in a bounded region throughout the network. Nevertheless, we show below that the input–output map can still have an input Lipschitz constant that grows exponentially with the depth m. Moreover, this growth is witnessed by uncountably many input pairs.

Theorem 2.3 (Exponential lower bound on $\mathrm{Lip}_{\mathcal{X}}(F_m)$)**.** *Fix* $k \ge 2$ *and constant* $\eta \in (0, 1)$*. There exist constants* $C_{k,\eta} > 0$ *and* $\gamma_{k,\eta} > 0$ *such that, for every depth* $m \in \mathbb{N}$*, one can choose orthogonal weights* $W_1, \dots, W_m \in \mathbb{O}(k)$ *with* $\|W_\ell\|_2 = 1$ *for all* $\ell \in [m]$*, for which the depth-*m *map* F_m *defined in* (2)–(3) *satisfies* $\mathrm{Lip}_{\mathcal{X}}(F_m) \ge C_{k,\eta}\, e^{\gamma_{k,\eta} m}$*, where constants* $C_{k,\eta}$ *and* $\gamma_{k,\eta}$ *depend on* k, η*. Moreover, for this same fixed choice of weights, the lower bound is witnessed by an uncountable family of input pairs.*

Our proof construction uses *orthogonal* weights, so the linear maps themselves do not create any expansion. The instability comes from the interaction between normalization and near rank-deficiency.

We choose two inputs, X_0 and Y_0, that are exponentially close, with Y_0 rank-deficient and X_0 only slightly perturbed away from being rank-deficient. After m carefully chosen normalization steps, the trajectory starting from X_0 becomes uniformly well-conditioned, while the trajectory starting from Y_0 remains rank-deficient. Thus the two outputs are separated by at least $\sqrt{\lambda_{\min}(X_m^\top X_m)}$ which is relatively high in a well-conditioned matrix, even though the original perturbation was exponentially small. This creates the exponential separation-to-perturbation ratio. We give the full proof of this result in Appendix B.6, and provide a proof sketch in Appendix B.1.

The next corollary translates the depth-dependent lower bound into a lower bound in terms of the number of parameters. Thus, even when the parameter count is the measure of complexity, the input Lipschitz constant can still grow exponentially.

Corollary 2.4. *Fix $\eta \in (0,1)$ and set $k = 2$. There exist constants $C_\eta > 0$ and $\Gamma_\eta > 0$ such that, for every integer $d \geq 4$, there is a depth-m network with at most d scalar weight parameters whose input–output map satisfies $\mathrm{Lip}_{\mathcal{X}}(F_m) \;\geq\; C_\eta\, e^{\Gamma_\eta d}$.*

Remark 2.5. *Although the constants in our general bounds in Theorem 2.3 depend on k, the exponential lower bounds hold uniformly over all widths $k \geq 2$. Indeed, the width-two construction embeds into any larger width via block-diagonal matrices, preserving the same exponential rate and constants independently of k; see Appendix B.10.*

2.3 Explosion of the Parameter Lipschitz Constant $\mathrm{Lip}_\Theta(\mathcal{F}_m)$

In Section 2.2, we showed that the input–output map can have an input Lipschitz constant $\mathrm{Lip}_{\mathcal{X}}(F_m)$ that grows exponentially with depth. We now show that the same phenomenon occurs when the input is fixed and the network is viewed as a function of its parameters. In other words, normalization can also lead to exponential growth of the parameter Lipschitz constant $\mathrm{Lip}_\Theta(\mathcal{F}_m)$.

Theorem 2.6 (Exponential lower bound on $\mathrm{Lip}_\Theta(\mathcal{F}_m)$)**.** *Fix $k \geq 2$ and constant $\eta \in (0,1)$. There exist constants $\widetilde{C}_{k,\eta} > 0$ and $\widetilde{\gamma}_{k,\eta} > 0$ depending on k and η such that, for every depth $m \geq 2$, there is a depth-m network of the form (2), with $\|W_\ell\|_2 \leq 1$ for all $\ell \in [m]$, for which the fixed-input parameter-to-output map $\mathcal{F}_m(\theta) := F_m^\theta(X_{\mathrm{in}})$ with $X_{\mathrm{in}} = \mathbf{I}_k$ satisfies $\mathrm{Lip}_\Theta(\mathcal{F}_m) \;\geq\; \widetilde{C}_{k,\eta}\, e^{\widetilde{\gamma}_{k,\eta} m}$.*

The proof is a direct reduction from the input-Lipschitz construction in Theorem 2.3. We fix the input to be $X_{\mathrm{in}} = \mathbf{I}_k$ and encode the two input witnesses from Theorem 2.3 into the first-layer weights. The two parameter choices differ only in this first layer, while all subsequent weights are kept the same as in the input construction. By the scale-invariance of batch normalization, the resulting forward pass reproduces the same two trajectories after the first layer. Thus the parameter perturbation is as small as the original input perturbation, while the output separation remains bounded below, giving the same exponential separation ratio. Full details are provided in Appendix B.7. As in Corollary 2.4, this depth-dependent lower bound can also be translated into an exponential lower bound in the parameter dimension d.

Corollary 2.7. *Fix any $\eta \in (0,1)$ and set $k = 2$. There exist constants $\widetilde{C}_\eta > 0$ and $\widetilde{\Gamma}_\eta > 0$ such that, for every integer $d \geq 8$, one can construct a depth-m network with at most d scalar weight parameters for which $\mathrm{Lip}_\Theta(\mathcal{F}_m) \;\geq\; \widetilde{C}_\eta\, e^{\widetilde{\Gamma}_\eta d}$.*

From the optimization viewpoint, the relevant quantity is the Lipschitz constant of the training objective that is fed by the network. To make this connection explicit, fix a realizable target parameter $\theta^\star \in \Theta$ and define

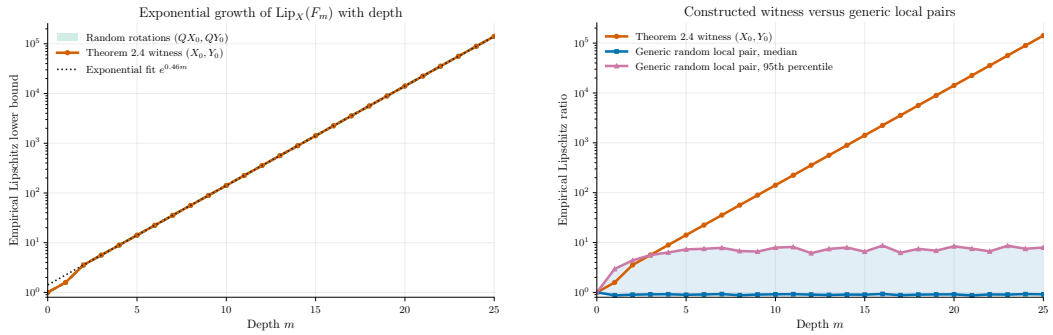
$$f(\theta) := \|\mathcal{F}_m(\theta) - \mathcal{F}_m(\theta^\star)\|_F, \qquad \mathrm{Lip}(f) := \sup_{\substack{\theta,\theta' \in \Theta \\ \theta \neq \theta'}} \frac{|f(\theta') - f(\theta)|}{\|\theta' - \theta\|_2}, \tag{4}$$

Here $\mathcal{F}_m(\theta^\star)$ plays the role of the observed label generated by the ground-truth network. The objective function $f(\theta)$ is equivalent to the mean-squared loss evaluated on the outputs of the neural network.

The next corollary shows that the exponential growth of $\mathrm{Lip}_\Theta(\mathcal{F}_m)$ transfers to the Lipschitz constant of this induced objective.

Corollary 2.8. *Let $\mathcal{F}_m$ be the parameter-to-output map constructed in Theorem 2.6. There exists a realizable target parameter $\theta^\star \in \Theta$ such that the objective f defined in (4) satisfies*

$$\mathrm{Lip}(f) \;\geq\; \widetilde{C}_\eta e^{\widetilde{\Gamma}_\eta d},$$



(a) Constructed witnesses exhibit exponential growth.

(b) Generic local pairs do not exhibit the same growth.

Figure 2: Empirical validation of Lipschitz explosion. The constructed witness pair (X_0, Y_0) and its random rotations (QX_0, QY_0) grow exponentially with depth, while the median and 95th percentile over generic random local pairs remain nearly flat.

up to the constants in Corollary 2.7.

Thus, the same exponential sensitivity appears in the induced optimization objective and directly affects convergence guarantees that scale with its Lipschitz constant.

2.4 Experiments for Lipschitz Explosion

We now empirically validate the exponential input-Lipschitz growth predicted by our theory. The results are shown in Figure 2 on a logarithmic scale. We focus on the setting of Corollary 2.4 with $k = 2$. Since the proof of Theorem 2.3 establishes existence through a nonconstructive probabilistic argument over the weights $\{W_\ell\}_{\ell=1}^m$, we use a constructive heuristic to select the weights in our experiments. Specifically, at each layer $\ell \in [m]$, we choose W_ℓ greedily so as to maximize the isotropy functional from Definition B.1. In the two-dimensional setting, this amounts to a deterministic grid search over rotation angles. At each layer, we scan a grid of candidate rotations and select the one that maximizes the isotropy of the next activation.

Figure 2a compares the empirical Lipschitz ratio obtained from the constructed witness pair (X_0, Y_0) with the predicted exponential growth. The same plot also shows witnesses obtained by applying random rotations Q to the input witness pair, yielding pairs of the form (QX_0, QY_0). As predicted by the orthogonal equivariance of the construction, these rotated witnesses exhibit the same exponential growth behavior. Figure 2b compares the constructed witness with generic local input pairs. The random local pairs are generated from i.i.d. standard Gaussian perturbations, and we report both the median and the 95th percentile of their empirical Lipschitz ratios. Unlike the constructed witness, these generic local pairs do not exhibit exponential growth with depth. These experiments support two complementary conclusions. First, the exponential growth predicted by our theory is clearly observable for the constructed witnesses. Second, this behavior is not typical for arbitrary local perturbations. Thus, the witness construction captures a genuine worst-case instability of normalized networks rather than a numerical artifact.

3 Implications for Optimization Theory in Deep Neural Networks

We now turn to the optimization consequences of the lower bound $L := \mathrm{Lip}(f) \geq \exp(\Omega(d))$ from Section 2.3. Dimension-free convergence rates are often viewed as especially desirable in deep learning, since modern networks can have a very large number of parameters. Our results show that, for normalized networks, this interpretation can be misleading. A guarantee may be formally independent of the parameter dimension d, but if it depends polynomially on the objective Lipschitz constant L, then Corollary 2.8 implies that the same guarantee can still scale exponentially with d. In this sense, Lipschitz-dependent "dimension-free" rates need not be truly dimension-free for deep normalized architectures.

This issue is particularly relevant in nonsmooth optimization. Neural-network training objectives are nonconvex, and they are often nonsmooth as well. Even if the outer loss is smooth, standard

architectural components such as ReLU [Nair and Hinton, 2010] and leaky-ReLU activations [Maas et al., 2013], max-pooling [Lee et al., 2009], absolute-value regularizers [Scardapane et al., 2017], and piecewise-linear combinatorial layers [Martins and Astudillo, 2016, Blondel et al., 2020] can introduce nonsmoothness into the objective. At the same time, these objectives are typically locally Lipschitz and differentiable almost everywhere, so generalized-gradient notions of stationarity provide the natural language for worst-case analysis. The relevant background on nonsmooth stationarity is recalled in Appendix C.1; here, we focus on how the Lipschitz explosion above changes the interpretation of existing convergence guarantees.

3.1 Complexity of Finding Goldstein Stationary Points

Existing algorithms for finding (δ, ϵ)-approximate Goldstein stationary points have oracle complexity that depends polynomially on the objective Lipschitz constant L [Zhang et al., 2020, Davis et al., 2022, Lin et al., 2022, Jordan et al., 2023, Kong and Lewis, 2023]. Such a dependence is acceptable when L is viewed as a fixed property of the optimization problem. In deep normalized networks, however, L is not an external constant; it is determined by the architecture and can grow with depth. Our results in Section 2 show that L can be as large as $\exp(\Omega(d))$. Consequently, a guarantee of the form $\mathcal{O}(\mathrm{poly}(L, 1/\epsilon, 1/\delta))$ can translate into exponential dependence on the parameter dimension d, even when the stated rate has no explicit dependence on d.

This raises a natural question: *can one design algorithms for Goldstein stationarity whose dependence on L is only logarithmic, for example with oracle complexity polynomial in d, $1/\epsilon$, $1/\delta$, and $\log L$?*

We show that, in full generality, the answer is negative. One cannot hope for a randomized first-order method whose oracle complexity is polynomial in d, $1/\epsilon$, $1/\delta$, and only logarithmic in L, uniformly over all Lipschitz objectives. The obstruction is not specific to neural networks or to nonconvexity: it already appears for convex Lipschitz functions. We rely directly on Kornowski and Shamir [2022, Theorem 3], which gives a lower bound on the oracle complexity of finding (δ, ϵ)-Goldstein stationary points of 1-Lipschitz functions.

Corollary 3.1 (Informal)**.** *In the worst case, polynomial dependence on the Lipschitz constant L is unavoidable for finding (δ, ϵ)-Goldstein stationary points. In particular, no randomized first-order algorithm can have oracle complexity only $\mathrm{poly}(d, 1/\epsilon, 1/\delta, \log L)$ uniformly over all Lipschitz objectives.*

3.2 Toward Tractable Solution Concepts

These observations highlight a mismatch between existing nonsmooth optimization guarantees and the regimes that arise in deep normalized networks. As we observed, current guarantees for Goldstein stationarity do not yield truly polynomial-time bounds once the architecture-induced Lipschitz constant L is taken into account. This suggests a natural quest: to identify weaker, but still meaningful, stationarity notions whose complexity is polynomial in d, $1/\varepsilon$, and $1/\delta$, and polynomial only in $\log L$. As a first step in this direction, we introduce probabilistic Goldstein stationarity, a simple directional relaxation of Goldstein stationarity. We use it as *a basic baseline rather than a final answer*, and give a simple function-value algorithm that finds such points with number of queries independent of L.

Goldstein stationarity is a worst-case directional condition. To see this, fix $\delta > 0$ and let $\partial_\delta f(x)$ denote the Goldstein subdifferential. For a direction $v \in \mathbb{S}^{d-1}$, consider the support function of the set of Goldstein subdifferentials $m_\delta(x; v) := \min_{g \in \partial_\delta f(x)} \langle g, v \rangle$. By Lemma C.3, Goldstein stationarity can be written in the equivalent form $\mathrm{dist}(0, \partial_\delta f(x)) \le \varepsilon \iff \max_{v \in \mathbb{S}^{d-1}} m_\delta(x; v) \le \varepsilon$. Thus, a (δ, ε)-Goldstein stationary point is one for which no direction has a large Goldstein margin. This dual viewpoint goes back to the work of Goldstein [1977]. We now relax this worst-case requirement by asking only that large-margin directions occupy a small fraction of the sphere.

Definition 3.2 (Probabilistic Goldstein stationarity (PGS))**.** *Fix $\delta > 0$, a margin threshold $\tau > 0$, and a probability level $\rho \in (0, 1)$. Let $V \sim \mathrm{Unif}(\mathbb{S}^{d-1})$. We say that x is (δ, τ, ρ)-*PGS *if $\Pr\left[m_\delta(x; V) \ge \tau\right] \le \rho$. We write x is (δ, ε)-*PGS *as shorthand for $(\delta, \tau, \rho) = (\delta, \varepsilon, \varepsilon)$.*

We view probabilistic Goldstein stationarity (PGS) as a first step toward understanding which stationarity notions are tractable for deep normalized networks, rather than as a definitive solution concept. Additionally, we give a simple algorithm for finding PGS points that uses only function valuations and prove its convergence guarantee.

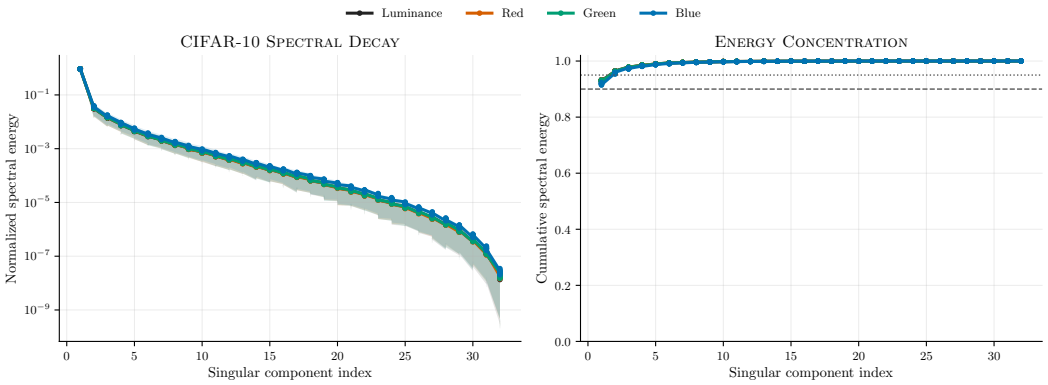


Figure 3: Spectral structure of CIFAR-10 luminance and RGB channel matrices. Left: normalized Gram eigenvalues $\lambda_j(X^\top X)/\operatorname{tr}(X^\top X)$. Right: cumulative spectral energy, with horizontal lines at 90% and 95%. Most of the energy lies in the first few of the 32 singular components.

Theorem 3.3. *There exists a randomized function-value algorithm that finds a (δ, τ, ρ)-PGS point with probability at least $1-\gamma$ using $\mathcal{O}\left(\frac{\Delta}{\tau\delta\rho}\log\frac{\Delta}{\gamma\tau\delta}\right)$ queries, with no polynomial dependence on L.*

4 Implications for Generalization and Robustness in Deep Neural Networks

The implications of Lipschitz explosion results for generalization are relatively direct. As discussed in Section 1, blindly assuming Lipschitzness L for deep normalized neural networks can make the resulting generalization bounds vacuous with depth (and number of parameters). We now turn to the unexpected implications of our results for the robustness of neural networks.

We next connect Lipschitz explosion to decision flips, showing that it can directly undermine the robustness of deep normalized networks to input noise. The input-Lipschitz lower bound in Theorem 2.3 shows that an exponentially small input perturbation can create a large separation in the network output. One might suspect that this only makes global Lipschitz certificates overly pessimistic, without leading to an actual change in the predicted label. The next two informal consequences show that this is not the case. The same construction can be converted into an explicit binary robustness failure. We defer the formal statements and proofs to Appendix D. We first show that a constant-margin decision flip can occur within an exponentially small adversarial radius. This provides a theoretical mechanism that helps explain why batch normalization can have an outsized effect on adversarial robustness [Galloway et al., 2019, Wang et al., 2022, Zeise and Vlaar, 2025].

Theorem 4.1 (Informal)**.** *For every depth m, consider the same batch-normalized network as in Theorem 2.3 and append a linear readout of Frobenius norm one. The resulting binary classifier has a clean input with score margin $\Omega_{k,\eta}(1)$, but the clean input can be misclassified after a perturbation of Frobenius norm at most $e^{-\Omega_{k,\eta}(m)}$.*

The readout is the hyperplane placed halfway between the two output features generated by the witness pair in the proof of Theorem 2.3. Thus, the result does not rely on a separate instability mechanism. It shows that the Lipschitz explosion itself can produce a genuine small-radius adversarial example once the learned representation is passed through a linear classifier. We now explain the mechanism behind the separation observed in Figure 1. The key point is that the adversarial perturbation is not only small in norm, but also rank-creating. This allows normalization to amplify the perturbation across layers, whereas perturbations that remain within the same low-rank branch do not experience the same amplification. This perspective is also consistent with the empirical geometry of standard image datasets, since MNIST, Fashion-MNIST, and CIFAR-10 images exhibit strong approximate low-rank structure in Figures 3 and 5.

Proposition 4.2 (Informal)**.** *In the same construction, the clean input is rank-deficient, while the adversarially perturbed input is full-rank and only $e^{-\Omega_{k,\eta}(m)}$ away. Moreover, the full-rank adversarial output is a constant distance away from the entire rank-deficient output branch of the*

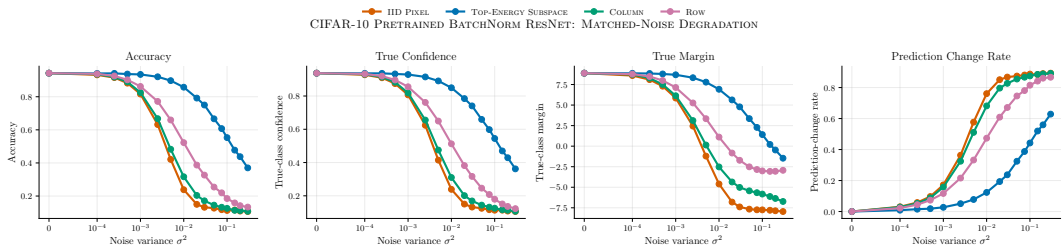


Figure 4: Effect of noise structure on a pretrained BatchNorm ResNet56 on CIFAR-10. Same-subspace perturbations are substantially less harmful than pixel-, row-, and column-wise noise.

network. Thus perturbations that leave the low-rank branch and perturbations that stay within it have fundamentally different behavior under normalization.

We empirically validate the prediction of Proposition 4.2 in both realistic and controlled settings. In Figure 4, we use a ResNet56 model pretrained on CIFAR-10, while in Figure 8 we use toy CNNs with batch normalization trained on MNIST and Fashion-MNIST. In all experiments, we vary the noise variance and compare perturbations with matched pre-clipping energy but different geometric structure. We consider four perturbation families: i.i.d. pixel-wise noise, i.i.d. column-wise noise, i.i.d. row-wise noise, and singular-value noise, which perturbs only the singular values and therefore keeps the image in the same singular subspace.

The results are consistent with the mechanism suggested by Proposition 4.2. Perturbations that introduce new directions or alter the rank structure of the input lead to a much stronger degradation in accuracy, confidence, and margin. In contrast, same-subspace perturbations are substantially less effective, even at comparable noise energy. This supports the view that the vulnerability induced by normalization is not determined only by perturbation size, but also by whether the perturbation moves the input away from its underlying low-rank structure. We defer the detailed experimental setup and additional results to Appendix E.

5 Conclusion

We showed that normalization layers can induce exponential growth in both input and parameter Lipschitz constants, even under strong per-layer norm control and already with linear activations. This Lipschitz explosion changes the interpretation of several standard guarantees for deep normalized networks, making Lipschitz-based optimization, generalization, and robustness analyses potentially vacuous in worst-case regimes. Our results also identify a rank-separation mechanism behind adversarial sensitivity and show that this mechanism is reflected empirically. We hope these findings motivate new stability notions and analysis tools that better capture the behavior of modern normalized architectures.

Acknowledgement

The authors thank Suvrit Sra for introducing them to nonsmooth optimization and its literature.

A.S. and P.J. were partially supported by the ONR grant N00014-24-1-2470. Part of this research was performed while A.S. was visiting the Institute for Mathematical and Statistical Innovation (IMSI), which is supported by the National Science Foundation (Grant No. DMS-2425650).

References

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.

Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. *Advances in neural information processing systems*, 30, 2017.

Sébastien Bubeck and Mark Sellke. A universal law of robustness via isoperimetry. *Advances in Neural Information Processing Systems*, 34:28811–28822, 2021.

Tim Salimans and Durk P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *Advances in neural information processing systems*, 29, 2016.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Sergey Ioffe. Batch renormalization: Towards reducing minibatch dependence in batch-normalized models. *Advances in neural information processing systems*, 30, 2017.

Hadi Daneshmand, Amir Joudaki, and Francis Bach. Batch normalization orthogonalizes representations in deep random networks. *Advances in Neural Information Processing Systems*, 34: 4896–4906, 2021.

Susanna Lange, Kyle Helfrich, and Qiang Ye. Batch normalization preconditioning for neural network training. *Journal of Machine Learning Research*, 23(72):1–41, 2022.

Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

Allen A Goldstein. Optimization of lipschitz continuous functions. *Mathematical Programming*, 13 (1):14–22, 1977.

Jingzhao Zhang, Hongzhou Lin, Stefanie Jegelka, Suvrit Sra, and Ali Jadbabaie. Complexity of finding stationary points of nonconvex nonsmooth functions. In *International Conference on Machine Learning*, pages 11173–11182. PMLR, 2020.

Damek Davis, Dmitriy Drusvyatskiy, Yin Tat Lee, Swati Padmanabhan, and Guanghao Ye. A gradient sampling method with complexity guarantees for lipschitz functions in high and low dimensions. *Advances in neural information processing systems*, 35:6692–6703, 2022.

Michael Jordan, Guy Kornowski, Tianyi Lin, Ohad Shamir, and Manolis Zampetakis. Deterministic nonsmooth nonconvex optimization. In *The Thirty Sixth Annual Conference on Learning Theory*, pages 4570–4597. PMLR, 2023.

Guy Kornowski and Ohad Shamir. On the complexity of finding small subgradients in nonsmooth optimization. *arXiv preprint arXiv:2209.10346*, 2022.

Songyan Hou, Parnian Kassraie, Anastasis Kratsios, Andreas Krause, and Jonas Rothfuss. Instance-dependent generalization bounds via optimal transport. *Journal of Machine Learning Research*, 24 (349):1–51, 2023.

Angus Galloway, Anna Golubeva, Thomas Tanay, Medhat Moussa, and Graham W Taylor. Batch normalization is a cause of adversarial vulnerability. *arXiv preprint arXiv:1905.02161*, 2019.

Haotao Wang, Aston Zhang, Shuai Zheng, Xingjian Shi, Mu Li, and Zhangyang Wang. Removing batch normalization boosts adversarial training. In *International Conference on Machine Learning*, pages 23433–23445. PMLR, 2022.

Noam Zeise and Tiffany Joyce Vlaar. Batchnorm layers have an outsized effect on adversarial robustness. In *OPT 2025: Optimization for Machine Learning*, 2025.

Berkant Savas and Lars Eldén. Handwritten digit classification using higher order singular value decomposition. *Pattern recognition*, 40(3):993–1003, 2007.

Yangmuzi Zhang, Zhuolin Jiang, and Larry S Davis. Learning structured low-rank representations for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 676–683, 2013.

Shuhang Gu, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng. Weighted nuclear norm minimization with application to image denoising. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2862–2869, 2014.

Jun Xu, Lei Zhang, Wangmeng Zuo, David Zhang, and Xiangchu Feng. Patch group based nonlocal self-similarity prior learning for image denoising. In *Proceedings of the IEEE international conference on computer vision*, pages 244–252, 2015.

Hangfan Liu, Xinfeng Zhang, and Ruiqin Xiong. Content-adaptive low rank regularization for image denoising. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3091–3095. IEEE, 2016.

Amir Joudaki, Hadi Daneshmand, and Francis Bach. On the impact of activation and normalization in obtaining isometric embeddings at initialization. *Advances in Neural Information Processing Systems*, 36:39855–39875, 2023.

Alexandru Meterez, Amir Joudaki, Francesco Orabona, Alexander Immer, Gunnar Ratsch, and Hadi Daneshmand. Towards training without depth limits: Batch normalization without gradient explosion. In *The Twelfth International Conference on Learning Representations*, 2024.

Greg Yang, Jeffrey Pennington, Vinay Rao, Jascha Sohl-Dickstein, and Samuel S Schoenholz. A mean field theory of batch normalization. *arXiv preprint arXiv:1902.08129*, 2019.

Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. Rectifier nonlinearities improve neural network acoustic models. In *Proceedings of the 30th annual international conference on machine learning*. Atlanta, GA, 2013.

Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th annual international conference on machine learning*, pages 609–616, 2009.

Simone Scardapane, Danilo Comminiello, Amir Hussain, and Aurelio Uncini. Group sparse regularization for deep neural networks. *Neurocomputing*, 241:81–89, 2017.

Andre Martins and Ramon Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *International conference on machine learning*, pages 1614–1623. PMLR, 2016.

Mathieu Blondel, Olivier Teboul, Quentin Berthet, and Josip Djolonga. Fast differentiable sorting and ranking. In *International Conference on Machine Learning*, pages 950–959. PMLR, 2020.

Tianyi Lin, Zeyu Zheng, and Michael Jordan. Gradient-free methods for deterministic and stochastic nonsmooth nonconvex optimization. *Advances in Neural Information Processing Systems*, 35: 26160–26175, 2022.

Siyu Kong and AS Lewis. The cost of nonconvexity in deterministic nonsmooth optimization. *Mathematics of Operations Research*, 2023.

Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks. *Advances in neural information processing systems*, 30, 2017.

Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM journal on optimization*, 23(4):2341–2368, 2013.

Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points—online stochastic gradient for tensor decomposition. In *Conference on learning theory*, pages 797–842. PMLR, 2015.

Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M Kakade, and Michael I Jordan. How to escape saddle points efficiently. In *International conference on machine learning*, pages 1724–1732. PMLR, 2017.

Chi Jin, Praneeth Netrapalli, Rong Ge, Sham M Kakade, and Michael I Jordan. On nonconvex optimization for machine learning: Gradients, stochasticity, and saddle points. *Journal of the ACM (JACM)*, 68(2):1–29, 2021.

Max Simchowitz, Ahmed El Alaoui, and Benjamin Recht. On the gap between strict-saddles and true convexity: An omega (log d) lower bound for eigenvector approximation. *arXiv preprint arXiv:1704.04548*, 2017.

Hadi Daneshmand, Jonas Kohler, Aurelien Lucchi, and Thomas Hofmann. Escaping saddles with stochastic gradients. In *International Conference on Machine Learning*, pages 1155–1164. PMLR, 2018.

Frank H Clarke. Generalized gradients and applications. *Transactions of the American Mathematical Society*, 205:247–262, 1975.

Michael I Jordan, Tianyi Lin, and Manolis Zampetakis. On the complexity of deterministic nonsmooth and nonconvex optimization. *arXiv preprint arXiv:2209.12463*, 2022.

Ashok Cutkosky, Harsh Mehta, and Francesco Orabona. Optimal stochastic non-smooth non-convex optimization through online-to-non-convex conversion. In *International Conference on Machine Learning*, pages 6643–6670. PMLR, 2023.

A Related Works

Deep learning theory. Our study is motivated by the fact that depth is both powerful and difficult to manage. Deeper networks have driven many empirical gains, but they also tend to make optimization more fragile. Many standard tools in modern architectures were introduced to make very deep models easier to train, including normalization layers [Ioffe and Szegedy, 2015, Ba et al., 2016], residual skip connections [He et al., 2016], and modified activation functions [Klambauer et al., 2017]. In practice, these tools often stabilize training and make it possible to scale networks to much greater depths. We ask whether this practical stabilization can come with unanticipated costs for training, robustness, and statistical behavior. As a first step toward this broader question, in this work, we study batch normalization.

The Lipschitz constant of a neural network appears in several learning-theoretic analyses through Lipschitz- and margin-sensitive complexity measures. For example, Bartlett et al. [2017] prove margin-based generalization bounds whose complexity term depends on a spectral complexity of the network, given by the product of the layer spectral norms together with an additional correction factor. Similarly, Bubeck and Sellke [2021] study robustness through the Lipschitz constant, showing tradeoffs between interpolation, model size, and smoothness. These results make Lipschitz control a natural quantity to study when reasoning about generalization and robustness. Our contribution is complementary. We show that, in deep networks with normalization layers, the relevant input Lipschitz constant can itself grow exponentially with depth, even under strong per-layer norm control. Thus, Lipschitz-dependent generalization and robustness guarantees can become vacuous for deep normalized networks in the worst case.

Our analysis is closely related to mean-field studies of batch normalization. Yang et al. [2019] show that, in the infinite-width limit, gradient norms in batch-normalized networks can grow exponentially with depth. More recently, Meterez et al. [2024] showed that this explosion can be avoided for linear activations under suitable full-rank conditions on the input batch. Our results show that this conditioning requirement is essential. Even with linear activations and norm-controlled weights, inputs that are close to rank deficiency can lead to exponential growth of the worst-case input Lipschitz constant, and hence to large worst-case sensitivity. The mechanism is tied to the isotropizing effect of normalization layers. Batch normalization tends to push representations toward a more isotropic geometry [Daneshmand et al., 2021], but near rank-deficiency this same effect can amplify small perturbations into large output separations.

Smooth optimization. Dimension-free convergence guarantees have a long history in smooth optimization. For smooth nonconvex objectives, gradient descent can find an approximate first-order stationary point at a rate that does not depend on the ambient dimension [Nesterov, 2013, Ghadimi and Lan, 2013]. A large body of work has tried to go beyond first-order stationarity and reach second-order stationary points, where the gradient is small and there are no directions of significant negative curvature [Ge et al., 2015, Jin et al., 2017, 2021]. Under smoothness assumptions, such guarantees are possible in polynomial time. However, dimension-free second-order guarantees cannot hold in full generality. Simchowitz et al. [2017] show that some dependence on the dimension is unavoidable for general smooth functions. Motivated by this limitation, Daneshmand et al. [2018] identify noise conditions under which stochastic gradient methods can still achieve dimension-free convergence to second-order stationary points, using structure that is natural in deep learning losses. These results rely on smoothness of the objective, an assumption that is often violated in modern neural networks because of ReLU activations, and max-pooling operations. For this reason, we focus here on nonsmooth optimization, where the objective need not be differentiable everywhere.

Nonsmooth optimization. The study of stationarity for nonsmooth Lipschitz functions goes back to the work of Goldstein [1977], which introduced what are now known as Goldstein stationary points. Unlike in the smooth setting, exact first-order stationarity is not a finite-time target in general nonsmooth nonconvex optimization. This has led recent work to focus on relaxed notions such as (δ, ϵ)-Goldstein stationarity. In this direction, Zhang et al. [2020] gave one of the first non-asymptotic algorithms for finding such points, with complexity scaling polynomially in the Lipschitz constant L, the radius parameter $1/\delta$, and the accuracy $1/\epsilon$. More recently, Davis et al. [2022] developed a gradient-sampling approach under a standard oracle model, improving the dependence on ϵ and L in certain regimes, at the cost of an additional dependence on the dimension. While these guarantees are polynomial for a fixed Lipschitz constant L, this distinction is important in deep learning, where

L is not an external constant. It is induced by the network architecture and can grow with depth. Our results show that, for deep networks with normalization layers, this growth can be exponential. Consequently, nonsmooth optimization guarantees that are polynomial in L can become exponential in depth, even when they are formally dimension-free.

B Proofs and Details for Section 2

We begin with a proof sketch of Theorem 2.3, and then give the full proof of Theorems 2.3 and 2.6 together with the required machinery and lemmas.

B.1 Proof Sketch of Theorem 2.3

We specialize to the square orthogonal regime (3) and the recursion (2). Since $W_\ell \in \mathbb{O}(k)$, the linear maps are non-expansive, so any growth must come from the normalization. Our argument exploits a tension between (i) the ability of $\mathrm{BN}(\cdot)$ to *regularize* anisotropy when combined with orthogonal mixing, and (ii) the instability of this regularization near rank-deficient inputs.

We follow Joudaki et al. [2023] and track isotropy via the functional $I(\cdot)$ in Definition B.1. For normalized matrices $X = \mathrm{BN}(\cdot)$ we have $\mathrm{tr}(X^\top X) = k$ by (1), hence $I(X) = \det(X^\top X)^{1/k} \in (0, 1]$. A key input is the one-step lift inequality of Meterez et al. [2024]: Theorem B.3 shows that for $X = \mathrm{BN}(\cdot)$ and Haar-uniform $W \in \mathbb{O}(k)$,

$$\mathbb{E}[I(\mathrm{BN}(XW)) \,|\, X] \;\geq\; \frac{1}{1 - S(X)}\, I(X),$$

where $S(\cdot)$ is the spectral deviation from Definition B.2. Lemma B.4 provides a uniform gap $S(X) \geq s_{k,\eta} > 0$ whenever $I(X) \leq 1 - \eta$. Combining these yields a deterministic multiplicative lift: Corollary B.5 guarantees that, as long as $I(X) \leq 1-\eta$, we can choose W so that $I(\mathrm{BN}(XW)) \geq r_{k,\eta} I(X)$ with $r_{k,\eta} = (1 - s_{k,\eta})^{-1} > 1$. Iterating this choice gives the greedy geometric increase Corollary B.6, so after m layers we can enforce $I(X_m) \geq 1 - \eta$ provided the initial isotropy is at least $r_{k,\eta}^{-m}(1 - \eta)$.

We realize such an initialization while keeping it exponentially close to rank deficiency. As in the appendix proof of Theorem 2.3, we set $G_\varepsilon = (1 - \varepsilon)\mathbf{1}\mathbf{1}^\top + \varepsilon \mathbf{I}$ and choose X_0 with $X_0^\top X_0 = G_\varepsilon$. Then $\mathrm{diag}(G_\varepsilon) = \mathbf{I}$, so $X_0 = \mathrm{BN}(X_0)$, and $I(X_0) = \det(G_\varepsilon)^{1/k} \geq \varepsilon^{(k-1)/k}$. Choosing

$$\varepsilon \;=\; (1 - \eta)^{\frac{k}{k-1}}\, r_{k,\eta}^{-\frac{mk}{k-1}}$$

ensures $r_{k,\eta}^m I(X_0) \geq 1 - \eta$, hence (by Corollary B.6) we obtain $I(X_m) \geq 1 - \eta$ along the resulting trajectory. Near-isotropy then forces uniform conditioning: Lemma B.7 yields $\lambda_{\min}(X_m^\top X_m) \geq c_k I(X_m)^k \geq c_k(1 - \eta)^k =: \delta_{k,\eta}$.

To turn this into a Lipschitz lower bound, we compare X_0 to a rank-$(k-1)$ neighbor. Let $X_0 = U\Sigma V^\top$ be an SVD with singular values $\sigma_1 \geq \cdots \geq \sigma_k > 0$ and define the matrix $Y_0 := U\,\mathrm{diag}(\sigma_1, \ldots, \sigma_{k-1}, 0)V^\top$. Then $\mathrm{rank}(Y_0) < k$ and $\|X_0 - Y_0\|_F = \sigma_k = \sqrt{\varepsilon}$. Let $Y_t := F_t(Y_0)$ under the same weights (one may, if desired, interpret BN via the standard ε-regularization from Remark 2.1 to avoid degenerate columns; this does not change the bound). Since each layer is an invertible right-multiplication followed by column-wise rescaling, rank cannot increase, so $\mathrm{rank}(Y_t) < k$ for all t. Applying Lemma B.8 at depth m gives $\|X_m - Y_m\|_F^2 \geq \lambda_{\min}(X_m^\top X_m) \geq \delta_{k,\eta}$, and therefore

$$\mathrm{Lip}_{\mathcal{X}}(F_m) \;\geq\; \frac{\|F_m(X_0) - F_m(Y_0)\|_F}{\|X_0 - Y_0\|_F} \;\geq\; \sqrt{\frac{\delta_{k,\eta}}{\varepsilon}} \;=\; C_{k,\eta}\, r_{k,\eta}^{\frac{mk}{2(k-1)}} \;=\; C_{k,\eta} e^{\gamma_{k,\eta} m},$$

with $\gamma_{k,\eta} = \frac{k}{2(k-1)} \log r_{k,\eta}$ and $C_{k,\eta}$ as in Theorem 2.3. Finally, the orthogonal equivariance $\mathrm{BN}((QX)W) = Q\,\mathrm{BN}(XW)$ for $Q \in \mathbb{O}(k)$ implies $F_m(QX) = QF_m(X)$, so the same lower bound is witnessed by the uncountable family of pairs (QX_0, QY_0), as recorded at the end of the proof of Theorem 2.3.

B.2 Isotropy and a One-step Lift Inequality

We use the notion of isotropy from Joudaki et al. [2023][1], together with the one-step lift developed in Meterez et al. [2024], to study the dynamics of linear neural networks with normalization across layers. Let $X \in \mathbb{R}^{k\times k}$ and assume $X^\top X \succ 0$.

Definition B.1 (Isotropy functional)**.** *Define*

$$I(X) := \frac{\det(X^\top X)^{1/k}}{\operatorname{tr}(X^\top X)/k} \in (0,1].$$

The quantity $I(X)$ is scale-invariant and satisfies $I(X) = 1$ if and only if $X^\top X$ is a scalar multiple of the identity. In this case, we say that we have "perfect isotropy". If $X = \mathrm{BN}(\cdot)$ (so that $\operatorname{tr}(X^\top X) = k$), then $I(X) = \det(X^\top X)^{1/k}$.

Definition B.2 (Spectral deviation)**.** *Suppose $X = \mathrm{BN}(\cdot)$ and let $\lambda_1, \dots, \lambda_k$ be the eigenvalues of $X^\top X$ (so $\sum_{i=1}^k \lambda_i = \operatorname{tr}(X^\top X) = k$). Define*

$$S(X) := \frac{1}{2k^2(k+2)} \sum_{i=1}^{k} (\lambda_i - 1)^2.$$

Theorem B.3 (One-step isotropy lift [Meterez et al., 2024, Theorem A.4])**.** *Let $X = \mathrm{BN}(\cdot)$ and let W be Haar-uniform on $\mathbb{O}(k)$. Then*

$$\mathbb{E}_W\big[I(\mathrm{BN}(XW)) \,\big|\, X\big] \;\geq\; \frac{1}{1 - S(X)}\, I(X).$$

B.3 A Uniform Multiplicative Lift Below an Isotropy Cap

Fix $\eta \in (0,1)$ and define the compact set of eigenvalue vectors

$$K_\eta := \Big\{\lambda \in [0,\infty)^k : \sum_{i=1}^{k} \lambda_i = k,\ \Big(\prod_{i=1}^{k} \lambda_i\Big)^{1/k} \leq 1 - \eta\Big\}.$$

Consider $S(\lambda) := \frac{1}{2k^2(k+2)} \sum_{i=1}^k (\lambda_i - 1)^2$ from Definition B.2.

We now prove a uniform spectral gap away from $I(X) = 1$.

Lemma B.4. *Let $\eta \in (0,1)$ and set $s_{k,\eta} := \min_{\lambda \in K_\eta} S(\lambda)$. Then $s_{k,\eta} > 0$. Moreover, if $X = \mathrm{BN}(\cdot)$ and $I(X) \leq 1 - \eta$, then $S(X) \geq s_{k,\eta}$.*

Proof. K_η is nonempty, closed, and bounded, hence compact, and S is continuous, so the minimum is attained. Also $S(\lambda) = 0$ iff $\lambda_1 = \cdots = \lambda_k = 1$, but that vector is not in K_η since its geometric mean is $1 > 1 - \eta$. Thus $s_{k,\eta} > 0$. The final statement follows because $S(X) = S(\lambda(X^\top X))$ and, for $X = \mathrm{BN}(\cdot)$, $I(X) = \big(\prod_{i=1}^k \lambda_i\big)^{1/k}$. □

In turn, we can convert this result into a deterministic multiplicative lift for the isotropy functional $I(X)$, as long as it is below $1 - \eta$.

Corollary B.5. *Fix $\eta \in (0,1)$ and define*

$$r_{k,\eta} := \frac{1}{1 - s_{k,\eta}} > 1.$$

If $X = \mathrm{BN}(\cdot)$ and $I(X) \leq 1 - \eta$, then there exists an orthogonal matrix $W \in \mathbb{O}(k)$ such that

$$I(\mathrm{BN}(XW)) \geq r_{k,\eta}\, I(X).$$

Proof. By Lemma B.4, $S(X) \geq s_{k,\eta}$. Theorem B.3 then gives $\mathbb{E}_W[I(\mathrm{BN}(XW)) \mid X] \geq r_{k,\eta} I(X)$. Since $I(\mathrm{BN}(XW)) \geq 0$, there must exist a realization of W achieving at least the expectation. □

[1] In Joudaki et al. [2023], this quantity is referred to as *isometry*; here, we use the term *isotropy* for terminological clarity and to better reflect its role in our analysis of Lipschitz constants.

Consequently, we can show that the isotropy functional $I(X)$ increases geometrically until X reaches near-isotropy. In the proof, we choose the weight matrices greedily according to Corollary B.5.

Corollary B.6. *Fix $\eta \in (0,1)$ and let $r_{k,\eta} > 1$ be as above. Let $X_0 = \mathrm{BN}(\cdot)$ and choose $W_1, \dots, W_m \in \mathbb{O}(k)$ greedily as follows: for each $t \geq 1$, if $I(X_{t-1}) \leq 1-\eta$ choose W_t satisfying Corollary B.5, and set $X_t := \mathrm{BN}(X_{t-1}W_t)$. Then for every $t \geq 0$,*

$$I(X_t) \;\geq\; \min\{r_{k,\eta}^t\, I(X_0),\; 1-\eta\}.$$

Proof. If $I(X_{t-1}) \leq 1-\eta$, then by construction $I(X_t) \geq r_{k,\eta} I(X_{t-1})$. Once $I(X_{t-1}) > 1-\eta$ we do not require any further increase. An induction yields the claimed bound. □

B.4 Near-isotropy Forces a Floor on $\lambda_{\min}$

A useful basic result is the inequality that connects the minimum eigenvalue in the spectrum of $X^\top X$ to the isotropy functional $I(X)$.

Lemma B.7. *Let $X = \mathrm{BN}(\cdot)$ and let $\lambda_1 \geq \cdots \geq \lambda_k$ be the eigenvalues of $X^\top X$ (so $\sum_{i=1}^k \lambda_i = k$ and $\prod_{i=1}^k \lambda_i = I(X)^k$). Then*

$$\lambda_{\min}(X^\top X) = \lambda_k \;\geq\; c_k\, I(X)^k, \qquad c_k := \Big(\frac{k-1}{k}\Big)^{k-1}.$$

Proof. Set $\beta := \lambda_k$ and note $\sum_{i=1}^{k-1} \lambda_i = k - \beta$. By AM–GM, for fixed sum $k-\beta$ the product of the first $k-1$ eigenvalues is maximized when they are equal, hence

$$\prod_{i=1}^{k-1} \lambda_i \;\leq\; \Big(\frac{k-\beta}{k-1}\Big)^{k-1} \;\leq\; \Big(\frac{k}{k-1}\Big)^{k-1}.$$

Therefore

$$I(X)^k = \prod_{i=1}^{k} \lambda_i = \beta \prod_{i=1}^{k-1} \lambda_i \;\leq\; \beta\Big(\frac{k}{k-1}\Big)^{k-1},$$

which rearranges to $\beta \geq ((k-1)/k)^{k-1} I(X)^k$. □

B.5 Distance to a Rank-deficient Trajectory

We next use a simple geometric fact that turns the rank mismatch between the two trajectories into a quantitative separation. In our construction, one trajectory becomes well-conditioned, while the other remains rank-deficient throughout the network. The following lemma shows that any rank-deficient matrix must stay at least a smallest-singular-value distance away from a well-conditioned full-rank matrix.

Lemma B.8. *Let $X, Y \in \mathbb{R}^{k\times k}$ and suppose that $\mathrm{rank}(Y) < k$. Then*

$$\|X - Y\|_F^2 \;\geq\; \lambda_{\min}(X^\top X).$$

Proof. Since $\mathrm{rank}(Y) < k$ there exists a unit vector $v \in \ker(Y)$. Then

$$\|X-Y\|_F^2 = \mathrm{tr}\big((X-Y)^\top (X-Y)\big) \geq v^\top (X-Y)^\top (X-Y) v = \|(X-Y)v\|_2^2 = \|Xv\|_2^2.$$

Finally, $\|Xv\|_2^2 = v^\top X^\top X v \geq \lambda_{\min}(X^\top X)$. □

B.6 Exponential growth of $\mathrm{Lip}_{\mathcal{X}}(F_m)$ with depth

Theorem 2.3 (Exponential lower bound on $\mathrm{Lip}_{\mathcal{X}}(F_m)$)**.** *Fix $k \geq 2$ and constant $\eta \in (0,1)$. There exist constants $C_{k,\eta} > 0$ and $\gamma_{k,\eta} > 0$ such that, for every depth $m \in \mathbb{N}$, one can choose orthogonal weights $W_1, \dots, W_m \in \mathbb{O}(k)$ with $\|W_\ell\|_2 = 1$ for all $\ell \in [m]$, for which the depth-m map F_m defined in (2)–(3) satisfies*

$$\mathrm{Lip}_{\mathcal{X}}(F_m) \;\geq\; C_{k,\eta}\, e^{\gamma_{k,\eta} m},$$

where constants $C_{k,\eta}$ and $\gamma_{k,\eta}$ depend on k, η. Moreover, for this same fixed choice of weights, the lower bound is witnessed by an uncountable family of input pairs.

Proof. Fix $m \in \mathbb{N}$ and let $r := r_{k,\eta} > 1$ from Corollary B.5. Define

$$\delta_{k,\eta} := c_k(1-\eta)^k, \qquad \gamma_{k,\eta} := \frac{k}{2(k-1)} \log r, \qquad C_{k,\eta} := \sqrt{\delta_{k,\eta}}\,(1-\eta)^{-\frac{k}{2(k-1)}}.$$

Step 1: choose an initialization X_0 with controlled anisotropy. Let

$$G_\varepsilon := (1-\varepsilon)\,\mathbf{1}\mathbf{1}^\top + \varepsilon\,\mathbf{I} \in \mathbb{R}^{k\times k}, \qquad \varepsilon \in (0,1].$$

Then $\mathrm{diag}(G_\varepsilon) = \mathbf{I}$ and the eigenvalues of G_ε are $\lambda_1 = k-(k-1)\varepsilon$ and $\lambda_2 = \cdots = \lambda_k = \varepsilon$. Choose any $X_0 \in \mathbb{R}^{k\times k}$ such that $X_0^\top X_0 = G_\varepsilon$ (e.g. take $X_0 = L^\top$ where $G_\varepsilon = LL^\top$ is a Cholesky factorization). Since $\mathrm{diag}(G_\varepsilon) = \mathbf{I}$, every column of X_0 has norm 1, hence $X_0 = \mathrm{BN}(X_0)$.

Moreover,

$$I(X_0) = \det(G_\varepsilon)^{1/k} = \left(\varepsilon^{k-1}(k-(k-1)\varepsilon)\right)^{1/k} \;\geq\; \varepsilon^{\frac{k-1}{k}},$$

because $k-(k-1)\varepsilon \geq 1$ for $\varepsilon \in (0,1]$.

Step 2: pick ε so that m greedy steps reach near-isotropy. Choose

$$\varepsilon := (1-\eta)^{\frac{k}{k-1}}\, r^{-\frac{mk}{k-1}}.$$

Then $r^m I(X_0) \geq r^m \varepsilon^{(k-1)/k} = 1-\eta$.

Now choose $W_1, \ldots, W_m$ greedily as in Corollary B.6 and set $X_t := \mathrm{BN}(X_{t-1}W_t)$. By Corollary B.6, $I(X_m) \geq 1-\eta$.

Step 3: eigenvalue floor after reaching near-isotropy. Lemma B.7 yields

$$\lambda_{\min}(X_m^\top X_m) \;\geq\; c_k\, I(X_m)^k \;\geq\; c_k(1-\eta)^k = \delta_{k,\eta}.$$

Step 4: a rank-deficient companion Y_0 and distance growth. Let $X_0 = U\Sigma V^\top$ be an SVD with singular values $\sigma_1 \geq \cdots \geq \sigma_k > 0$. Define the rank-$(k-1)$ matrix

$$Y_0 := U\,\mathrm{diag}(\sigma_1, \ldots, \sigma_{k-1}, 0)V^\top.$$

Then $\mathrm{rank}(Y_0) < k$ and

$$\|X_0 - Y_0\|_F = \sigma_k = \sqrt{\lambda_{\min}(X_0^\top X_0)} = \sqrt{\varepsilon}.$$

Let $Y_t := F_t(Y_0)$. Since each layer is multiplication by an invertible matrix (W_t) followed by multiplication by a diagonal matrix (coming from BN), the rank cannot increase, hence $\mathrm{rank}(Y_t) < k$ for all t.

Applying Lemma B.8 at depth m gives

$$\|X_m - Y_m\|_F^2 \geq \lambda_{\min}(X_m^\top X_m) \geq \delta_{k,\eta}.$$

Therefore

$$\mathrm{Lip}_{\mathcal{X}}(F_m) \;\geq\; \frac{\|F_m(X_0) - F_m(Y_0)\|_F}{\|X_0 - Y_0\|_F} \;\geq\; \sqrt{\frac{\delta_{k,\eta}}{\varepsilon}} = C_{k,\eta}\, r^{\frac{mk}{2(k-1)}} = C_{k,\eta}\, e^{\gamma_{k,\eta} m}.$$

Uncountably many witness pairs for the same weights. For any orthogonal $Q \in \mathbb{O}(k)$, we have for all X and all orthogonal W,

$$\mathrm{BN}((QX)W) = Q\;\mathrm{BN}(XW),$$

because $((QX)W)^\top((QX)W) = W^\top X^\top Q^\top QXW = W^\top X^\top XW$. Hence $F_m(QX) = QF_m(X)$ by induction. Consequently, the ratio $\|F_m(QX_0) - F_m(QY_0)\|_F / \|QX_0 - QY_0\|_F$ equals the ratio for (X_0, Y_0) for every $Q \in \mathbb{O}(k)$. Since $\mathbb{O}(k)$ is uncountable, this yields uncountably many witness pairs. □

Corollary 2.4. *Fix $\eta \in (0,1)$ and set $k = 2$. There exist constants $C_\eta > 0$ and $\Gamma_\eta > 0$ such that, for every integer $d \geq 4$, there is a depth-m network with at most d scalar weight parameters whose input–output map satisfies*

$$\mathrm{Lip}_{\mathcal{X}}(F_m) \;\geq\; C_\eta\, e^{\Gamma_\eta d}.$$

Proof. When $k = 2$, each layer contains 4 scalar parameters if the weight matrix is counted as an unconstrained 2×2 matrix. Thus, a depth-m network has $4m$ scalar parameters. Given a budget $d \geq 4$, choose $m = \lfloor d/4 \rfloor$, so that $4m \leq d$. Applying Theorem 2.3 with $k = 2$ gives

$$\mathrm{Lip}_{\mathcal{X}}(F_m) \geq C_{2,\eta} e^{\gamma_{2,\eta}\lfloor d/4 \rfloor} \geq C_{2,\eta} e^{-\gamma_{2,\eta}} e^{(\gamma_{2,\eta}/4)d}.$$

The claim follows by setting

$$C_\eta := C_{2,\eta} e^{-\gamma_{2,\eta}}, \qquad \Gamma_\eta := \gamma_{2,\eta}/4.$$

□

B.7 Exponential growth of $\mathrm{Lip}_\Theta(\mathcal{F}_m)$ with depth

Theorem 2.6 (Exponential lower bound on $\mathrm{Lip}_\Theta(\mathcal{F}_m)$)**.** *Fix $k \geq 2$ and constant $\eta \in (0,1)$. There exist constants $\widetilde{C}_{k,\eta} > 0$ and $\widetilde{\gamma}_{k,\eta} > 0$ depending on k and η such that, for every depth $m \geq 2$, there is a depth-m network of the form (2), with $\|W_\ell\|_2 \leq 1$ for all $\ell \in [m]$, for which the fixed-input parameter-to-output map $\mathcal{F}_m(\theta) := F_m^\theta(X_{\mathrm{in}})$ with $X_{\mathrm{in}} = \mathbf{I}_k$ satisfies*

$$\mathrm{Lip}_\Theta(\mathcal{F}_m) \;\geq\; \widetilde{C}_{k,\eta}\, e^{\widetilde{\gamma}_{k,\eta} m}.$$

Proof. Apply Theorem 2.3 with depth $m-1$ to obtain orthogonal matrices $\widetilde{W}_2, \ldots, \widetilde{W}_m \in \mathbb{O}(k)$ and matrices $X_0, Y_0 \in \mathbb{R}^{k\times k}$ with $\mathrm{rank}(Y_0) < k$ such that the corresponding depth-$(m-1)$ map

$$G_{m-1}(Z) := \mathrm{BN}(\cdots \mathrm{BN}(\mathrm{BN}(Z\widetilde{W}_2)\widetilde{W}_3)\cdots \widetilde{W}_m)$$

satisfies

$$\|G_{m-1}(X_0) - G_{m-1}(Y_0)\|_F \;\geq\; \sqrt{\delta_{k,\eta}}, \qquad \text{and} \qquad \|X_0 - Y_0\|_F = \sqrt{\varepsilon},$$

where ε is exponentially small in m as in the proof of Theorem 2.3.

Now consider a depth-m network with input $X_{\mathrm{in}} = \mathbf{I}_k$ and weights

$$W_1 := \alpha X_0, \qquad W_1' := \alpha Y_0, \qquad W_\ell := \widetilde{W}_\ell \;\; (\ell = 2, \ldots, m),$$

where $\alpha > 0$ is chosen so that $\|W_1\|_2, \|W_1'\|_2 \leq 1$. Since column-wise batch normalization is scale-invariant,

$$\mathrm{BN}(\alpha Z) = \alpha Z\,(D(\alpha Z))^{-1/2} = \alpha Z\,(\alpha^2 D(Z))^{-1/2} = \mathrm{BN}(Z),$$

we have $\mathrm{BN}(X_{\mathrm{in}}W_1) = \mathrm{BN}(X_0)$ and $\mathrm{BN}(X_{\mathrm{in}}W_1') = \mathrm{BN}(Y_0)$. Moreover, $\mathrm{rank}(\mathrm{BN}(Y_0)) < k$.

Let $\theta = (W_1, \ldots, W_m)$ and $\theta' = (W_1', W_2, \ldots, W_m)$. Then

$$\mathcal{F}_m(\theta) = G_{m-1}(\mathrm{BN}(X_0)) \qquad \text{and} \qquad \mathcal{F}_m(\theta') = G_{m-1}(\mathrm{BN}(Y_0)).$$

The second trajectory remains rank-deficient at every depth, so Lemma B.8 (applied at depth $m-1$) and the conditioning lower bound from the proof of Theorem 2.3 yield

$$\|\mathcal{F}_m(\theta) - \mathcal{F}_m(\theta')\|_F \;\geq\; \sqrt{\delta_{k,\eta}}.$$

On the other hand,

$$\|\theta - \theta'\|_2 = \|W_1 - W_1'\|_F = \alpha\|X_0 - Y_0\|_F = \alpha\sqrt{\varepsilon}.$$

Therefore

$$\mathrm{Lip}_\Theta(\mathcal{F}_m) \;\geq\; \frac{\|\mathcal{F}_m(\theta) - \mathcal{F}_m(\theta')\|_F}{\|\theta - \theta'\|_2} \;\geq\; \frac{\sqrt{\delta_{k,\eta}}}{\alpha\sqrt{\varepsilon}} \;=\; \widetilde{C}_{k,\eta} e^{\widetilde{\gamma}_{k,\eta} m},$$

since ε is exponentially small in m (same as in Theorem 2.3). □

Corollary 2.8. *Let $\mathcal{F}_m$ be the parameter-to-output map constructed in Theorem 2.6. There exists a realizable target parameter $\theta^\star \in \Theta$ such that the objective f defined in (4) satisfies*

$$\mathrm{Lip}(f) \;\geq\; \widetilde{C}_\eta e^{\widetilde{\Gamma}_\eta d},$$

up to the constants in Corollary 2.7.

Proof. In the construction of Theorem 2.6, let $\theta^\star$ be one of the two parameter choices witnessing the lower bound on $\mathrm{Lip}_\Theta(\mathcal{F}_m)$, and let θ be the other one. Since the target is realizable, $f(\theta^\star) = 0$. Therefore,

$$\mathrm{Lip}(f) \geq \frac{|f(\theta) - f(\theta^\star)|}{\|\theta - \theta^\star\|_2} = \frac{|f(\theta)|}{\|\theta - \theta^\star\|_2} = \frac{\|\mathcal{F}_m(\theta) - \mathcal{F}_m(\theta^\star)\|_F}{\|\theta - \theta^\star\|_2}.$$

The right-hand side is exactly the same separation ratio used in the proof of Theorem 2.6. Applying Corollary 2.7 gives the claimed exponential lower bound. □

B.8 Extension to Normalization with Mean Subtraction

The results of Sections 2.2 and 2.3 extend to batch normalization with mean subtraction. Under our notation in Section 2, the centered normalization map is defined, for $X \in \mathbb{R}^{n\times k}$, by

$$\mathrm{BN}_{\mathrm{ctr}}(X) \coloneqq \mathrm{BN}\left(\left(\mathbf{I}_n - \frac{1}{n}\mathbf{1}\mathbf{1}^\top\right) X\right),$$

whenever every centered column is nonzero. This corresponds to the standard batch-variance normalization with a fixed affine scale of $1/\sqrt{n}$.

We now try to embed our hard instances into the centered subspace. Suppose $n \geq k+1$, and choose $U \in \mathbb{R}^{n\times k}$ satisfying

$$U^\top U = \mathbf{I}_k, \qquad \mathbf{1}^\top U = 0.$$

Such a matrix exists since the orthogonal complement of $\mathbf{1}$ in $\mathbb{R}^n$ has dimension $n - 1 \geq k$. Left multiplication by U is an isometry and maps every matrix into the subspace of column-centered matrices. In particular, for every $Z \in \mathbb{R}^{k\times k}$,

$$(UZ)^\top (UZ) = Z^\top Z,$$

so the embedding preserves rank, Gram eigenvalues, column norms, and Frobenius distances.

Moreover, since $\mathbf{1}^\top U = 0$, mean subtraction leaves UZW unchanged. Consequently, for every Z and W for which normalization is defined,

$$\mathrm{BN}_{\mathrm{ctr}}(UZW) = UZWD(ZW)^{-1/2} = U\,\mathrm{BN}(ZW). \tag{5}$$

Thus, centered normalization reproduces the original normalization dynamics exactly on the embedded subspace.

Let $\overline{F}_m$ denote the depth-m network obtained by replacing each normalization layer of F_m with $\mathrm{BN}_{\mathrm{ctr}}$, while keeping the same weight matrices. Applying (5) inductively yields

$$\overline{F}_m(UZ) = UF_m(Z).$$

In particular, applying this identity to the witness pair (X_0, Y_0) from Theorem 2.3 gives

$$\frac{\|\overline{F}_m(UX_0) - \overline{F}_m(UY_0)\|_F}{\|UX_0 - UY_0\|_F} = \frac{\|F_m(X_0) - F_m(Y_0)\|_F}{\|X_0 - Y_0\|_F}.$$

Hence the exponential input Lipschitz lower bound carries over with exactly the same initial perturbation size and final output separation. The Gram-eigenvalue bounds underlying the construction are also preserved by the embedding.

The parameter Lipschitz lower bound follows from the same identity. Indeed, replacing the fixed input $\mathbf{I}_k$ in Theorem 2.6 with U, and using the same two weight tuples, preserves both their parameter distance and the corresponding output separation. Since the weight matrices remain $k \times k$, the parameter count is unchanged.

Finally, the robustness construction of Theorem 4.1 extends directly to the centered network. Using the readout UA_m and the same offset b_m, we have $\|UA_m\|_F = 1$ and

$$\langle UA_m, \overline{F}_m(UZ)\rangle_F + b_m = \langle A_m, F_m(Z)\rangle_F + b_m.$$

Thus, the embedded clean and adversarial inputs achieve exactly the same classification margins and perturbation distance as in Theorem 4.1. In particular, the exponential input and parameter sensitivity bounds, as well as the exponentially small adversarial radius, hold for mean-subtracted batch normalization.

B.9 Extension to ReLU Activations

Our construction in Appendix B.8 also extends to networks with centered batch normalization followed by ReLU activations. We use the standard affine shift of batch normalization, fixed to 2 in every channel, and define

$$\widetilde{f}_W(H) \coloneqq \mathrm{ReLU}\big(\mathrm{BN}_{\mathrm{ctr}}(HW) + 2\mathbf{1}\mathbf{1}^\top\big),$$

where ReLU acts entrywise and $\mathbf{1}\mathbf{1}^\top \in \mathbb{R}^{n\times k}$ denotes the all-ones matrix. Due to normalization layers, each column of $\mathrm{BN}_{\mathrm{ctr}}(HW)$ has unit Euclidean norm, so every entry lies in $[-1, 1]$. Consequently, the affine shift makes every entry strictly positive, and ReLU acts as the identity.

The key observation is that mean subtraction eliminates any matrix with identical rows, even after multiplication by a weight matrix. Combining this property with (5), we obtain

$$\begin{aligned}\widetilde{f}_W(UZ) &= Uf_W(Z) + 2\mathbf{1}\mathbf{1}^\top,\\ \widetilde{f}_W(UZ + 2\mathbf{1}\mathbf{1}^\top) &= Uf_W(Z) + 2\mathbf{1}\mathbf{1}^\top.\end{aligned}$$

Thus, the affine shift introduced by one layer is removed by the centering operation in the next layer, while the centered component follows exactly the dynamics of the original network.

Writing $\widetilde{F}_m = \widetilde{f}_{W_m} \circ \cdots \circ \widetilde{f}_{W_1}$, induction yields

$$\widetilde{F}_m(UZ) = UF_m(Z) + 2\mathbf{1}\mathbf{1}^\top.$$

In particular, for the witness pair (X_0, Y_0),

$$\widetilde{F}_m(UX_0) - \widetilde{F}_m(UY_0) = U\big(F_m(X_0) - F_m(Y_0)\big).$$

Since $U^\top U = \mathbf{I}_k$, the input and output distances are preserved exactly. The exponential input Lipschitz lower bound therefore holds with the same witness ratio. Likewise, using the fixed input U and the weight tuples from Theorem 2.6 preserves their parameter distance and output separation, yielding the same parameter Lipschitz lower bound. All affine shifts are fixed, so the number of free weight parameters is unchanged.

Finally, the robustness construction transfers under the same readout UA_m and offset b_m. Indeed, since $\mathbf{1}^\top U = 0$,

$$\begin{aligned}\langle UA_m, \widetilde{F}_m(UZ)\rangle_F + b_m &= \langle UA_m, UF_m(Z) + 2\mathbf{1}\mathbf{1}^\top\rangle_F + b_m\\ &= \langle A_m, F_m(Z)\rangle_F + b_m.\end{aligned}$$

Hence the clean and adversarial inputs keep the same score margins and perturbation distance as in Theorem 4.1.

We conclude that the exponential input and parameter Lipschitz lower bounds, together with the exponentially small adversarial radius, are also attained by networks with centered batch normalization and ReLU activations.

B.10 Embedding the Construction at Larger Widths

We show that the exponential lower bounds can be achieved with constants independent of width by embedding the width-two construction into networks of arbitrary width $k \geq 2$.

Let X_0, Y_0 and $W_1, \ldots, W_m$ be the witnesses from Theorem 2.3 for $k = 2$. For any $k \geq 2$, consider the block-diagonal matrices

$$\mathrm{diag}(X_0, \mathbf{I}_{k-2}), \qquad \mathrm{diag}(Y_0, \mathbf{I}_{k-2}), \qquad \mathrm{diag}(W_\ell, \mathbf{I}_{k-2}), \quad \ell \in [m],$$

where the identity blocks are omitted when $k = 2$. Since normalization acts columnwise and the additional columns have unit norm, the two trajectories at depth t are precisely $\mathrm{diag}(X_t, \mathbf{I}_{k-2})$ and $\mathrm{diag}(Y_t, \mathbf{I}_{k-2})$. Their input distance and final output separation are therefore identical to those of the width-two construction. Consequently, the input Lipschitz lower bound

$$\mathrm{Lip}_X(F_m) \geq C_{2,\eta} e^{\gamma_{2,\eta} m}$$

holds for every $k \geq 2$, with constants independent of width.

Similarly, applying the same embedding to the parameter witnesses in Theorem 2.6, with fixed input $\mathbf{I}_k$, preserves their parameter distance, output separation, and unit operator norms. Thus, the parameter Lipschitz lower bound also holds uniformly over width, with the constants of the width-two construction.

These bounds have width-independent exponential rates in depth. When all entries of the weight matrices are counted, however, the parameter dimension is $d = mk^2$, and the input lower bound becomes

$$\mathrm{Lip}_X(F_m) \geq C_{2,\eta} \exp\Big(\frac{\gamma_{2,\eta}}{k^2} d\Big).$$

C Proofs and Details for Section 3

C.1 Background on Goldstein Stationarity and First-order Oracle

Notation and Background. Let $f : \mathbb{R}^d \to \mathbb{R}$ be (locally) L-Lipschitz with respect to $\|\cdot\|_2$:

$$|f(x) - f(y)| \leq L\|x - y\|_2, \qquad \forall x, y \in \mathbb{R}^d.$$

Let $\partial f(x)$ denote the *Clarke subdifferential* at x, which coincides with $\{\nabla f(x)\}$ when f is differentiable at x and agrees with the usual convex subdifferential when f is convex. The canonical solution concept studied in nonsmooth nonconvex optimization is that of a *Clarke stationary point* [Clarke, 1975]. A point $x \in \mathbb{R}^d$ is ε-*Clarke stationary* if $\mathrm{dist}(0, \partial f(x)) \leq \varepsilon$, where $\mathrm{dist}(0, Q) := \min_{g \in \partial f(x)} \|g\|_2$.

Zhang et al. [2020] showed that for general nonconvex Lipschitz functions, Clarke stationarity cannot be guaranteed in a finite number of oracle queries in the worst case, even under very strong first-order oracle access. This result shows that the consideration of more relaxed solution concepts, in favor of tractability, is inevitable.

Goldstein Stationarity. A classical and now-standard relaxation is due to Goldstein [1977]. Fix $\delta > 0$. The key idea is to *convexify the subdifferential over a neighborhood*:

$$\partial_\delta f(x) \;:=\; \mathrm{Conv}\Big(\bigcup_{y:\ \|y-x\|_2 < \delta} \partial f(y) \Big).$$

We say that x is (δ, ε)-*Goldstein stationary* if

$$\mathrm{dist}\big(0, \partial_\delta f(x)\big) \;\leq\; \varepsilon.$$

This definition recovers the smooth optimality condition when f is differentiable and δ is sufficiently small. The recent breakthrough of Zhang et al. [2020] highlighted the tractability of this solution concept in nonsmooth settings, and subsequent works have further developed and studied it extensively [Davis et al., 2022, Lin et al., 2022, Jordan et al., 2022, 2023, Kong and Lewis, 2023, Cutkosky et al., 2023].

Oracle model. We consider the standard first-order oracle model used in the nonsmooth optimization literature. At each query point x_t, the oracle reveals the function value $f(x_t)$ together with subgradient information at x_t (e.g., a Clarke subgradient $g_t \in \partial f(x_t)$ or even the full set $\partial f(x_t)$). The lower bounds stated in the paper holds even when the oracle returns the entire set $\partial f(x_t)$.

C.2 Proofs and Details for Section 3.1

Theorem C.1 ([Kornowski and Shamir, 2022, Theorem 3]). *For any $\varepsilon < 1$ and any $\delta \le \frac{1}{12\varepsilon}$, the following holds. For every randomized first-order algorithm $\mathcal{A}$, there exists a convex 1-Lipschitz function $f : \mathbb{R}^d \to \mathbb{R}$, with $d = \widetilde{\mathcal{O}}(1/\varepsilon^6)$, such that $f(x_1) - \inf_x f(x) \le 1$, but*

$$\Pr_{\mathcal{A}}\Big[\exists\, t \le T \text{ such that } x_t \text{ is } (\delta,\varepsilon)\text{-Goldstein stationary}\Big] < \frac{1}{3},$$

unless $T = \Omega(1/\varepsilon^2)$.

We now immediately show that no randomized algorithm can avoid a polynomial dependence on L.

Corollary 3.1 (Formal). *Fix $L > 0$ and $\varepsilon \in (0, L)$, and suppose that $\delta \le \frac{L}{12\varepsilon}$. Then, for any randomized first-order algorithm $\mathcal{A}$, there exists a convex L-Lipschitz function $F : \mathbb{R}^d \to \mathbb{R}$, with $d = \widetilde{\mathcal{O}}((L/\varepsilon)^6)$, such that $F(x_1) - \inf_x F(x) \le L$, and*

$$\Pr_{\mathcal{A}}\Big[\exists\, t \le T \text{ such that } x_t \text{ is } (\delta,\varepsilon)\text{-Goldstein stationary}\Big] < \frac{2}{3}$$

unless

$$T = \Omega\left(\frac{L^2}{\varepsilon^2}\right).$$

In particular, in the worst case, any randomized first-order method needs $\Omega(L^2/\varepsilon^2)$ oracle calls to find a (δ,ε)-Goldstein stationary point with probability at least $2/3$.

Proof. Set $\varepsilon' := \varepsilon/L$. Since $\varepsilon \in (0, L)$, we have $\varepsilon' \in (0, 1)$. Applying Theorem C.1 with accuracy ε' gives a convex 1-Lipschitz function $f : \mathbb{R}^d \to \mathbb{R}$, with $d = \widetilde{\mathcal{O}}(1/\varepsilon'^6) = \widetilde{\mathcal{O}}((L/\varepsilon)^6)$, such that $f(x_1) - \inf_x f(x) \le 1$, and no randomized first-order algorithm can find a (δ,ε')-Goldstein stationary point with constant probability unless

$$T = \Omega(1/\varepsilon'^2).$$

The condition on δ becomes

$$\delta \le \frac{1}{12\varepsilon'} = \frac{L}{12\varepsilon}.$$

Now define $F := Lf$. Then F is convex and L-Lipschitz, and

$$F(x_1) - \inf_x F(x) = L\big(f(x_1) - \inf_x f(x)\big) \le L.$$

Moreover, by positive homogeneity of the Clarke and Goldstein subdifferentials,

$$\partial_\delta F(x) = L\,\partial_\delta f(x),$$

and therefore

$$\operatorname{dist}(0, \partial_\delta F(x)) = L\,\operatorname{dist}(0, \partial_\delta f(x)).$$

Thus, x is (δ,ε)-Goldstein stationary for F if and only if it is $(\delta,\varepsilon/L)$-Goldstein stationary for f, i.e., (δ,ε')-Goldstein stationary for f.

Consequently, any randomized first-order algorithm that finds a (δ,ε)-Goldstein stationary point of F with probability at least $2/3$ would also find a (δ,ε')-Goldstein stationary point of f with the same probability. By Theorem C.1, this requires

$$T = \Omega(1/\varepsilon'^2) = \Omega\left(\frac{L^2}{\varepsilon^2}\right).$$

This proves the claim. □

Remark C.2. *The regime in Corollary 3.1 is compatible with our exploding-Lipschitz setting. After rescaling the lower bound from [Kornowski and Shamir, 2022], the hard instance has dimension $\widetilde{\mathcal{O}}((L/\varepsilon)^6)$ and requires $\delta \le L/(12\varepsilon)$. In our deep normalized networks, L can be $\Omega(\exp(d))$, so these conditions are mild for fixed or polynomially scaled accuracy parameters. Thus the lower-bound regime overlaps with the regime arising in our construction, and substituting $L = \Omega(\exp(d))$ into $\Omega(L^2/\varepsilon^2)$ yields exponential dependence on the network dimension.*

Algorithm 1 Probabilistic Greedy Goldstein Descent (PGGD)

Require: $x_0 \in \mathbb{R}^d$, $\delta > 0$, $\tau > 0$, $\rho \in (0,1)$, $\gamma \in (0,1)$, $f_{\mathrm{lb}} \le \inf f$.
$\Delta \leftarrow f(x_0) - f_{\mathrm{lb}}$
$T_{\max} \leftarrow \lceil \Delta/(\tau\delta) \rceil$
$m \leftarrow \left\lceil \frac{1}{\rho} \log \frac{T_{\max}+1}{\gamma} \right\rceil$
$x \leftarrow x_0$
for $t = 0$ **to** $T_{\max}$ **do**
  Sample i.i.d. $v_1, \dots, v_m \sim \mathrm{Unif}(\mathbb{S}^{d-1})$
  Query $f(x - \delta v_i)$ for all i and let $i^\star \in \arg\min_i f(x - \delta v_i)$
  if $f(x - \delta v_{i^\star}) \le f(x) - \tau\delta$ **then**
    $x \leftarrow x - \delta v_{i^\star}$
  else
    return x
  end if
end for
return x

Why exploding L undermines "dimension-free" rates. The discussion in Section 2 shows that the Lipschitz constant cannot be treated as a harmless constant in deep normalized networks. Existing guarantees for finding (δ, ε)-Goldstein stationary points scale polynomially in the objective Lipschitz constant L, and Corollary 3.1 shows that some polynomial dependence on L is unavoidable in the worst case. But in our setting, L is induced by the network itself. By Theorem 2.6, it can grow exponentially with depth, and hence with the parameter dimension. Therefore, a guarantee of the form $\mathrm{poly}(L, 1/\delta, 1/\varepsilon)$ may be formally independent of dimension, but it can still translate into exponential dependence on the dimension for deep normalized networks.

C.3 Proofs and Details for Section 3.2

For fixed x and $\delta > 0$, recall

$$m_\delta(x; v) := \min_{g \in \partial_\delta f(x)} \langle g, v \rangle, \qquad v \in \mathbb{S}^{d-1}.$$

The following elementary identity is the dual form of Goldstein stationarity.

Lemma C.3. *For any $\varepsilon \ge 0$,*

$$\mathrm{dist}(0, \partial_\delta f(x)) \le \varepsilon \qquad \Longleftrightarrow \qquad \max_{v \in \mathbb{S}^{d-1}} m_\delta(x; v) \le \varepsilon.$$

Moreover, if $\mathrm{dist}(0, \partial_\delta f(x)) > 0$, then

$$\mathrm{dist}(0, \partial_\delta f(x)) = \max_{v \in \mathbb{S}^{d-1}} m_\delta(x; v).$$

Proof. Let $Q = \partial_\delta f(x)$ and let $g^\star$ be the projection of 0 onto Q. For any unit vector v,

$$m_\delta(x; v) \le \langle g^\star, v \rangle \le \|g^\star\|_2.$$

Thus $\max_v m_\delta(x; v) \le \mathrm{dist}(0, Q)$. If $g^\star \neq 0$, take $v^\star = g^\star / \|g^\star\|_2$. The projection optimality condition gives $\langle g - g^\star, g^\star \rangle \ge 0$ for all $g \in Q$, hence

$$\langle g, v^\star \rangle \ge \|g^\star\|_2 \qquad \text{for all } g \in Q.$$

Therefore $m_\delta(x; v^\star) = \|g^\star\|_2 = \mathrm{dist}(0, Q)$. If $g^\star = 0$, then $\mathrm{dist}(0, Q) = 0$, and the stated equivalence for $\varepsilon \ge 0$ follows immediately. □

Lemma C.4. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be Lipschitz. Fix $x \in \mathbb{R}^d$ and $\delta > 0$. There exists a full-measure set $\mathcal{V}_x \subset \mathbb{S}^{d-1}$ such that, for every $v \in \mathcal{V}_x$ and every $\tau \in \mathbb{R}$,*

$$m_\delta(x; v) \ge \tau \qquad \Longrightarrow \qquad f(x - \delta v) \le f(x) - \tau\delta.$$

Proof. Fix x. By Davis et al. [2022, Lemma 2.3], there is a set $\mathcal{V}_x \subset \mathbb{S}^{d-1}$ of full surface measure such that, for every $v \in \mathcal{V}_x$, the function $t \mapsto f(x - tv)$ is differentiable for a.e. $t \in [0, \delta]$ and satisfies

$$f(x) - f(x - \delta v) = \int_0^\delta \langle \nabla f(x - tv), v \rangle \, dt.$$

Now suppose $m_\delta(x; v) \geq \tau$. For a.e. $t \in [0, \delta]$, whenever f is differentiable at $x - tv$, we have

$$\nabla f(x - tv) \in \partial f(x - tv) \subseteq \partial_\delta f(x),$$

since $x - tv \in B_\delta(x)$ for all $t < \delta$ and the endpoint is irrelevant for the integral. Therefore,

$$\langle \nabla f(x - tv), v \rangle \geq \tau \qquad \text{for a.e. } t \in [0, \delta].$$

Integrating gives

$$f(x) - f(x - \delta v) \geq \int_0^\delta \tau \, dt = \tau\delta,$$

which is equivalent to the desired inequality. □

Theorem 3.3. *Algorithm 1 finds a (δ, τ, ρ)-PGS point with probability at least $1 - \gamma$ using $\mathcal{O}\Big(\frac{\Delta}{\tau\delta\rho} \log \frac{\Delta}{\gamma\tau\delta}\Big)$ queries. In particular, for $(\tau, \rho) = (\varepsilon, \varepsilon)$, this becomes $\mathcal{O}\Big(\frac{\Delta}{\delta\varepsilon^2} \log \frac{\Delta}{\gamma\delta\varepsilon}\Big)$, with no polynomial dependence on L.*

Proof. Each accepted step decreases f by at least $\tau\delta$. Since $f(x_0) - f_{\mathrm{lb}} = \Delta$, the algorithm can accept at most $T_{\max} \leq \Delta/(\tau\delta)$ steps, and hence performs at most $T_{\max} + 1$ stopping checks.

Call a point x bad if

$$\Pr_{V \sim \mathrm{Unif}(\mathbb{S}^{d-1})} \big[f(x - \delta V) \leq f(x) - \tau\delta \big] > \rho.$$

At any bad point, the probability that all m sampled directions fail to give sufficient decrease is at most

$$(1 - \rho)^m \leq e^{-\rho m} \leq \frac{\gamma}{T_{\max} + 1}.$$

A union bound over the stopping checks shows that, with probability at least $1 - \gamma$, the algorithm never stops at a bad point. Thus the returned point x_{out} satisfies

$$\Pr_V \big[f(x_{\mathrm{out}} - \delta V) \leq f(x_{\mathrm{out}}) - \tau\delta \big] \leq \rho.$$

By Lemma C.4, for almost every V,

$$m_\delta(x_{\mathrm{out}}; V) \geq \tau \quad \Longrightarrow \quad f(x_{\mathrm{out}} - \delta V) \leq f(x_{\mathrm{out}}) - \tau\delta.$$

Therefore,

$$\Pr_V \big[m_\delta(x_{\mathrm{out}}; V) \geq \tau \big] \leq \rho,$$

so x_{out} is (δ, τ, ρ)-PGS. The query bound follows from at most $T_{\max} + 1$ iterations and $m + 1$ function evaluations per iteration. □

D Proofs and Details for Section 4

This appendix makes the two informal robustness statements from Section 4 precise. We reuse the rank-separated witness pair constructed in the proof of Theorem 2.3.

Theorem 4.1. *Fix $k \geq 2$ and $\eta \in (0, 1)$. There exist constants $a_{k,\eta} > 0$, $\gamma_{k,\eta} > 0$, and $\mu_{k,\eta} > 0$ such that, for every depth $m \in \mathbb{N}$, one can choose orthogonal weights $W_1, \ldots, W_m \in \mathbb{O}(k)$ and inputs $X_{\mathrm{clean}}, X_{\mathrm{adv}} \in \mathbb{R}^{k \times k}$ with the following properties:*

$$\mathrm{rank}(X_{\mathrm{clean}}) = k - 1, \qquad \mathrm{rank}(X_{\mathrm{adv}}) = k, \qquad \|X_{\mathrm{adv}} - X_{\mathrm{clean}}\|_F \leq a_{k,\eta} e^{-\gamma_{k,\eta} m}.$$

Moreover, there is a linear readout $A_m \in \mathbb{R}^{k \times k}$ and an offset $b_m \in \mathbb{R}$ with $\|A_m\|_F = 1$ such that the scalar score

$$s_m(Z) := \langle A_m, F_m(Z) \rangle_F + b_m$$

satisfies

$$s_m(X_{\mathrm{clean}}) \geq \mu_{k,\eta}, \qquad s_m(X_{\mathrm{adv}}) \leq -\mu_{k,\eta}.$$

Consequently, for the binary classifier $Z \mapsto \mathrm{sign}(s_m(Z))$ with clean label $+1$ at X_{clean},

$$\inf\big\{ \|E\|_F : \; s_m(X_{\mathrm{clean}} + E) \leq 0 \big\} \leq a_{k,\eta} e^{-\gamma_{k,\eta} m}.$$

Proof. Use the construction in the proof of Theorem 2.3. With $r = r_{k,\eta} > 1$ from Corollary B.5, that proof constructs inputs Y_0 and X_0 such that

$$\operatorname{rank}(Y_0) = k-1, \qquad \operatorname{rank}(X_0) = k, \qquad \|X_0 - Y_0\|_F \le 2(1-\eta)^{\frac{k}{2(k-1)}} r^{-\frac{mk}{2(k-1)}}.$$

It also constructs orthogonal weights for which, writing $Y_m = F_m(Y_0)$ and $X_m = F_m(X_0)$,

$$\|X_m - Y_m\|_F^2 \ge \delta_{k,\eta}, \qquad \delta_{k,\eta} := c_k(1-\eta)^k,$$

where $c_k = ((k-1)/k)^{k-1}$ is the constant from Lemma B.7. Set

$$X_{\text{clean}} := Y_0, \qquad X_{\text{adv}} := X_0, \qquad \gamma_{k,\eta} := \frac{k}{2(k-1)} \log r, \qquad a_{k,\eta} := 2(1-\eta)^{\frac{k}{2(k-1)}}.$$

Then $\|X_{\text{adv}} - X_{\text{clean}}\|_F \le a_{k,\eta} e^{-\gamma_{k,\eta} m}$.

It remains to choose the readout. Let

$$\Delta_m := \|Y_m - X_m\|_F \ge \sqrt{\delta_{k,\eta}}, \qquad A_m := \frac{Y_m - X_m}{\Delta_m}, \qquad b_m := -\frac{1}{2}\langle A_m, Y_m + X_m\rangle_F.$$

Then $\|A_m\|_F = 1$, and direct calculation gives

$$s_m(X_{\text{clean}}) = \langle A_m, Y_m\rangle_F + b_m = \frac{\Delta_m}{2}, \qquad s_m(X_{\text{adv}}) = \langle A_m, X_m\rangle_F + b_m = -\frac{\Delta_m}{2}.$$

Thus both inequalities hold with

$$\mu_{k,\eta} := \frac{1}{2}\sqrt{\delta_{k,\eta}}.$$

Finally, the perturbation $E = X_{\text{adv}} - X_{\text{clean}}$ is feasible for the displayed infimum because $s_m(X_{\text{clean}} + E) = s_m(X_{\text{adv}}) \le 0$. This proves the adversarial-radius bound. □

Proposition 4.2. *In the construction of Theorem 4.1, the perturbation $X_{\text{adv}} - X_{\text{clean}}$ is rank-creating:*

$$\operatorname{rank}(X_{\text{clean}}) = k-1, \qquad \operatorname{rank}(X_{\text{adv}}) = k.$$

Furthermore, if $Z \in \mathbb{R}^{k\times k}$ is any rank-deficient input for which the forward pass is well-defined, then $F_m(Z)$ is rank-deficient and

$$\|F_m(X_{\text{adv}}) - F_m(Z)\|_F \ge \sqrt{\delta_{k,\eta}}, \qquad \delta_{k,\eta} = c_k(1-\eta)^k.$$

Thus the full-rank adversarial branch is separated by a constant from the entire rank-deficient output branch.

Proof. The rank identities for X_{clean} and X_{adv} were proved in Theorem 4.1. Now let Z be rank-deficient and suppose the forward pass is well-defined. Each layer maps

$$Z_{t-1} \longmapsto \mathrm{BN}(Z_{t-1}W_t) = Z_{t-1}W_t D(Z_{t-1}W_t)^{-1/2}.$$

Here W_t is invertible and, because the normalization is well-defined, $D(Z_{t-1}W_t)^{-1/2}$ is an invertible diagonal matrix. Hence the rank cannot increase through a layer, so $F_m(Z)$ is rank-deficient.

On the full-rank branch of the construction, the proof of Theorem 2.3 gives

$$\lambda_{\min}(F_m(X_{\text{adv}})^\top F_m(X_{\text{adv}})) \ge \delta_{k,\eta}.$$

Applying Lemma B.8 with $X = F_m(X_{\text{adv}})$ and $Y = F_m(Z)$ yields

$$\|F_m(X_{\text{adv}}) - F_m(Z)\|_F^2 \ge \lambda_{\min}(F_m(X_{\text{adv}})^\top F_m(X_{\text{adv}})) \ge \delta_{k,\eta},$$

which proves the claim. □

E Experiments

Low-rank structure in image datasets. To support the rank-separation mechanism studied in the main text, we examine the spectral structure of standard image datasets. As shown in Figures 3 and 5, images from MNIST, Fashion-MNIST, and CIFAR-10 exhibit strong approximate low-rank structure. Their spectra decay rapidly, and most of the spectral energy is concentrated in a small number of singular components. Thus, while the images are not necessarily exactly rank-deficient, they typically lie close to a low-rank set. This observation is consistent with classical low-rank models of image data [Savas and Eldén, 2007, Zhang et al., 2013, Gu et al., 2014, Xu et al., 2015, Liu et al., 2016] and motivates our focus on perturbations that move an input away from its low-rank structure.

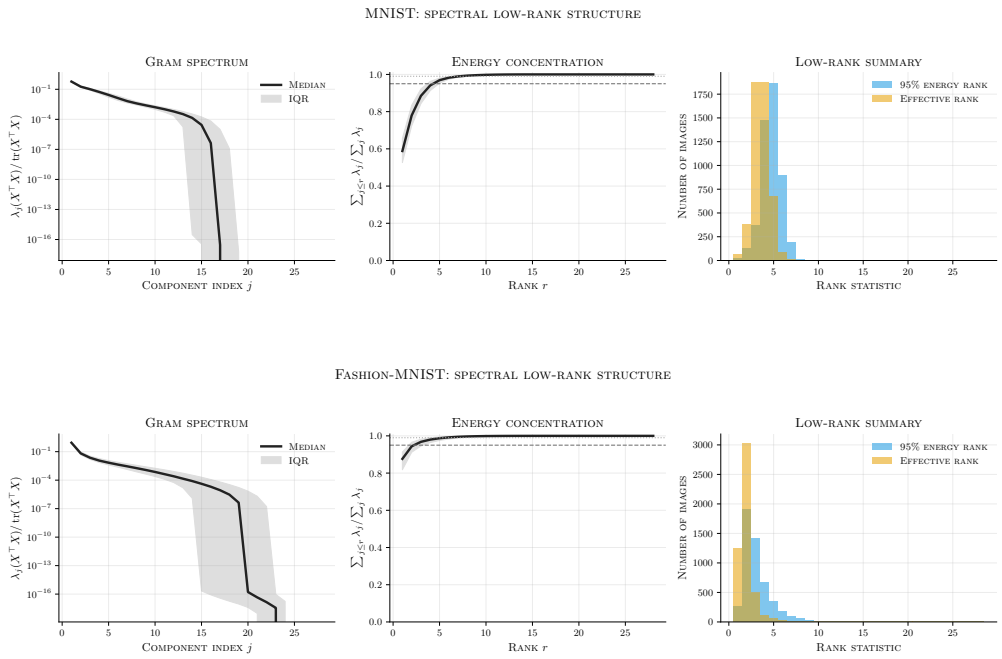


Figure 5: Spectral evidence for approximate low-rank structure in MNIST and Fashion-MNIST. For each dataset, the left panel shows the median normalized Gram spectrum $\lambda_j(X^\top X)/\operatorname{tr}(X^\top X)$ with interquartile range, the middle panel shows cumulative spectral energy with the 90% and 95% thresholds, and the right panel reports the distributions of effective rank and the rank needed to capture 95% of the energy. In both datasets, most of the spectral energy is concentrated in a few components, indicating that typical images lie close to a low-rank set.

Perturbation model for MNIST and Fashion-MNIST. In the MNIST and Fashion-MNIST experiments, we view each image as a matrix $X \in [0,1]^{28\times 28}$ and compare several perturbation families at matched pre-clipping energy. Given a perturbation Δ, the perturbed image is

$$\widetilde{X} = \Pi_{[0,1]}(X + \Delta)\,,$$

where $\Pi_{[0,1]}$ denotes entrywise clipping to the valid pixel range. The noise level is controlled by a variance parameter σ^2, and before clipping every sampled perturbation is rescaled so that

$$\|\Delta\|_F = \sigma\sqrt{28 \cdot 28}.$$

Thus all noise families have the same root-mean-square pixel energy before clipping, making the curves directly comparable across perturbation types. We consider four perturbation structures. IID pixel noise samples a full-dimensional Gaussian direction and is full rank with probability one, so it directly tests rank-creating perturbations. Singular-value (or same-subspace) noise perturbs only the singular values of the image and therefore stays within the row and column subspaces already present in X. Column-offset and row-offset noise broadcast one Gaussian vector across rows or columns, respectively, producing structured rank-one perturbations. Together, these perturbation families allow us to separate the effect of perturbation magnitude from the effect of perturbation geometry.

Rank effects of matched-energy perturbations. We next examine how different perturbation geometries affect the intrinsic rank structure of the images when their pre-clipping energy is matched. As shown in Figures 6 and 7, i.i.d. pixel noise consistently increases effective rank, stable rank, and the number of components needed to capture most of the spectral energy. In contrast, same-subspace perturbations have a much weaker effect, while row- and column-structured perturbations often preserve or even reduce these rank-based quantities. These observations support the mechanism in the main text that perturbations that create new spectral directions are more disruptive to the low-rank structure of images than perturbations that remain aligned with it.

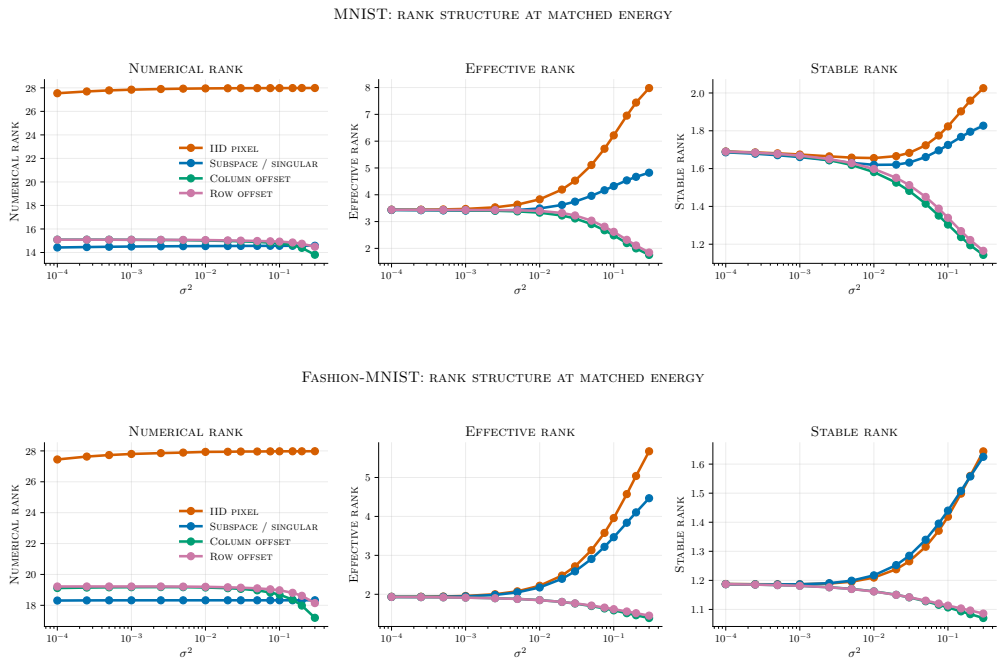


Figure 6: Rank structure under matched-energy perturbations on MNIST and Fashion-MNIST. For each dataset, we report numerical rank, effective rank, and stable rank as the perturbation variance σ^2 increases. IID pixel noise strongly inflates the intrinsic rank measures, while same-subspace, column-wise, and row-wise perturbations produce much smaller changes despite having the same pre-clipping energy.

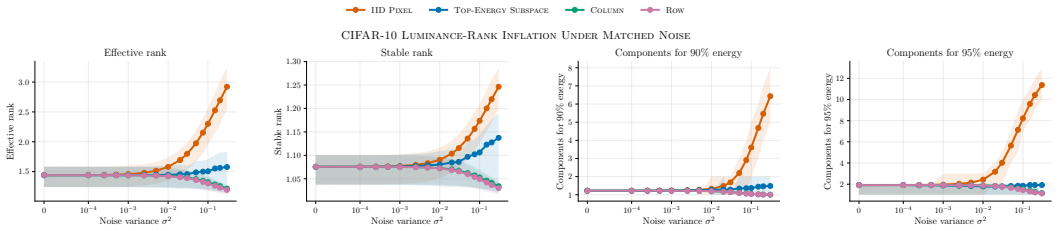


Figure 7: Rank inflation under matched-energy perturbations on CIFAR-10 luminance images. The plots show effective rank, stable rank, and the number of components required to capture 90% and 95% of the spectral energy. IID pixel noise substantially increases all rank-based measures as σ^2 grows, whereas same-subspace and structured row or column perturbations have a much milder effect.

Robustness under matched-energy perturbations. We next evaluate whether the geometric differences between perturbation families translate into different prediction behavior. Figure 8 shows that, even when perturbations are matched in pre-clipping energy, their effect on BatchNorm networks can differ substantially. Across MNIST, Fashion-MNIST, and CIFAR-10, subspace-preserving perturbations tend to preserve accuracy, confidence, and margin for larger noise levels, and they produce lower prediction-flip rates. In contrast, i.i.d. pixel noise and structured row or column perturbations often cause earlier confidence collapse, margin degradation, and prediction instability. This is consistent with the rank-separation mechanism from the main text: perturbations that move images away from their dominant low-rank structure are amplified more strongly than perturbations that remain aligned with that structure.

Batch normalization versus matched controls. We next compare models with batch normalization to matched architectures without batch normalization under the same perturbation families. As shown in Figure 9, BatchNorm models often maintain high accuracy at very small noise levels but can

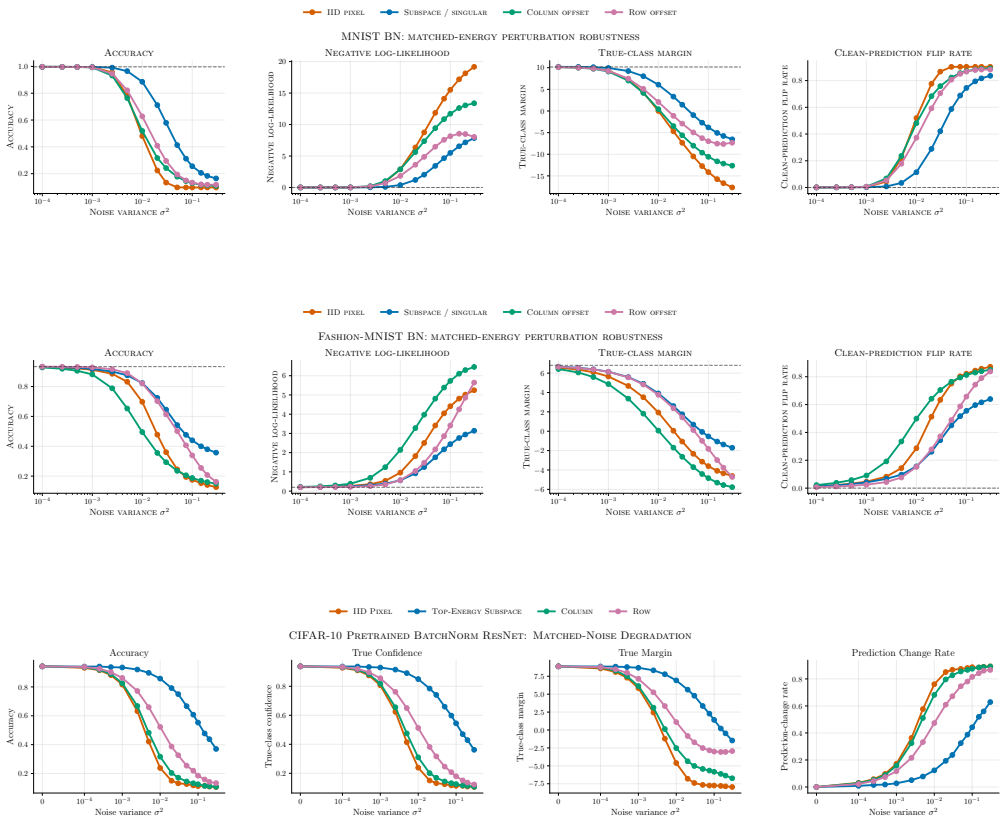


Figure 8: Matched-energy perturbation robustness across MNIST, Fashion-MNIST, and CIFAR-10 BatchNorm models. Each row shows performance as the noise variance σ^2 increases under i.i.d. pixel noise, subspace-preserving perturbations, column offsets, and row offsets. Subspace-preserving perturbations typically degrade accuracy, confidence, and margin more slowly and induce fewer prediction changes, while perturbations that move away from the image subspace cause earlier instability.

undergo a much sharper accuracy drop once the perturbation variance crosses a moderate threshold. This effect is especially clear on MNIST under i.i.d. pixel noise, column offsets, and row offsets. In contrast, the matched NoBN models tend to degrade more gradually. Same-subspace perturbations remain comparatively less damaging, which is consistent with the rank-separation mechanism discussed in the main text.

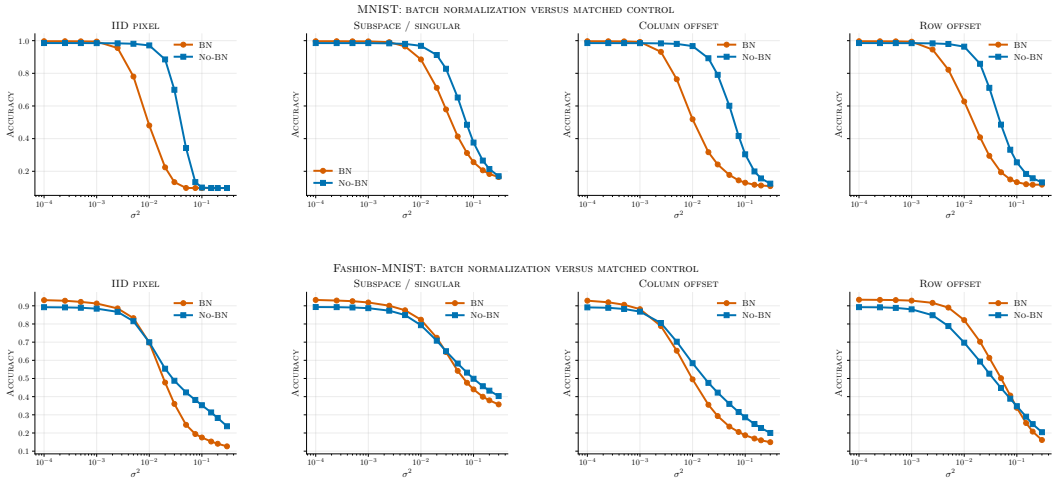


Figure 9: Batch normalization versus matched NoBN controls under structured input perturbations. Top row shows MNIST and bottom row shows Fashion-MNIST. For each dataset, we compare accuracy as the perturbation variance σ^2 increases under i.i.d. pixel noise, same-subspace singular perturbations, column offsets, and row offsets. BatchNorm models often remain accurate at low noise levels but exhibit sharper degradation at larger perturbation strengths, while matched NoBN controls typically degrade more gradually.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Complete proofs and experimental evidence are provided.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: This is *only a first step* toward understanding the unexpected implications of neural-network components, starting with batch normalization.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Complete proofs and clear assumptions are provided.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Details of the experiments are provided in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Code will be made available.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [N/A]

   Justification: The experiments are very small-scale, so this part does not apply.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [N/A]

   Justification: The experiments are very small-scale, so this part does not apply.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [N/A]

   Justification: The experiments are very small-scale, so this part does not apply.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: This is primarily a theoretical study.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: This is primarily a theoretical study.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [No]

    Justification: This is primarily a theoretical study.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No]

    Justification: This is primarily a theoretical study.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This is primarily a theoretical study.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This is primarily a theoretical study.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: [N/A]

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.