Adversarial Examples Are Not Bugs, They Are Superposition

Liv Gorton
Goodfire
San Francisco, CA
liv@goodfire.ai

Owen Lewis
Goodfire
San Francisco, CA
owen@goodfire.ai

Abstract

Adversarial examples—inputs with imperceptible perturbations that fool neural networks—remain one of deep learning's most perplexing phenomena despite nearly a decade of research. While numerous defenses and explanations have been proposed, there is no consensus on the fundamental mechanism. One underexplored hypothesis is that *superposition*, a concept from mechanistic interpretability, may be a major contributing factor, or even the primary cause. We present four lines of evidence in support of this hypothesis, greatly extending prior arguments by Elhage et al. [2022]: (1) superposition can theoretically explain a range of adversarial phenomena, (2) in toy models, intervening on superposition controls robustness, (3) in toy models, intervening on robustness (via adversarial training) controls superposition, and (4) in ResNet18, intervening on robustness (via adversarial training) controls superposition.

1 Introduction

Adversarial examples represent one of the most perplexing phenomena in deep learning: neural networks that achieve superhuman performance on many tasks can be fooled by perturbations so small they are imperceptible to humans. Despite nearly a decade of intensive research and many different hypotheses, there is no widely accepted explanation. In this paper, we explore an alternative hypothesis: superposition.

Superposition is a concept from the mechanistic interpretability literature. At a high level, superposition exploits the geometry of high-dimensional spaces to allow neural networks to represent more features than they have neurons. However, this strategy comes at a cost. Features in superposition necessarily interfere. On distribution, this interference is small, but in worst-case scenarios, it can be significant. One of the foundational papers on superposition hypothesized this interference could be linked to adversarial examples [Elhage et al., 2022], yet this hypothesis remains unexplored.

Our primary contribution is three experiments testing the relationship between superposition and robustness, in both toy models an ResNet18. These experiments are summarized in Figure 1. For toy models, we demonstrate both that superposition can control robustness, and that robustness can control superposition. For ResNet18, we show only that robustness can control superposition. (Unfortunately, without a method for controlling superposition in real models, we are unable to demonstrate the other direction in real models.)

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

	Superposition → Robustness	Robustness → Superposition
Toy Models	Feature Sparsity (Intervene) → Superposition (Measure) → Robustness (Measure)	Robustness (Intervene via adversarial training) → Superposition (Measure)
ResNet18	*(No methodology available)*	Robustness (Intervene via adversarial training) → Superposition (*(Unable to directly measure)*) → SAE Loss (Measure)

Figure 1: **Overview of experiments.** The three primary experiments test the relationship between superposition and robustness in different ways.

Combined, these results strongly imply that superposition is at least one causal factor in the existence of adversarial examples. They don't necessarily suggest that it's the only factor, as we can't intervene on superposition in real models to isolate this.

At the same time, we take seriously the possibility that it might be the primary explanation. Although it isn't the primary focus of this paper, it seems to us that superposition is sufficient to theoretically explain all the adversarial phenomena we're aware of. This is summarized in Table 1.

Table 1: Six adversarial example phenomena and potential explanations.

Phenomenon	**Superposition Explanation**
Existence: Adversarial examples exist across essentially all neural networks [Szegedy et al., 2014, Goodfellow et al., 2015]	Features can be attacked by perturbing all the features in superposition with them. An attacker can do this iteratively at each layer.
Noise-like structure: Adversarial perturbations appear as unstructured high-frequency noise rather than semantic patterns [Goodfellow et al., 2015, Sharma et al., 2019]	Adversarial attacks work by attacking many features, which are totally unrelated except for the fact that they're in superposition with the actual targets.
Attack Transferability: Adversarial examples transfer between independently trained models [Goodfellow et al., 2015, Liu et al., 2017]	If the same features are in superposition with each other, attacks based on superposition will transfer. Features which are anti-correlated are preferentially put in superposition with each other [Elhage et al., 2022] and therefore attacks should transfer.
Training difficulty: Adversarial training is fundamentally difficult, requiring significant computational resources and degrading natural accuracy [Madry et al., 2019]	Superposition increases the capacity of models. If improving model robustness requires reducing superposition, that fundamentally reduces model capacity.
Interpretability: Adversarially trained models become markedly more interpretable with neurons that correspond to human-understandable concepts [Engstrom et al., 2019]	In the absence of superposition, neurons can be monosemantic, and also less noisy.
Training on Attacks Transfers Clean Performance: Training on *mislabeled* data with adversarial attack towards the erroneous label induces correct behavior on clean data [Ilyas et al., 2019]	Training on adversarial attacks transfers to clean data because adversarial attacks encode interfering combinations of genuinely useful circuits.

2 Background

The mechanistic interpretability literature often assumes that model representations are linear. That is, the hidden activations h of some layer can be understood as

$$h = \sum_{i<k} a_i \vec{f}_i + \vec{b}$$

where k is the total number of features, a_i is the activation of a feature i, and f_i is a direction in activation space representing that feature. Roughly, activation represents the intensity or strength of a feature in response to a particular input.[1]

[1] Typically, features are imagined to be one-dimensional, but this can be generalized to allow more dimensions.

One might expect that if a neural network representation has n dimensions, it can only represent $k \leq n$ linear features. However, results from an area of mathematics called compressed sensing suggest that neural networks could represent many more features ($k >> n$), so long as features are sparse (that is, zero on most examples). This is called the superposition hypothesis.

Superposition necessarily entails *interference*. When $k > n$ features are represented in an n-dimensional space, the feature vectors $\{\vec{f_i}\}_{i=1}^{k}$ cannot all be mutually orthogonal. This non-orthogonality means that activating feature i with coefficient a_i produces (apparent) spurious activations in feature j proportional to $a_i\langle\vec{f_i}, \vec{f_j}\rangle$. Models can partially compensate for this interference by learning negative biases $b_j < 0$ that suppress small spurious activations below a threshold. However, this compensation mechanism assumes the total interference $\sum_{i\neq j} a_i\langle\vec{f_i}, \vec{f_j}\rangle$ remains bounded. In worst-case scenarios, an adversary can coordinate activations to make this sum arbitrarily large, overwhelming the bias term. (This aligns with compressed sensing theory, which only guarantees reconstruction with high probability under random, not adversarial, conditions.)

Elhage et al. [2022] demonstrated that this interference mechanism enables adversarial attacks in toy models. Specifically, consider a target feature $\vec{f}_{\text{target}}$ in superposition with features $\{\vec{f_1}, \ldots, \vec{f_m}\}$ where $\langle\vec{f}_{\text{target}}, \vec{f_i}\rangle = \epsilon_i \neq 0$. An adversary can exploit this by adding input perturbations that activate each interfering feature by a small amount δ_i. While each individual contribution $\delta_i\epsilon_i$ to the target feature's activation is negligible, the cumulative effect $\sum_{i=1}^{m} \delta_i\epsilon_i$ can be made arbitrarily large by choosing appropriate δ_i values (subject to the perturbation budget). This is precisely the interference that models attempt to suppress through learned biases under normal operating conditions.

This vulnerability compounds across layers. At each layer, the adversary can exploit superposition to create unwanted feature activations, which then propagate to the next layer as inputs. These corrupted activations at the next layer can then be constructed to do the same kind of attack, allowing errors to accumulate through the network.

3 Causal Evidence from Toy Models of Superposition

To test whether superposition causally contributes to adversarial vulnerability, we extend the toy models of Elhage et al. [2022], the standard theoretical model of superposition. In the toy model setup, it is possible to exactly measure superposition, which is not possible in real models because it requires knowledge of the ground truth features learned by the model. It also allows us to control superposition by manipulating feature sparsity. This will allow us to show both that superposition controls robustness and that robustness controls superposition in the toy models setup.

3.1 Setup

3.1.1 Toy Models

We consider a simplified[2] version of the basic setup of Elhage et al. [2022]. Our data consists of $n = 100$ features. They are linearly projected into a $m = 20$ hidden units, $h = Wx$, and then reconstructed by a ReLU layer, $x' = \text{ReLU}(W^T x + b)$. The loss is mean squared error.

The behavior of this toy model varies based on the feature sparsity, S. This is the probability that the input features are zero. When features are sparse, this setup exhibits superposition, representing more features than there are hidden dimensions. The amount of superposition increases with sparsity.

3.1.2 Measuring Superposition

One reason for our interest in the toy model setting is that superposition can be exactly measured. One way to do this is by looking at the features per dimension [Elhage et al., 2022], i.e., how many features the model is attempting to represent per feature dimension:

$$\frac{||W||_F^2}{n} \tag{1}$$

[2] We consider only uniform feature importance, causing the loss to simplify into mean squared error.

This works because features are roughly represented with unit norm when learned. When the features per dimension > 1, the model must be using superposition, as it represents more features than it has dimensions.

3.1.3 Measuring Robustness

We also need to know how vulnerable our models are to adversarial examples. To measure adversarial vulnerability, we generate L_2-bounded adversarial examples. For each input x, we find the worst-case perturbation within an ϵ-ball that maximizes reconstruction error:

$$x_{adv} = x + \epsilon \cdot \arg\max_{\|\delta\|_2 \leq 1} \mathcal{L}(x + \epsilon\delta) \tag{2}$$

We set ϵ to 10% of the average input norm.

We reproduce the approach of Elhage et al. [2022], who exploit the toy model setup to analytically construct attacks that optimally attack each specific output feature, and then take the worst such attack. They take this approach to avoid gradient masking issues from ReLU. However, while this would be an optimal attack in terms of L_∞ in the output space, it has the potential to be quite suboptimal for affecting the output as measured by L_2/MSE. For this reason, we primarily consider a more traditional adversarial attack. We add a small amount of noise to avoid gradient issues, and then do a one-step gradient L2 attack. All results in the main paper are based on this attack.

To compare the vulnerability of models, we consider how many times more vulnerable it is than a model without superposition (i.e., our model with the highest input feature density, with every feature present in all training inputs).

3.1.4 Adversarial Training Protocol

Since we want to test whether causality flows from adversarial robustness to superposition, we also need to be able to produce adversarially robust versions of our toy models. To do this, we train new toy models over the same range of feature densities, but using a mixture of clean and adversarial training examples:

$$\mathcal{L}_{adv} = \alpha \cdot \mathcal{L}(x) + (1 - \alpha) \cdot \mathcal{L}(x_{adv}) \tag{3}$$

where $\alpha = 0.5$ balances clean and *robust* accuracy. We use L_2 attack with $\epsilon = 0.1\|x\|_2$. We can generate these attacks on-the-fly using either approach from the previous section, but unless otherwise specified, we use the more standard gradient attack rather than the Elhage method. We train a model with the same configuration as the model used in Section 3.1.1 for 150,000 steps with a learning rate 10^{-3}. (This follows a common practice in adversarial training where models are trained for extended periods compared to standard training due to the unique optimization dynamics; see e.g., Rice et al. [2020] for discussion of adversarial training dynamics.)

3.2 Intervening on Superposition Controls Adversarial Vulnerability

We use feature sparsity to manipulate the level of superposition, and observe resulting changes in adversarial robustness. In particular, we vary the feature density $(1 - \text{sparsity})$ exponentially from 1.0 to 0.1, training 30 models simultaneously with different sparsity levels, and observe the resulting adversarial robustness. This is the general setup of [Elhage et al., 2022], but we focus on more powerful noise-plus-gradient adversarial attacks. (A reproduction of the original Elhage experiment can be found in the appendix, see figure 7.)

Our first goal is to confirm that intervening on feature sparsity has the expected effect on superposition, in order to validate it as a way to manipulate superposition in our larger experiment. Panel A of figure 2 shows the expected results, including a temporary plateau corresponding to antipodal superposition.

Having validated our instrumental variable, we now proceed to the core result. Panel B of figure 2 shows that adversarial vulnerability increases with both feature sparsity and superposition (quantified as features per dimension). There is one striking dip corresponding to antipodal superposition.

The mechanism is intuitive: when features are in superposition, they share directions in activation space. An adversary can exploit this by perturbing all interfering features simultaneously. Since features in superposition are not orthogonal, small perturbations to many features accumulate into large changes in the target feature's reconstruction.

It is worth noting that there is some subtlety to comparing adversarial robustness across different feature densities, since the distribution we are evaluating on changes. However, this should, if anything, bias in the opposite direction of the trend we're observing. Having fewer features active should tend to make models more robust, since fewer ReLUs would be open, allowing gradients through. Thus, we believe this concern would cause us to *underestimate* the relationship between superposition and adversarial vulnerability. However, we do get some cross-validation from the robust models in the next section, since these shift superposition independently of the data distribution, and we still see the same trend.

3.3 Intervening on Adversarial Robustness Controls Superposition

To establish bidirectional causality, we next ask: does improving adversarial robustness reduce superposition? We perform adversarial training on our toy models and measure the resulting changes in superposition.

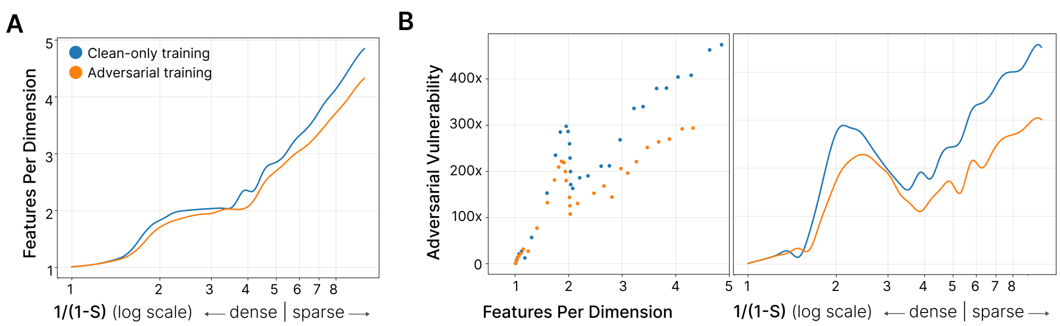

Figure 2: **Adversarial training reduces superposition.** Comparison of models before (blue) and after (orange) adversarial training. **A:** Features per dimension decreases for a given sparsity level. **B:** (Left) Models become more vulnerable to adversarial examples as superposition increases. (Right) Models become more vulnerable to adversarial examples as feature sparsity increases (with a drop for antipodal superposition).

Figure 2 demonstrates that adversarial training reduces superposition. Models that underwent adversarial training decreased their adversarial vulnerability and decreased features per dimension for some original input sparsity. However, we note two surprising phenomena. Firstly, as discussed earlier, we note a drop in vulnerability to adversarial examples when models switch to antipodal superposition. Secondly, we note that robust models are often more robust than expected for their superposition level. Our interpretation is that the overall level of superposition doesn't tell the full story; we conjecture that some superposition structures (that is, the matrix of interference between features) are more or less vulnerable to superposition. See Discussion (section 5).

In contrast to the previous section, where we reproduced and extended the results of Elhage et al. [2022], to the best of our knowledge, these results are the first to demonstrate causality from robustness to superposition.

3.3.1 Theoretical Intuition

While not a formal derivation, we find it useful to conceptualize the difference between standard and adversarial training through the lens of interference minimization.[3]

Given dataset $\mathcal{D}$, neural network parameters θ, and a measure of interference I, we might conceptualize neural network training as:

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}}[I(x,y;\theta)]$$

That is, the goal is to minimize the *average* expected interference. Whereas during adversarial training, it might be better instead to conceptualize the objective with respect to interference as:

$$\min_{\theta} \max_{\mathcal{D}\in\mathcal{D}_{\mathrm{OOD}}} \mathbb{E}_{(x,y)\sim\mathcal{D}}[I(x,y;\theta)]$$

[3]This is a conceptual framework for building intuition rather than a formal theoretical result. The actual optimization dynamics are considerably more complex.

That is, the goal is to minimize the *maximum* expected interference over out-of-distribution data.

This conceptualization suggests that adversarial training forces the model to consider worst-case interference patterns rather than average-case, potentially explaining why it reduces superposition in our experiments.

3.3.2 Adversarial Examples Exploit Feature Interference

We constructed superposition geometry graphs similarly to Elhage et al. [2022], where each feature has a node, and edge (i, j) represents $(W_i \cdot W_j)^2$.

These graphs can then be used to understand how this geometry is being exploited in adversarial attacks, and subsequently why a model is adversarially robust.

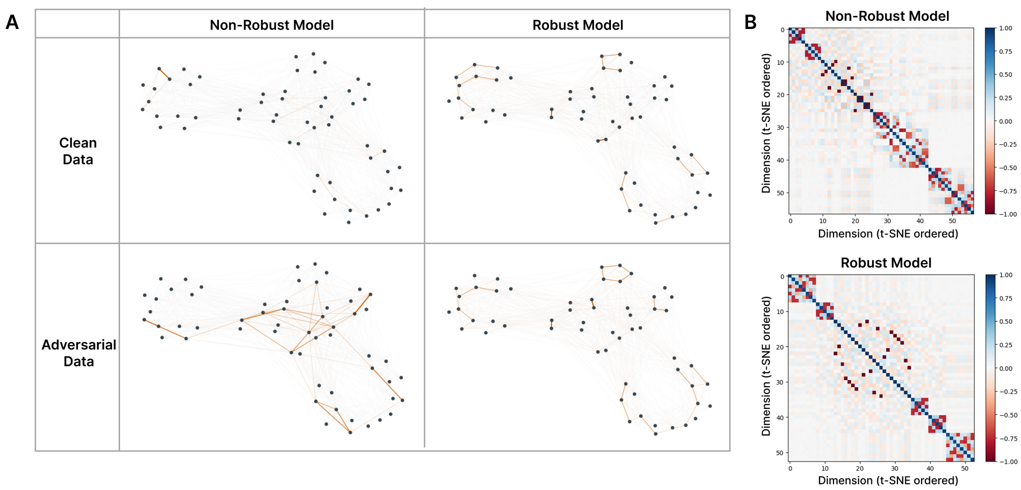

Figure 3: **Adversarial attacks activate interfering features in superposition.** (A) We consider two models, one robust and one non-robust, as well as clean and adversarial data. We visualize the superstructure of each toy model as a graph. Edge thickness is dependent on $(W_i \cdot W_j)^2$. We then highlight the superposition affecting that input in orange. (B) We plot heatmaps of the interference ($W^\top W$) for the robust and non-robust models used in (A). Non-robust models have a mean off-diagonal interference $2\times$ that of robust models.

Figure 3 illustrates how adversarial attacks exploit feature interference patterns. In non-robust models (left column), clean inputs activate relatively few features with minimal interference between them, as shown by the sparse orange highlighting in the superposition graph. Adversarial inputs, however, activate many interfering features simultaneously, precisely the pattern expected if attacks exploit superposition geometry. In contrast, robust models (right column) show similar sparse activation patterns for both clean and adversarial inputs, suggesting that adversarial training has reorganized the feature geometry to prevent interference-based attacks. The heatmaps in panel (B) confirm this: non-robust models exhibit mean off-diagonal interference approximately 2× that of robust models, indicating denser superposition structure.

3.4 Superposition Geometry

We can also use the graph visualization technique to compare a larger set of models. In figure 4, we look at pairs of non-robust and robust models trained at the same sparsity level. The robust models have less superposition (corresponding to a further left position) but strikingly similar superposition geometries.

4 Evidence From Real Models

We now turn our attention to real models. Unfortunately, since we have no way to intervene on superposition in real models, we can't test the causal effect of superposition on robustness. However,

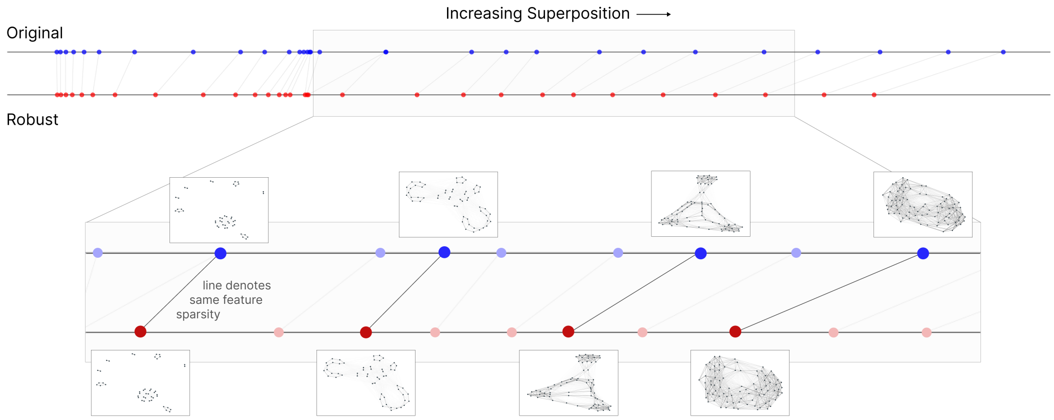

Figure 4: **Adversarial training reduces superposition while preserving geometric structure.** We plot all our robust and non-robust models as "points on a superposition number line". A line connects models trained on the same level of sparsity. We can see that robust models have lower superposition. For selected models, we visualize the superposition structure as a graph.

we can still adversarially train models to control robustness and observe the effect on superposition via the proxy of sparse autoencoder loss (discussed further in Section 4.2)

4.1 Methods

4.1.1 Adversarially Robust Models

To study adversarial robustness in real models, we used robust ResNet18s trained on ImageNet [Russakovsky et al., 2015] from Salman et al. [2020].[4] These robust models are trained against different attack sizes, varying their robustness.

4.1.2 Sparse Autoencoders

We train sparse autoencoders (SAEs) on the outputs of ResNet18's four residual stages (conv2_x through conv5_x), which produce 256-, 512-, 1024-, and 2048-dimensional feature maps at progressively lower spatial resolutions. We trained both L1 ReLU SAEs [Conerly et al., 2024] and TopK SAEs [Gao et al., 2024] on standardized activations to mitigate the effect on training of activation statistics. Additional training details can be found in appendix B.

4.2 Robust Models Achieve Better SAE Reconstruction

There is no direct way to measure the amount of superposition in real models, and so instead we must consider proxies of superposition.

SAEs are designed to model superposition and will naturally have a higher loss when there is more superposition. There are several reasons for this: (1) if a model of a fixed size has more superposition, it has more total features that an SAE has to model, (2) with more total features, there will also be more active features on any example, (3) in denser superposition, the SAE will be forced to either sometimes model a strongly activating feature as activating other features, or sometimes not represent small activations.

Figure 5 shows that for a given sparsity level, more robust models consistently achieve better reconstruction loss. Does this imply robustness effects superposition? The only way we see to avoid this is if some other change to the model could lower SAE loss independent of superposition, and we don't have any hypotheses for what that could be.[5]

[4] `https://huggingface.co/madrylab/robust-imagenet-models`

[5] From a Popperian perspective, the hypothesis that robustness influences superposition should gain credit for predicting a surprising phenomenon, even if some alternative explanation can retrospectively be proposed.

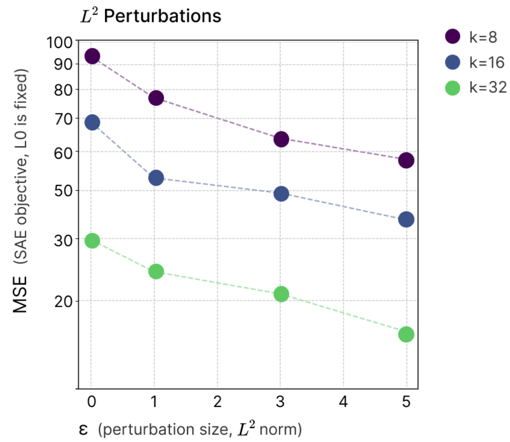

Figure 5: **Robust models achieve better reconstruction at a given sparsity level.** TopK SAEs with different sparsity levels ($k = \{8, 16, 32\}$) were trained on ResNet18 models with varying L2 robustness ($\epsilon \in \{0, 1, 3, 5\}$). Lower MSE at fixed sparsity likely indicates less interference and therefore less superposition.

213 ### 4.3 Adversarial Examples Increase L0

Our sparse autoencoders provide the opportunity for an additional experiment. If adversarial attacks do exploit interference, we'd expect them to activate more features. Each feature can both be attacked via interference and used to attack later features.

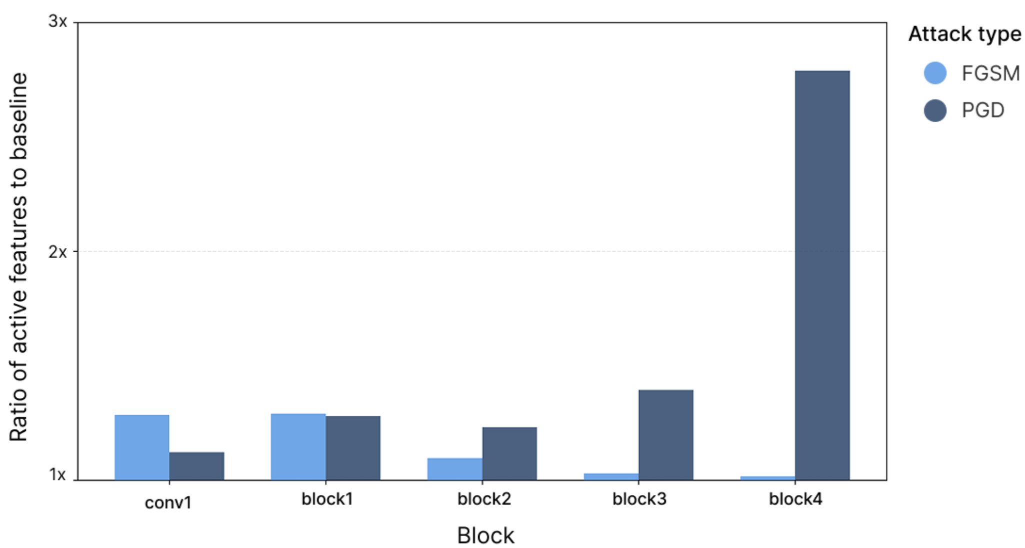

Figure 6: **Adversarial examples activate more features than clean inputs.** Ratio of L0 (active features) for FGSM [Goodfellow et al., 2015] and PGD [Madry et al., 2019] attacks versus clean data across ResNet18 layers. PGD shows a dramatic increase at layer 4 (2.8×), suggesting adversarial attacks can increasingly exploit feature interference deeper in the network. See Table 2 for detailed statistics.

In figure 6, we observe that adversarial examples consistently activate more features than clean inputs across all layers. For FGSM attacks, we see modest increases of 1.3× at conv1, maintaining similar levels through layers 1-3, before dropping to near baseline (1.02×) at layer 4. PGD attacks show a different pattern: starting with 1.1-1.3× increases in early layers (conv1 and layer1), maintaining moderate increases through layers 2-3 (1.2-1.4×), then dramatically spiking to 2.8× at layer 4. This striking divergence between attack types at the final layer suggests that iterative attacks (PGD) can more effectively exploit accumulated interference in deeper representations. This aligns with the well-established finding that PGD, as a stronger multi-step optimization-based attack, typically achieves higher success rates than single-step methods like FGSM [Madry et al., 2019].

5 Discussion

We have argued that adversarial examples are caused, at least in part, by superposition. Beyond the theoretical arguments, three lines of empirical evidence support this hypothesis: (1) in toy models, superposition controls robustness, (2) in toy models, robustness controls superposition, and (3) in real models, robustness controls superposition.

While these arguments appear compelling, several limitations warrant consideration. First, our analysis relies substantially on proxy variables to control and measure effects, particularly in real models. These proxies may fail to capture the full complexity of the phenomena. Second, our experimental results could be consistent with adversarial examples having multiple causal factors beyond superposition, especially in real models. Without methods to directly manipulate superposition in real models and observe resulting changes in robustness, we cannot quantify the relative magnitude of superposition's contribution, only establish the potentiality of a causal relationship. Despite these limitations, the evidence strongly suggests that superposition constitutes a major factor in adversarial robustness. Further confidence in this hypothesis will require developing more sophisticated tools for measuring and manipulating superposition in real models.

Several unexpected findings merit further investigation: (1) the temporary improvement in robustness observed near antipodal superposition configurations, and (2) the observation that models with equivalent overall superposition levels but different superposition structures exhibit varying robustness to L2 adversarial attacks. These phenomena warrant deeper theoretical and empirical examination.

If superposition represents a primary cause of adversarial examples, this implies a fundamental and unavoidable trade-off. Superposition enables models to effectively simulate substantially larger sparse models; achieving robustness would necessitate sacrificing this computational advantage. Conversely, this relationship would indicate a profound alignment between the objectives of interpretability and robustness research.

6 Related Work

Adversarial Examples. Since their discovery [Szegedy et al., 2014, Goodfellow et al., 2015], numerous attacks emerged [Moosavi-Dezfooli et al., 2015, Carlini and Wagner, 2016, Madry et al., 2019, Croce and Hein, 2020], extending to physical [Kurakin et al., 2016] and universal perturbations [Moosavi-Dezfooli et al., 2017].

Theoretical Explanations. Beyond the linear hypothesis [Goodfellow et al., 2015], explanations include geometric perspectives [Gilmer et al., 2018, Khoury and Hadfield-Menell, 2019, Shafahi et al., 2020, Shamir et al., 2022], concentration of measure [Mahloujifar et al., 2018, 2019], high-dimensional inevitability [Tanner et al., 2024], and manifold analyses [Xiao et al., 2022]. The "robust features" hypothesis [Ilyas et al., 2019] suggests models exploit non-robust but predictive patterns.

Defenses. Adversarial training remains dominant [Madry et al., 2019, Zhang et al., 2019, Shafahi et al., 2019], while certified approaches use verification [Zhang et al., 2018, Gowal et al., 2019, Wang et al., 2021] or randomized smoothing [Cohen et al., 2019, Lecuyer et al., 2019].

Robustness-Accuracy Tradeoff. Fundamental tension exists between standard and robust accuracy [Tsipras et al., 2019, Zhang et al., 2019, Javanmard et al., 2020, Rice et al., 2020, Schmidt et al., 2018], with mitigations via unlabeled data [Carmon et al., 2022, Raghunathan et al., 2020].

Interpretability. Robust models exhibit aligned gradients and interpretable features [Engstrom et al., 2019, Tsipras et al., 2019, Ganz et al., 2023, Srinivas et al., 2024]; disentangled representations improve robustness [Yang et al., 2021, Guesmi et al., 2024].

Transferability and Compression. Examples transfer due to shared representations [Demontis et al., 2019, Wu et al., 2018]; compression-robustness connections reveal capacity constraints [Ye et al., 2021, Gui et al., 2019, Xie et al., 2019, Yi et al., 2020].

Superposition and Mechanistic Interpretability. Superposition allows exponentially many features in high-dimensional spaces [Elhage et al., 2022]. SAEs decompose superposed features [Cunningham et al., 2023, Bricken et al., 2023, Templeton et al., 2024, Gao et al., 2024], though computational bounds exist [Adler and Shavit, 2025].

Acknowledgments

We would like to thank Chris Olah for helpful discussions that contributed to the development of this work. We are also grateful to Michael Byun, Tom McGrath, and Michael Pearce for their feedback on drafts of this manuscript.

References

Micah Adler and Nir Shavit. On the complexity of neural computation in superposition, 2025. URL https://arxiv.org/abs/2409.15318.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. *CoRR*, abs/1608.04644, 2016. URL http://arxiv.org/abs/1608.04644.

Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, Percy Liang, and John C. Duchi. Unlabeled data improves adversarial robustness, 2022. URL https://arxiv.org/abs/1905.13736.

Jeremy M Cohen, Elan Rosenfeld, and J. Zico Kolter. Certified adversarial robustness via randomized smoothing, 2019. URL https://arxiv.org/abs/1902.02918.

Tom Conerly, Adly Templeton, Trenton Bricken, Jonathon Marcus, and Tom Henighan. Update on how we train saes, 2024. https://transformer-circuits.pub/2024/april-update/index.html.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. *CoRR*, abs/2003.01690, 2020. URL https://arxiv.org/abs/2003.01690.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models, 2023. URL https://arxiv.org/abs/2309.08600.

Ambra Demontis, Marco Melis, Maura Pintor, Matthew Jagielski, Battista Biggio, Alina Oprea, Cristina Nita-Rotaru, and Fabio Roli. Why do adversarial attacks transfer? explaining transferability of evasion and poisoning attacks. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 321–338, Santa Clara, CA, August 2019. USENIX Association. ISBN 978-1-939133-06-9. URL https://www.usenix.org/conference/usenixsecurity19/presentation/demontis.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition. *Transformer Circuits Thread*, 2022. URL https://transformer-circuits.pub/2022/toy_model/index.html.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Brandon Tran, and Aleksander Madry. Adversarial robustness as a prior for learned representations, 2019. URL https://arxiv.org/abs/1906.00945.

Roy Ganz, Bahjat Kawar, and Michael Elad. Do perceptually aligned gradients imply adversarial robustness?, 2023. URL https://arxiv.org/abs/2207.11378.

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders, 2024. URL https://arxiv.org/abs/2406.04093.

Justin Gilmer, Luke Metz, Fartash Faghri, Samuel S. Schoenholz, Maithra Raghu, Martin Wattenberg, and Ian Goodfellow. Adversarial spheres, 2018. URL https://arxiv.org/abs/1801.02774.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2015. URL https://arxiv.org/abs/1412.6572.

Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Mann, and Pushmeet Kohli. On the effectiveness of interval bound propagation for training verifiably robust models, 2019. URL https://arxiv.org/abs/1810.12715.

Amira Guesmi, Nishant Suresh Aswani, and Muhammad Shafique. Exploring the interplay of interpretability and robustness in deep neural networks: A saliency-guided approach, 2024. URL https://arxiv.org/abs/2405.06278.

Shupeng Gui, Haotao Wang, Chen Yu, Haichuan Yang, Zhangyang Wang, and Ji Liu. Model compression with adversarial robustness: A unified optimization framework, 2019. URL https://arxiv.org/abs/1902.03538.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features, 2019. URL https://arxiv.org/abs/1905.02175.

Adel Javanmard, Mahdi Soltanolkotabi, and Hamed Hassani. Precise tradeoffs in adversarial training for linear regression, 2020. URL https://arxiv.org/abs/2002.10477.

Marc Khoury and Dylan Hadfield-Menell. On the geometry of adversarial examples, 2019. URL https://openreview.net/forum?id=H1lug3R5FX.

Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *CoRR*, abs/1607.02533, 2016. URL http://arxiv.org/abs/1607.02533.

Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy, 2019. URL https://arxiv.org/abs/1802.03471.

Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks, 2017. URL https://arxiv.org/abs/1611.02770.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks, 2019. URL https://arxiv.org/abs/1706.06083.

Saeed Mahloujifar, Dimitrios I. Diochnos, and Mohammad Mahmoody. The curse of concentration in robust learning: Evasion and poisoning attacks from concentration of measure, 2018. URL https://arxiv.org/abs/1809.03063.

Saeed Mahloujifar, Xiao Zhang, Mohammad Mahmoody, and David Evans. Empirically measuring concentration: Fundamental limits on intrinsic robustness, 2019. URL https://arxiv.org/abs/1905.12202.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. *CoRR*, abs/1511.04599, 2015. URL http://arxiv.org/abs/1511.04599.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations, 2017. URL https://arxiv.org/abs/1610.08401.

Aditi Raghunathan, Sang Michael Xie, Fanny Yang, John Duchi, and Percy Liang. Understanding and mitigating the tradeoff between robustness and accuracy, 2020. URL https://arxiv.org/abs/2002.10716.

Leslie Rice, Eric Wong, and J. Zico Kolter. Overfitting in adversarially robust deep learning, 2020. URL https://arxiv.org/abs/2002.11569.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115 (3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Madry. Do adversarially robust imagenet models transfer better?, 2020. URL https://arxiv.org/abs/2007.08489.

Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Mądry. Adversarially robust generalization requires more data, 2018. URL https://arxiv.org/abs/1804.11285.

Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S. Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free!, 2019. URL https://arxiv.org/abs/1904.12843.

Ali Shafahi, W. Ronny Huang, Christoph Studer, Soheil Feizi, and Tom Goldstein. Are adversarial examples inevitable?, 2020. URL https://arxiv.org/abs/1809.02104.

Adi Shamir, Odelia Melamed, and Oriel BenShmuel. The dimpled manifold model of adversarial examples in machine learning, 2022. URL https://arxiv.org/abs/2106.10151.

Yash Sharma, Gavin Weiguang Ding, and Marcus Brubaker. On the effectiveness of low frequency perturbations, 2019. URL https://arxiv.org/abs/1903.00073.

Suraj Srinivas, Sebastian Bordt, and Hima Lakkaraju. Which models have perceptually-aligned gradients? an explanation via off-manifold robustness, 2024. URL https://arxiv.org/abs/2305.19101.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks, 2014. URL https://arxiv.org/abs/1312.6199.

Kasimir Tanner, Matteo Vilucchio, Bruno Loureiro, and Florent Krzakala. A high dimensional statistical model for adversarial training: Geometry and trade-offs, 2024. URL https://arxiv.org/abs/2402.05674.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*, 2024. URL https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy, 2019. URL https://arxiv.org/abs/1805.12152.

Shiqi Wang, Huan Zhang, Kaidi Xu, Xue Lin, Suman Jana, Cho-Jui Hsieh, and J. Zico Kolter. Beta-crown: Efficient bound propagation with per-neuron split constraints for complete and incomplete neural network robustness verification, 2021. URL https://arxiv.org/abs/2103.06624.

Lei Wu, Zhanxing Zhu, Cheng Tai, and Weinan E. Understanding and enhancing the transferability of adversarial examples, 2018. URL https://arxiv.org/abs/1802.09707.

Jiancong Xiao, Liusha Yang, Yanbo Fan, Jue Wang, and Zhi-Quan Luo. Understanding adversarial robustness against on-manifold adversarial examples, 2022. URL `https://arxiv.org/abs/2210.00430`.

Hui Xie, Jirong Yi, Weiyu Xu, and Raghu Mudumbai. An information-theoretic explanation for the adversarial fragility of ai classifiers, 2019. URL `https://arxiv.org/abs/1901.09413`.

Shuo Yang, Tianyu Guo, Yunhe Wang, and Chang Xu. Adversarial robustness through disentangled representations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(4):3145–3153, May 2021. doi: 10.1609/aaai.v35i4.16424. URL `https://ojs.aaai.org/index.php/AAAI/article/view/16424`.

Shaokai Ye, Kaidi Xu, Sijia Liu, Jan-Henrik Lambrechts, Huan Zhang, Aojun Zhou, Kaisheng Ma, Yanzhi Wang, and Xue Lin. Adversarial robustness vs model compression, or both?, 2021. URL `https://arxiv.org/abs/1903.12561`.

Jirong Yi, Raghu Mudumbai, and Weiyu Xu. Derivation of information-theoretically optimal adversarial attacks with applications to robust machine learning, 2020. URL `https://arxiv.org/abs/2007.14042`.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy, 2019. URL `https://arxiv.org/abs/1901.08573`.

Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient neural network robustness certification with general activation functions, 2018. URL `https://arxiv.org/abs/1811.00866`.

A Toy Models of Superposition Replication

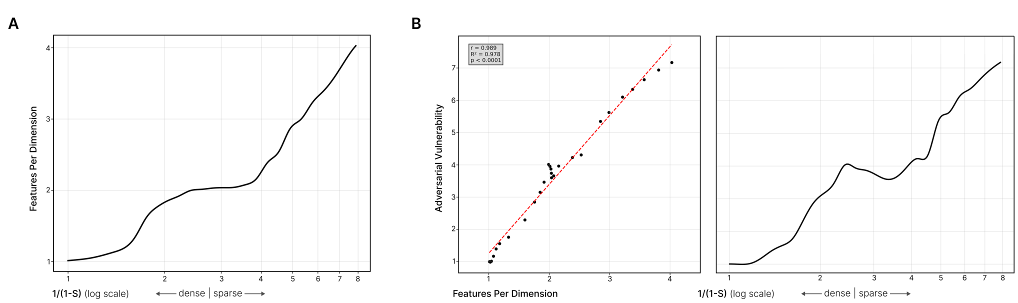

Figure 7: **Sparsity controls superposition, which drives adversarial vulnerability.** (A) Features per dimension increases with sparsity level 1/(1-S), with phase transitions at ~ 1.7 and ~ 4 corresponding to the onset of superposition and beyond-antipodal arrangements Elhage et al. [2022]. (B) Left: Adversarial vulnerability increases with feature sparsity. Right: Direct correlation between superposition (features per dimension) and adversarial vulnerability ($r \approx 0.99, p < 0.0001$). Each point represents a model trained at different sparsity. Results shown for Elhage-style attacks; see Figure 2 for gradient-based attacks.

B Sparse Autoencoder Training Details

All SAEs were trained with a batch size of 4096, a learning rate of 5×10^{-4}, and an expansion factor of $8\times$. Activations from models trained with different epsilons had slightly different distributions. Thus, for SAE training, activations were standardized using the mean and standard deviation for that specific model computed over a subset of the training data.

When training TopK SAEs, top-k_{aux} was 512 and the auxiliary loss weight was 1.

C Supplementary L0 Statistics

Table 2: L0 activation values (mean ± SEM) for clean and adversarial images across network layers. Statistics computed from n=100,000 images per condition.

Layer	Clean L0	FGSM L0	PGD L0
conv1	35.958 ± 0.0256	46.201 ± 0.0201	40.353 ± 0.0230
layer1	32.876 ± 0.0195	42.426 ± 0.0102	42.092 ± 0.0081
layer2	61.630 ± 0.0266	67.601 ± 0.0151	75.876 ± 0.0097
layer3	72.798 ± 0.0341	74.987 ± 0.0282	101.469 ± 0.0227
layer4	126.016 ± 0.0680	128.128 ± 0.0643	351.368 ± 0.1148

Revision History

15th September, 2025 There was a bug in the plotting code for Figure 6 increasing the difference between clean and adversarial images which has now been fixed.