Adversarial Attacks Leverage Interference Between Features in Superposition

Edward Stevinson, Lucas Prieto, Melih Barsbey, Tolga Birdal
Imperial College London
e.stevinson22@imperial.ac.uk

Abstract

Fundamental questions remain about why adversarial examples (AExs) arise in neural networks (NNs). In this paper, we argue that adversarial vulnerability can emerge from *efficient* information encoding in NNs. Specifically, we show that superposition – where networks represent more features than they have dimensions – creates arrangements of latent representations that adversaries can exploit. We demonstrate that adversarial perturbations leverage interference between superposed features to craft attacks, making attack patterns predictable from feature arrangements. Our framework provides a mechanistic explanation for two known phenomena: adversarial attack transferability between models with similar training regimes and class-specific vulnerability. In synthetic settings with precisely controlled superposition, we establish that superposition *suffices* to create adversarial vulnerability. We then demonstrate that these findings persist in a vision transformer (ViT) trained on CIFAR-10. These findings reveal adversarial vulnerability can be a byproduct of networks' representational compression, rather than flaws in the learning process or non-robust inputs. We release our implementation under: https://circle-group.github.io/AdversarialSuperposition/.

1 Introduction

Despite extensive research on AExs [Szegedy et al., 2014, Goodfellow et al., 2014, Bartoldson et al., 2024], no consensus exists on their fundamental causes, leaving us unable to predict which perturbations succeed, explain why attacks transfer between models, or design principled defences. This paper presents a mechanistic interpretability perspective demonstrating that AExs can exploit interference between learned representations in superposition—a mechanism that normally enables additional representation capacity—to craft effective perturbations that manipulate model outputs.

Existing explanations broadly fall into two camps [Nakkiran, 2019]: the *bug* perspective views adversarial perturbations as exploiting model-specific vulnerabilities in arbitrary directions unrelated to the true distribution [Schmidt et al., 2018, Nakkiran, 2019, Fawzi et al., 2016], whilst the *feature* perspective argues they exploit predictive but non-robust statistical patterns in the data [Ilyas et al., 2019]. Neither approach reconciles how representational constraints interact with data semantics. We bridge this divide, demonstrating how adversarial vulnerability emerges from the interaction between architectural constraints and data semantics (*i.e.* human-interpretable properties of the data).

Our account draws on the linear representation hypothesis (LRH) [Park et al., 2024] and the theory of superposition [Elhage et al., 2022]. The LRH posits that *input features*—fundamental abstractions of data—are represented as linear directions in a network's representation space. It is hypothesised that NNs can represent significantly more of these features than they have neurons through superposition, enabling networks to efficiently pack multiple features into shared dimensions at the cost of introducing interference. Such interference means perturbing one feature can affect others in non-obvious ways. This paper investigates whether this interference creates vulnerabilities that AExs exploit, and what insights this offers for understanding adversarial phenomena.

Mechanistic Interpretability Workshop at NeurIPS 2025.

Contributions: Using toy models with controlled superposition under projected gradient descent (PGD) attacks we reveal a mechanistic pathway: input correlations constrain feature arrangements, these arrangements determine interference patterns, and these interference patterns dictate attack characteristics and transferability. This framework enables prediction of which perturbations succeed and why they transfer between models. We replicate these findings in a ViT trained on CIFAR-10 with an engineered bottleneck. This framework reveals adversarial vulnerability can arise from efficient information encoding rather than learning flaws or non-robust features.

2 Background

We outline the tools used in our analysis: the LRH, superposition, and PGD, with full definitions in App. A. Let $\mathbf{x} \in \mathcal{X}$ denote the input and $\mathbf{h}^{(l)} \in \mathbb{R}^{d_l}$ the activation vector of the l-th layer.

Linear representation and superposition. The LRH posits that NNs represent many variables of their computation such as semantic properties of their inputs, as linear directions in activation space, which can be used as abstractions for reasoning [Park et al., 2024, Guerner et al., 2023]. We conceptualise these as a set of M semantically meaningful **latent features** (*e.g.* concepts such as "presence of shape") as linear directions $\{\mathbf{v}_j\}_{j=1}^{M} \subset \mathbb{R}^{d_l}$, such that $\mathbf{h}^{(l)}(\mathbf{x}) \approx \sum_{j=1}^{M} a_j(\mathbf{x})\, \mathbf{v}_j$, where $a_j(\mathbf{x}) \in \mathbb{R}$ represents the activation magnitude. Superposition occurs when $M > d_l$: networks represent more features than dimensions using non-orthogonal directions $\{\mathbf{v}_j\}_{j=1}^{M}$, enabled by sparse feature activation ($\mathbb{E}_{\mathbf{x}}[\|\mathbf{a}(\mathbf{x})\|_0] \ll M$). This creates polysemanticity—individual neurons representing multiple features—and interference [Elhage et al., 2022], where activating one feature activates others.

Adversarial attacks. Adversarial attacks force misclassification via small perturbations $\boldsymbol{\delta}$: $\mathbf{x}' = \mathbf{x} + \boldsymbol{\delta}$ with $\|\boldsymbol{\delta}\|_p \leq \epsilon$. We use PGD [Madry et al., 2018], which for untargeted attacks iteratively maximises loss: $\mathbf{x}'^{(k+1)} = \Pi_S(\mathbf{x}'^{(k)} + \alpha \mathbf{g}_k)$, where $\mathbf{g}_k$ is the normalised gradient, α is step size, and Π_S projects onto the ϵ-ball. For ℓ_∞ constraints, $\mathbf{g}_k = \text{sign}(\nabla_{\mathbf{x}} \mathcal{L}(\cdot))$; for ℓ_2, $\mathbf{g}_k = \nabla_{\mathbf{x}} \mathcal{L} / \|\nabla_{\mathbf{x}} \mathcal{L}(\cdot)\|_2$.

3 Superposition geometry determines adversarial attacks

We investigate if and how adversarial attacks exploit the interference rooted in the superposition of latent features. Specifically, we ask three questions:

1. *Do adversarial perturbations exploit the interference between superposed features?*
2. *Do correlations in the input shape the geometric arrangement of superposed latent features?*
3. *Can shared latent geometry explain why attacks transfer between independently trained models?*

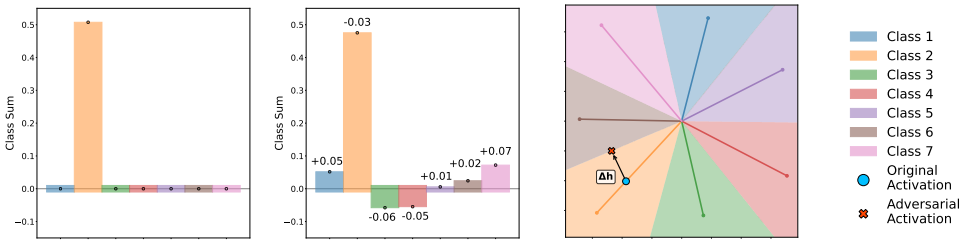
Setup. We create a synthetic task designed to: (1) provide an intuitive classification setting for studying AExs; (2) represent class concepts as linear directions per the LRH; (3) induce controlled superposition between these latent features; and (4) retain a priori knowledge of how inputs correspond to the superposed features – enabling testable predictions about adversarial mechanisms.

The input $\mathbf{x} \in \mathbb{R}^d$ is partitioned into k groups $\mathbf{x} = [\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(k)}]$, where $\mathbf{x}^{(j)} \in \mathbb{R}^p$ represents input features for class j. The task identifies which group has the largest sum: $y = \text{argmax}_{j \in \{1,\ldots,k\}} \sum_{i=1}^{p} x_i^{(j)}$. Each $x_i^{(j)} \sim \text{Uniform}(0, 1)$ at sparsity S (probability of being zero). A two-layer network with encoder $\mathbf{h} = \sigma(\mathbf{W}_e \mathbf{x} + \mathbf{b}_e) \in \mathbb{R}^m$ (where $m < k < d$) compresses k class representations into m dimensions, followed by a linear decoder for classification. Our primary setup uses cross entropy (CE) loss without ReLUs/biases. Since input feature $x_i^{(j)}$ affects only class j, the columns $\{\mathbf{W}_e[:, i] : i \in \text{class } j\}$ align, and we interpret them as class representations $\mathbf{v}_j$. AEx are generated with PGD and must (1) change the model's prediction (*e.g.* from class A to B) whilst (2) preserving the true class, *i.e.* the class with the largest sum remains the same (*e.g.* still class A).

3.1 Empirical results

We present our empirical findings qualitatively here, but provide quantitative details in App. C. Results remain consistent across a range of hidden dimensions, classes, and features per class.

Attacks exploit geometric interference. An attack must move a sample across a decision boundary to change its class, but what determines the required input perturbations? Two contrasting intuitions exist: the *feature intuition* suggests increasing input features of the target class, while the *bug intuition*



(a) Original Input (b) Adversarial Input (c) Latent Features, Activations & Boundaries

Figure 1: **An adversarial attack exploiting superposition geometry** ($k = 7$, $m = 2$). **(a)** The original sample. **(b)** The adversarially perturbed sample, whose ground truth remains the same but is misclassified. The sign and magnitude of an input perturbation is determined by the configuration of latent representations. **(c)** The original and adversarial sample in activation space. The coloured arrows are the column vectors of $\mathbf{W}_e$, the latent representations of the input features.

suggests adversarial perturbations modify inputs in unpredictable ways, with knowledge of feature representations providing no insight.

Fig. 1 shows a typical AEx in our setting, showing an input of Class 2 (orange) and its perturbed value that misclassifies it as Class 6 (brown). The perturbations appear arbitrary – we clearly do not simply increase the target class features, contradicting the feature intuition. Yet these perturbations are not random either. Instead, they follow a precise pattern with a systematic correspondence between $\boldsymbol{\delta}$ and the configuration of latent features. This relationship suggests a third intuition: *attacks are mediated by latent geometry*. Each input feature is perturbed (both magnitude and sign) in proportion to how its latent representation aligns with the vector travelled to cross the decision boundary ($\Delta\mathbf{h} = \mathbf{h}_{adv} - \mathbf{h}_{orig}$). We quantify this observation by comparing PGD-discovered attacks with theoretically optimal perturbations that we show leverage superposition (derived in Sec. 3.2), finding near-perfect alignment.

Finding: Adversarial attacks systematically exploit interference between superposed features. Successful PGD attacks are predictable given specific superposition geometry, rather than arbitrary.

Correlations determine geometry. Data correlations constrain how features arrange in latent space. Fig. 2 shows three correlation patterns: (a) with i.i.d. data, representations order randomly between different model seeds; (b) with pairwise correlations (input feature pairs that co-activate, as per Elhage et al. [2022]) models develop partially constrained structures with correlated features orthogonal; (c) with global correlations (cyclic correlations where adjacent classes are more likely to co-occur), models converge to a fixed ordering (up to rotation). The mechanism at play is *interference avoidance*: frequently co-activating features are arranged to minimise mutual interference.

Finding: Input correlations constrain feature geometry. Stronger correlations reduce geometric degrees of freedom, forcing different initialisations to converge to similar arrangements.

Shared superposition geometry explains transferability. When data correlations create consistent latent geometries between models, models share similar interference patterns between superposed features. As indicated in Fig. 2, this shared interference determines adversarial transfer rates: the globally correlated models yield 94% transfer versus 18% for the uncorrelated condition. Each perturbation component amplifies or suppresses feature activations, creating constructive interference that pushes representations across decision boundaries. However, when the same perturbation is applied to a model with different feature arrangements, features that previously constructively interfered instead cancel out or interfere destructively, causing the attack to fail.

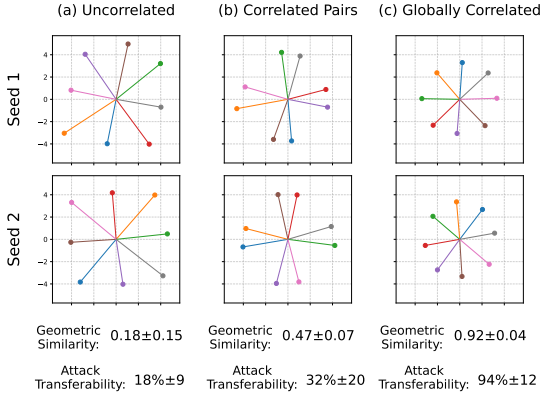


Figure 2: Greater input correlations create more consistent geometries between initialisations, driving attack transferability from 18% (uncorrelated) to 94% (global). Error ranges show standard deviations.

Finding: Attack transferability is governed by shared interference patterns.

Removing superposition eliminates attacks. With $m = k$, networks learn orthogonal class representations and no successful attacks exist: moving a sample across the decision boundary requires genuinely changing the class (*i.e.* changing the class with the highest sum). Furthermore, forcing one class feature orthogonal to others leaves its inputs unperturbed during attacks between other classes, confirming interference is necessary for these attacks (Appendix Fig. 4).

Finding: Adversarial attacks use their budget to exploit those features in superposition.

3.2 Formal analysis

We now formalise the mechanisms underlying our empirical observations. In our linear setting, we can derive exact relationships between superposition interference and adversarial vulnerability, extending our results beyond observational correlation.

We adopt the notation from our experimental setup, where $\mathbf{W}_e \in \mathbb{R}^{m \times d}$ with columns $\mathbf{v}_i$ induces superposition, and the decoder corresponds to the encoder transposed (as empirically observed).

Proposition 1. *The optimal input perturbations $\boldsymbol{\delta}$ that maximise movement from class j to class k under constraint $\|\boldsymbol{\delta}\|_2 = \epsilon$ satisfy $\boldsymbol{\delta} \propto \mathbf{W}_e^\top \mathbf{n}$, where $\mathbf{n} = (\mathbf{v}_k - \mathbf{v}_j)$ is the normal to the decision boundary between classes.*

Corollary 1 (Interference drives vulnerability) *The adversarial perturbation magnitude for feature i is $|\delta_i| \propto |\mathbf{v}_i^\top (\mathbf{v}_k - \mathbf{v}_j)|$, directly proportional to the differential interference between feature i and the class representations.*

This reveals how superposition creates adversarial vulnerability. Each input feature i is perturbed proportionally to its interference with the class representations. Under superposition, the non-orthogonality means semantically unrelated features interfere with the class decision—adversarial perturbations exploit these dependencies to manipulate outputs.

Proposition 2. *Models with feature representations related by orthogonal transformation $\mathbf{Q}$ (where $\mathbf{Q}^\top \mathbf{Q} = \mathbf{I}$) share identical optimal attack directions in input space.*

Together with our empirical findings, these propositions establish a **mechanistic pathway**. Specifically: (i) input correlations constrain feature arrangements in superposition, (ii) these geometric arrangements determine interference patterns, (iii) interference patterns dictate optimal perturbations via $\boldsymbol{\delta} \propto \mathbf{W}_e^\top (\mathbf{v}_k - \mathbf{v}_j)$ (Proposition 1), and (iv) shared geometric constraints yield similar interference patterns across models, enabling transferability (Proposition 2):
Correlations $\xrightarrow{\text{constrain}}$ **Feature Geometry** $\xrightarrow{\text{determines}}$ **Interference Patterns** $\xrightarrow{\text{enable}}$ **Transferability**

4 Attacks in vision models

We extend our analysis to a ViT [Dosovitskiy et al., 2020] trained on CIFAR-10 [Krizhevsky, 2009] with an engineered bottleneck to induce controlled superposition between class representations.

Setup. We train ViTs (6 layers, 512-dim embeddings, 81% accuracy) on CIFAR-10. We then replace the classification head on these base models with a bottleneck: a linear encoder that projects down to m dimensions followed by a decoder back to the 10 classes. We train this bottleneck with frozen ViT weights, and the class representations are placed in superposition. We vary $m \in \{2, 3, 5, 10\}$ to control compression degree. AExs are generated using ℓ_∞- and ℓ_2-norm PGD and transferability measured across five seeds. See App. D for complete results.

Results. Three key findings emerge that mirror our toy model observations from Sec. 3:

(1) As bottleneck dimension decreases, normalised robust accuracy decreases (81%→60%) and attack transferability across different model initialisations increases (25%→45%) (Fig. 3, right). The increased transferability follows from our findings that higher superposition reduces the degrees of freedom in potential feature geometry. With a more constrained representational space available, the network has fewer viable geometric arrangements for its class features. This leads to different model initialisations converging to more similar superposition geometries and, consequently, more shared interference patterns, resulting in greater attack transferability.

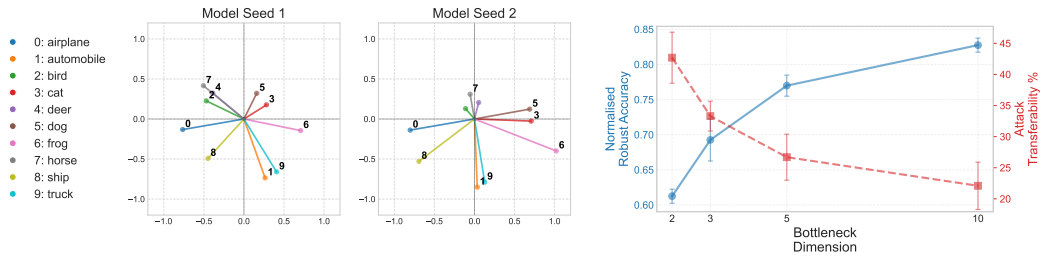


Figure 3: Left: CIFAR-10 class representation structure remains similar between models across different seeds. Right: Attack transferability and robust accuracy as bottleneck dimension is increased.

(2) Despite random initialisation, models converge to similar class arrangements, as measured by cosine similarity (Fig. 3, left). This consistency, akin to findings in Sec. 3, suggests that correlations guide the formation of a superposed geometry. Notably, semantically related classes ('cat'/'dog', 'car'/'truck') consistently cluster together.

(3) When AExs from bottlenecked models are run through the base ViTs they produce similar relative logit changes—both models respond consistently to the same perturbations. This indicates the perturbations are not solely an artefact of the bottleneck. However, without the increased compression these do not lead to misclassifications.

5 Related work

Superposition & latent representations. Research identifies data correlations as shaping latent arrangements: Elhage et al. [2022] demonstrate correlated features become orthogonal; Chan [2024] identify correlations as driving superposition; and Prieto et al. [2025] as creating semantic clustering. Gurnee et al. [2023] discuss interference patterns and mitigation via non-linearities. Representation geometry is further shaped by factors including spectral biases [Rahaman et al., 2019], neural collapse [Kothapalli, 2023], and optimisation objectives [Casper, 2023].

Explanations of adversarial vulnerability broadly fit into two camps, the *bug* perspective and the *feature* perspective [Nakkiran, 2019]. Bug perspectives include limited training data [Schmidt et al., 2018], finite-sample overfitting [Nakkiran, 2019], and properties of decision boundaries in high-dimensional spaces [Fawzi et al., 2016]. The feature perspective proposes AExs exploit predictive yet non-robust statistical patterns in the data [Ilyas et al., 2019]. Elhage et al. [2022] suggest superposition's link to adversarial examples, built on by Gorton and Lewis [2025], and debated by Casper [2023]). Other works describe perturbations pushing representations across boundaries [Zhang et al., 2021]; show PGD targets final layers [Ganeshan et al., 2019]; and find dataset-specific patterns Maiya et al. [2021]. Transferability is attributed to representation similarities [Li et al., 2023, Wang et al., 2024], with Wiedeman and Wang [2022] reducing transfer by decorrelating features between models.

6 Discussion & concluding remarks

We demonstrate that adversarial attacks can exploit interference patterns arising from superposed feature geometry in NNs. Data properties—correlations and sparsity—induce specific superposition geometries creating predictable vulnerabilities. These geometric arrangements determine attack characteristics and explain phenomena including transferability and class-specific susceptibility. This mechanistic account reveals superposition as a sufficient condition for adversarial vulnerability. Our new perspective frames adversarial vulnerability as a potential, inherent consequence of how networks efficiently encode vast amounts of information via superposition.

Limitations & future work. Our insights derive from simplified settings which use class features in engineered superposition. As large-scale models involve interference between unknown, unlabelled features across multiple layers, future work should study such interference, as well as investigating different attack types and how robust training reshapes feature geometry.

Acknowledgments

L. Prieto was supported by the UKRI Centre for Doctoral Training in Safe and Trusted AI [EP/S0233356/1]. M. Barsbey was supported by the EPSRC Project GNOMON (EP/X011364/1). T. Birdal acknowledges support from the Engineering and Physical Sciences Research Council [grant EP/X011364/1]. T. Birdal was supported by a UKRI Future Leaders Fellowship.[grant number MR/Y018818/1].

References

Brian R. Bartoldson, James Diffenderfer, Konstantinos Parasyris, and Bhavya Kailkhura. Adversarial robustness limits via scaling-law and human-alignment studies. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, 2024.

Leonard Bereska and Stratis Gavves. Mechanistic interpretability for AI safety - a review. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.

Stephen Casper. EIS IX: Interpretability and adversaries. *AI Alignment Forum*, February 2023.

Lawrence Chan. Superposition is not "just" neuron polysemanticity. *AI Alignment Forum*, 2024.

Valérie Costa, Thomas Fel, Ekdeep Singh Lubana, Bahareh Tolooshams, and Demba Ba. From flat to hierarchical: Extracting sparse representations with matching pursuit. *arXiv preprint arXiv:2506.03093*, 2025.

A Dosovitskiy, L Beyer, A Kolesnikov, D Weissenborn, X Zhai, T Unterthiner, M Dehghani, M Minderer, G Heigold, S Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, 2022.

Alhussein Fawzi, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Robustness of classifiers: from adversarial to random noise. *Advances in neural information processing systems*, 29, 2016.

Aditya Ganeshan, Vivek BS, and R Venkatesh Babu. FDA: Feature disruptive attack. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8069–8079, 2019.

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. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, 2025*, 2025.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Liv Gorton and Owen Lewis. Adversarial examples are not bugs, they are superposition, 2025.

Clément Guerner, Anej Svete, Tianyu Liu, Alexander Warstadt, and Ryan Cotterell. A geometric notion of causal probing. *arXiv preprint arXiv:2307.15054*, 2023.

Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. Finding neurons in a haystack: Case studies with sparse probing. *Trans. Mach. Learn. Res.*, 2023.

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.

Nikolaus Howe, Ian McKenzie, Oskar Hollinsworth, Michał Zajac, Tom Tseng, Aaron Tucker, Pierre-Luc Bacon, and Adam Gleave. Scaling trends in language model robustness, 2025.

A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems 32*, 2019.

Vignesh Kothapalli. Neural collapse: A review on modelling principles and generalization. *Trans. Mach. Learn. Res.*, 2023.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. CIFAR-10 dataset.

Victor Lecomte, Kushal Thaman, Rylan Schaeffer, Naomi Bashkansky, Trevor Chow, and Sanmi Koyejo. What causes polysemanticity? an alternative origin story of mixed selectivity from incidental causes. *arXiv preprint arXiv:2312.03096*, 2023.

Ang Li, Yifei Wang, Yiwen Guo, and Yisen Wang. Adversarial examples are not real features. *Advances in Neural Information Processing Systems*, 2023.

J Lindsey, W Gurnee, E Ameisen, B Chen, A Pearce, N L Turner, C Citro, D Abrahams, S Carter, B Hosmer, J Marcus, M Sklar, A Templeton, T Bricken, C McDougall, H Cunningham, T Henighan, A Jermyn, A Jones, A Persic, Z Qi, T B Thompson, S Zimmerman, K Rivoire, T Conerly, C Olah, and J Batson. On the biology of a large language model. *Transformer Circuits Thread*, 2025.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations*, February 2018.

Shishira R. Maiya, Max Ehrlich, Vatsal Agarwal, Ser-Nam Lim, Tom Goldstein, and Abhinav Shrivastava. A Frequency Perspective of Adversarial Robustness. *arXiv*, 2021.

Preetum Nakkiran. A discussion of 'adversarial examples are not bugs, they are features': Adversarial examples are just bugs, too. *Distill*, 2019.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024.

L Prieto, E Stevinson, M Barsbey, T Birdal, and P Mediano. Correlations in the data lead to semantically rich feature geometry under superposition. *Mechanistic Interpretability Workshop at NeurIPS 2025*, 2025.

N Rahaman, A Baratin, D Arpit, F Draxler, M Lin, F Hamprecht, Y Bengio, and A Courville. On the spectral bias of neural networks. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, California, USA*, Proceedings of Machine Learning Research. PMLR, 2019.

Shashata Sawmya, Linghao Kong, Ilia Markov, Dan Alistarh, and Nir N Shavit. Wasserstein distances, neuronal entanglement, and sparsity. In *The Thirteenth International Conference on Learning Representations*, 2025.

Adam Scherlis, Kshitij Sachan, Adam S Jermyn, Joe Benton, and Buck Shlegeris. Polysemanticity and capacity in neural networks. *arXiv preprint arXiv:2210.01892*, 2022.

Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. In *Advances in Neural Information Processing Systems, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 5019–5031, 2018.

C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations, ICLR*, 2014.

Tomasz Tkocz. An upper bound for spherical caps. *The American Mathematical Monthly*, 119(7): 606–607, 2012.

Donghua Wang, Wen Yao, Tingsong Jiang, Xiaohu Zheng, Junqi Wu, and Xiaoqian Chen. Improving the Transferability of Adversarial Examples by Feature Augmentation. *arXiv*, 2024.

Christopher Wiedeman and Ge Wang. Disrupting adversarial transferability in deep neural networks. *Patterns*, 3(5), 2022.

Shufei Zhang, Zhuang Qian, Kaizhu Huang, Qiufeng Wang, Rui Zhang, and Xinping Yi. Towards better robust generalization with shift consistency regularization. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pages 12524–12534. PMLR, 2021.

Appendix

A Definitions

The LRH posits that NNs represent many variables of their computation, such as semantic properties of their inputs, as linear directions in their activation space, which can be used as abstractions for reasoning [Park et al., 2024, Guerner et al., 2023]. This bias towards representing linear features is hypothesised because linear separability allows networks to easily recognise and manipulate features, and because dot products with subsequent layer weights efficiently process such directional features. Growing research supports this [Gurnee et al., 2023, Park et al., 2024]. Let $\mathcal{C} = \{c_1, \ldots, c_M\}$ denote a set of M semantically meaningful latent features (*e.g.* concepts like "presence of shape," or "indoor vs. outdoor"). Formally:

Definition 1 (Linear Representation Hypothesis (LRH)) *A neural network layer with activations $\mathbf{h}^{(l)} \in \mathbb{R}^{d_l}$ satisfies the LRH if it represents the latent features $\mathcal{C} = \{c_1, \ldots, c_M\}$ as linear directions $\{\mathbf{v}_j\}_{j=1}^{M} \subset \mathbb{R}^{d_l}$ such that:*

$$\mathbf{h}^{(l)}(\mathbf{x}) \approx \sum_{j=1}^{M} a_j(\mathbf{x})\, \mathbf{v}_j$$

*where $a_j(\mathbf{x}) \in \mathbb{R}$ represents the activation magnitude of feature c_j, and $\mathbf{v}_j$ is the corresponding linear direction. The features are **linearly accessible** [Costa et al., 2025]: inputs $\mathbf{x}_0, \mathbf{x}_1 \in \mathcal{X}$ that differ mainly in the value of feature c_j while holding other features approximately constant (i.e., $a_j(\mathbf{x}_1) - a_j(\mathbf{x}_0) = k$ and $|a_i(\mathbf{x}_1) - a_i(\mathbf{x}_0)| < \lambda$ for all $i \neq j$ and some small $\lambda > 0$) satisfy:*

$$\mathbf{h}^{(l)}(\mathbf{x}_1) - \mathbf{h}^{(l)}(\mathbf{x}_0) \approx k\mathbf{v}_j$$

where k reflects the change in c_j.

Superposition and sparsity NNs are capable of representing many more latent features than there are available dimensions in activation space: $M > d_l$. For example, LLMs can reference many more place names than they have residual stream dimensions. One framework for analysing this phenomenon is superposition, according to which networks use an overcomplete and non-orthogonal set of feature directions $\{\mathbf{v}_j\}_{j=1}^{M}$. This leverages the fact that $2^{\Theta(d\epsilon^2)}$ almost orthogonal vectors ($< \epsilon$ cosine similarity) can be represented in d-dimensional space [Tkocz, 2012], and that sparse vectors can be accurately recovered after projection into lower-dimensional spaces [Elhage et al., 2022, Bereska and Gavves, 2024, Sawmya et al., 2025]. This creates two challenges. First, *polysemanticity* [Scherlis et al., 2022, Lecomte et al., 2023] emerges where individual neurons contribute to multiple different features, meaning a neuron's activation does not correspond to a single interpretable concept. Second, non-orthogonal feature directions create *interference* between features – activating one feature activates others. Networks can mitigate these issues through non-linear operations (*e.g.* ReLU, softmax) that disambiguate superposed features [Gurnee et al., 2023].

Definition 2 (Superposition Hypothesis) *A network layer represents features in superposition if:*

1. *The number of latent features exceeds layer dimensionality: $M > d_l$*
2. *There exists non-orthogonal feature directions: $\exists\, i, j$ with $i \neq j$ such that $\mathbf{v}_i \cdot \mathbf{v}_j \neq 0$*
3. *Latent feature activations are sparse: $\mathbb{E}_{\mathbf{x}}[\|\mathbf{a}(\mathbf{x})\|_0] \ll M$, where $\mathbf{a}(\mathbf{x}) = [a_1(\mathbf{x}), \ldots, a_M(\mathbf{x})]^T$*

The network trades representational capacity against feature interference by packing more features than dimensions.

B Omitted proofs & theoretical results

To understand how adversarial attacks exploit feature representations, we prove that optimal perturbations weight each input dimension by how much its corresponding feature aligns with the path to the decision boundary. We analyse our linear models without ReLU activation functions.

Consider input $\mathbf{x} \in \mathbb{R}^d$ encoded via $\phi(\mathbf{x}) = \mathbf{W}_e\, \mathbf{x}$ to latent representation $\mathbf{h} \in \mathbb{R}^m$, where the columns of $\mathbf{W}_e \in \mathbb{R}^{m \times d}$ are an overcomplete dictionary $\{\mathbf{v}_i\}_{i=1}^{d}$ ($m < d$). For our argmax task,

we empirically observe that the trained encoder-decoder pair has the decoder as the transpose of the encoder: $\mathbf{W}_d = \mathbf{W}_e^\top$. This means the logit for class j is computed as $z_j = \mathbf{v}_j^\top \mathbf{h}$, where $\mathbf{v}_j$ is the j-th column of $\mathbf{W}_e$. The predicted class is $\hat{y} = \arg\max_j z_j$. Input perturbations map to latent perturbations as $\Delta\mathbf{h} = \mathbf{W}_e\boldsymbol{\delta}$, where $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_d)^\top$ are the perturbation coefficients.

The decision boundary between classes j and k is the set of points where their logits are equal: $\mathcal{B}_{jk} = \{\mathbf{h} : z_j = z_k\}$, or equivalently $\{\mathbf{h} : (\mathbf{v}_k - \mathbf{v}_j)^\top \mathbf{h} = 0\}$. The vector $\mathbf{n} = (\mathbf{v}_k - \mathbf{v}_j)$ acts as the normal to this pairwise decision boundary, pointing in the direction that maximally increases the margin $z_k - z_j$. We briefly recall the propositions before providing corresponding proofs.

Proposition 1 (Optimal targeted attack) The optimal input perturbations $\boldsymbol{\delta}$ that maximise movement from class j toward class k under constraint $\|\boldsymbol{\delta}\|_2 = \epsilon$ satisfy:

$$\boldsymbol{\delta} \propto \mathbf{W}_e^\top \mathbf{n}$$

Proof. To move a sample from being classified as j to being classified as k, we need to maximise the logit margin $z_k - z_j$. Under perturbation $\boldsymbol{\delta}$, the new margin becomes:

$$z'_k - z'_j = \mathbf{v}_k^\top(\mathbf{h} + \Delta\mathbf{h}) - \mathbf{v}_j^\top(\mathbf{h} + \Delta\mathbf{h}) \tag{1}$$

$$= (\mathbf{v}_k - \mathbf{v}_j)^\top(\mathbf{h} + \mathbf{W}_e\boldsymbol{\delta}) \tag{2}$$

$$= (\mathbf{v}_k - \mathbf{v}_j)^\top\mathbf{h} + (\mathbf{v}_k - \mathbf{v}_j)^\top\mathbf{W}_e\boldsymbol{\delta} \tag{3}$$

The change in margin is $(\mathbf{v}_k - \mathbf{v}_j)^\top\mathbf{W}_e\boldsymbol{\delta} = \boldsymbol{\delta}^\top\mathbf{W}_e^\top(\mathbf{v}_k - \mathbf{v}_j)$. We seek:

$$\max_{\boldsymbol{\delta}} \boldsymbol{\delta}^\top\mathbf{W}_e^\top(\mathbf{v}_k - \mathbf{v}_j) \quad \text{subject to} \quad \|\boldsymbol{\delta}\|_2 = \epsilon$$

Let $\mathbf{g} = \mathbf{W}_e^\top(\mathbf{v}_k - \mathbf{v}_j)$. By the Cauchy-Schwarz inequality:

$$|\boldsymbol{\delta}^\top\mathbf{g}| \leq \|\boldsymbol{\delta}\|_2\|\mathbf{g}\|_2 = \epsilon\|\mathbf{g}\|_2$$

Equality is achieved when $\boldsymbol{\delta}$ and $\mathbf{g}$ are parallel. Given the constraint $\|\boldsymbol{\delta}\|_2 = \epsilon$:

$$\boldsymbol{\delta} = \frac{\epsilon}{\|\mathbf{W}_e^\top(\mathbf{v}_k - \mathbf{v}_j)\|_2}\mathbf{W}_e^\top(\mathbf{v}_k - \mathbf{v}_j)$$

□

Corollary (Interference drives vulnerability) For a targeted attack from class j to class k, the adversarial perturbation magnitude for input feature i is:

$$|\delta_i| \propto |\mathbf{v}_i^\top(\mathbf{v}_k - \mathbf{v}_j)|$$

where $\mathbf{v}_i$ is the i-th column of $\mathbf{W}_e$. This quantity represents the differential interference between feature i and the class representations.

This reveals the mechanism by which superposition creates adversarial vulnerability. Each input feature i is perturbed proportionally to $\mathbf{v}_i^\top(\mathbf{v}_k - \mathbf{v}_j)$—the differential interference between feature i and the class representations. Under superposition, the non-orthogonality means that even semantically unrelated features have non-zero inner products with $(\mathbf{v}_k - \mathbf{v}_j)$, creating exploitable interference patterns. The multi-class setting amplifies this vulnerability, as with k classes there are multiple possible pairwise boundaries, each creating a distinct interference pattern. Adversarial perturbations leverage these cross-feature dependencies—they manipulate features that affect it through their interference with the class representations. This explains the vulnerability we observe empirically: attacks succeed not by directly increasing target class features, but by exploiting the web of interference created by superposition.

Prop. 1 characterises the optimal perturbation direction for moving from class j to k, providing the gradient for maximising the margin $z_k - z_j$. When intervening classes exist (where $z_j < z_i < z_k$ for some class i), following this gradient might cause the model to predict i before reaching target k. Iterative methods like PGD handle this by recomputing gradients at each step—the global attack path emerges from repeated local decisions rather than a single optimisation. Furthermore, while our analysis assumes $\mathbf{W}_d = \mathbf{W}_e^\top$ based on our empirical observations, the framework extends to arbitrary decoders. In the general case, with decoder $\mathbf{W}_d \in \mathbb{R}^{k\times m}$ having rows $\mathbf{w}_j^\top$, the optimal perturbation becomes $\boldsymbol{\delta} \propto \mathbf{W}_e^\top(\mathbf{w}_k - \mathbf{w}_j)$, capturing interference between encoder directions and decoder weights.

Proposition 2 (Attack transferability) Consider encoders ϕ and ψ with basis matrices $\mathbf{W}_e \in \mathbb{R}^{m\times d}$ and $\mathbf{W}'_e \in \mathbb{R}^{m\times d}$ whose columns are related by orthogonal transformation $\mathbf{v}'_i = \mathbf{Q}\mathbf{v}_i$ (where $\mathbf{Q}^\top\mathbf{Q} = \mathbf{I}$). If both models use their encoder transpose as decoder (i.e., $\mathbf{W}_d = \mathbf{W}_e^\top$ and $\mathbf{W}'_d = (\mathbf{W}'_e)^\top$), then both models have identical optimal input perturbation vectors for any targeted attack from class j to class k.

Proof. For model ϕ, the optimal perturbation from Prop. 1 is:

$$\boldsymbol{\delta}^\phi \propto \mathbf{W}_e^\top(\mathbf{v}_k - \mathbf{v}_j)$$

For model ψ with transformed columns $\mathbf{v}'_i = \mathbf{Q}\mathbf{v}_i$:

$$\boldsymbol{\delta}^\psi \propto (\mathbf{W}'_e)^\top(\mathbf{v}'_k - \mathbf{v}'_j) = (\mathbf{W}'_e)^\top(\mathbf{Q}\mathbf{v}_k - \mathbf{Q}\mathbf{v}_j)$$

Since $\mathbf{W}'_e = \mathbf{Q}\mathbf{W}_e$ (all columns are transformed), we have:

$$\boldsymbol{\delta}^\psi \propto (\mathbf{Q}\mathbf{W}_e)^\top\mathbf{Q}(\mathbf{v}_k - \mathbf{v}_j) \tag{4}$$

$$= \mathbf{W}_e^\top\mathbf{Q}^\top\mathbf{Q}(\mathbf{v}_k - \mathbf{v}_j) \tag{5}$$

$$= \mathbf{W}_e^\top(\mathbf{v}_k - \mathbf{v}_j) \tag{6}$$

where the last equality uses $\mathbf{Q}^\top\mathbf{Q} = \mathbf{I}$. Thus $\boldsymbol{\delta}^\phi$ and $\boldsymbol{\delta}^\psi$ have identical proportionality. Under the same norm constraint $\|\boldsymbol{\delta}\|_2 = \epsilon$, we have $\boldsymbol{\delta}^\phi = \boldsymbol{\delta}^\psi$. □

Prop. 2 is used to explain why attacks transfer between models with similar training regimes. When models learn feature representations that differ only by orthogonal transformation—essentially the same geometric structure in different orientations—they share identical vulnerability patterns in input space. The orthogonal transformation preserves all inner products between features, maintaining the interference patterns that attacks exploit. This provides an explanation for our empirical observation that models trained on data with the same correlations exhibit high attack transferability: they discover similar feature geometries up to rotation, leading to shared adversarial vulnerabilities.

C Toy model experiments

This section provides supplementary details, extended results and further discussion for the toy model experiments discussed in the main paper. We present model accuracies across a wider range of parameters than shown in the main text, offering insight into how model capacity and data characteristics like sparsity influence the learning process and the conditions under which feature superposition appears. Subsequently, we offer additional visual examples that correspond to Fig. 1, illustrating the mechanics of adversarial attacks under various conditions.

C.1 Hypotheses testing framework

We explicitly state our hypotheses for the three research questions in Sec. 3.

Research Q1: Do adversarial perturbations exploit superposition geometry?

- H_0: Adversarial perturbations are random with respect to feature geometry.
- H_1: Adversarial perturbations systematically exploit geometric relationships between superposed representations.

Research Q2: Do data correlations determine superposition geometry?

- H_0: Input correlations have no systematic effect on learned geometries.
- H_1: Input correlations determine geometric arrangements across model initializations.

Research Q3: Does shared geometry explain attack transferability?

- H_0: Attack transferability is independent of geometric similarity.
- H_1: Transferability increases with shared latent structure.

We test these hypotheses through controlled experiments:

- $H_1(1)$: We measure the input perturbation profile alignment with a class's latent representation and the latent attack vector.
- $H_1(2)$: We systematically vary input correlations and measure resulting geometries.
- $H_1(3)$: We quantify transferability rates across models with varying geometric similarity.

C.2 Accuracy of toy model for a range of parameters

The toy model experiments presented in the main paper predominantly used low-dimensionality settings for conceptual clarity. To demonstrate the model's behaviour more broadly, this subsection details the classification accuracies achieved by the CE toy model. These results are presented across varying hidden layer size (h), number of classes (k), number of features, and levels of sparsity (S), to provide insight into when the models learn to represent features in superposition. The sparsity level represents the probability that any individual input feature $x_i^{(j)}$ is set to zero, with higher values of S indicating greater input sparsity. We report results as a function of feature density $1 - S$, the probability that a feature is non-zero. Tab. 1 and Tab. 2 provide context on the model's performance limits and its ability to learn latent representations in superposition.

Table 1: Classification accuracy of the CE toy model with a fixed bottleneck dimension ($m = 2$) across various numbers of classes (k), total input features (features = $k \times 3$), and input feature densities ($1 - S$). These results illustrate how input sparsity controls performance degradation as the number of classes to be superposed within a constrained latent space increases.

Classes (k)	Features	Hidden (m)	Accuracy at Input Feature Density ($1 - S$)							
			1.0	0.57	0.33	0.19	0.11	0.06	0.04	0.02
3	9	2	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
4	12	2	0.67	0.71	0.82	0.92	0.98	0.99	1.00	1.00
5	15	2	0.53	0.50	0.65	0.77	0.89	0.95	0.98	0.99
6	18	2	0.40	0.43	0.51	0.66	0.82	0.93	0.97	0.99
7	21	2	0.34	0.34	0.40	0.53	0.73	0.87	0.95	0.98
8	24	2	0.30	0.30	0.33	0.40	0.63	0.82	0.93	0.97
9	27	2	0.24	0.26	0.30	0.35	0.57	0.75	0.89	0.96
10	30	2	0.22	0.24	0.26	0.31	0.50	0.72	0.87	0.95
15	45	2	0.14	0.15	0.16	0.17	0.25	0.40	0.65	0.86
20	60	2	0.10	0.10	0.11	0.13	0.15	0.24	0.44	0.76
25	75	2	0.07	0.08	0.09	0.10	0.12	0.16	0.26	0.62
30	90	2	0.06	0.07	0.07	0.07	0.09	0.11	0.21	0.45

C.3 Quantitative results from Section 3

This subsection provides the quantitative details and statistical analyses for the empirical findings presented in Section 3.

Alignment between PGD and optimal attacks. To evaluate whether PGD attacks specifically exploit superposition geometry, we compare PGD-generated attacks against theoretically optimal perturbations derived in Sec. 3.2. For each configuration (k, m), we train 5 models with different random seeds and generate 1000 PGD attacks per model, calculating the cosine similarity between each successful perturbation and the corresponding optimum. We establish a random baseline by generating perturbations with matching ℓ_2-norm and computing their similarity to the theoretical optimum.

Tab. 3 shows near-perfect alignment between PGD and optimal attacks across various dimensionalities. We filter out samples that have an ℓ_2-norm less than ϵ to ensure meaningful perturbations. One-sample t-tests comparing observed similarities against the random baseline yield $p < 10^{-10}$ for all configurations, demonstrating that PGD attacks systematically match theoretical predictions rather than occurring by chance. Given that optimal attacks leverage interference and PGD attacks achieve near-perfect alignment with these optima, we conclude that PGD exploits superposition interference.

Table 2: Classification accuracy of the CE toy model for varying numbers of classes (k), total input features, bottleneck dimensions (m), and input feature sparsity levels (S). This table reports different numbers of features per class (p).

Classes (k)	Features	Hidden (m)	Accuracy at Input Feature Density ($1 - S$)							
			1.0	0.57	0.33	0.19	0.11	0.06	0.04	0.02
30	30	30	0.23	0.24	0.38	0.62	0.83	0.94	0.99	1.00
30	90	90	0.27	0.27	0.33	0.41	0.51	0.67	0.85	0.94
40	40	30	0.67	0.54	0.73	0.77	0.88	0.96	0.99	0.99
40	120	30	0.71	0.64	0.65	0.72	0.73	0.79	0.89	0.96
60	60	10	0.05	0.07	0.12	0.25	0.47	0.73	0.90	0.97
60	180	10	0.08	0.10	0.13	0.17	0.22	0.32	0.52	0.75
80	80	30	0.15	0.17	0.23	0.41	0.63	0.79	0.91	0.98
80	240	30	0.23	0.22	0.31	0.41	0.48	0.53	0.66	0.80
100	100	10	0.03	0.04	0.05	0.10	0.21	0.43	0.69	0.87
100	300	10	0.04	0.05	0.06	0.09	0.12	0.15	0.25	0.45

Table 3: Alignment between PGD-discovered attacks and theoretically optimal perturbations across configurations. Results show mean $\pm$ std over 1000 attacks per condition. All p-values are below 10^{-10}.

k	m	Cosine Sim. (PGD vs Theory)	Cosine Sim. (Random Baseline)
6	2	0.97 ± 0.02	0.00 ± 0.02
30	10	0.96 ± 0.00	0.00 ± 0.01
90	30	0.92 ± 0.00	0.00 ± 0.00

Geometric similarity across configurations. We quantify geometric similarity by comparing the pairwise cosine similarity matrices between all feature pairs in each model, then measuring the correlation between these matrices across different random seeds. Higher correlation indicates more similar geometric arrangements. For each correlation condition, we tested 25 seed pairs.

Tab. 4 displays geometric similarity results across correlation types and superposition configurations. The results show a consistent monotonic relationship across all configurations: uncorrelated data yields highly variable geometries across seeds, paired correlations partially constrain the arrangements, whilst global correlations force near-identical geometries. Statistical significance was assessed using pairwise two-sample t-tests between all three correlation modes with Bonferroni correction ($p < 10^{-3}$ for all comparisons, $\alpha = 0.05$ corrected for 3 comparisons).

Table 4: Geometric similarity results across correlation types and superposition configurations.

Correlation Type	k	m	Geometric Similarity
Uncorrelated	6	2	0.18 ± 0.15
	30	10	0.17 ± 0.02
	90	30	0.17 ± 0.01
Paired	6	2	0.47 ± 0.07
	30	10	0.26 ± 0.07
	90	30	0.25 ± 0.01
Global	6	2	0.92 ± 0.04
	30	10	0.88 ± 0.01
	90	30	0.80 ± 0.01

Attack transferability. Using the same seed pairs as above, we generate adversarial attacks on source models and evaluate their success when applied to target models within each correlation condition (25 transfer measurements per condition). All models achieve $> 95\%$ clean accuracy. Transfer rates correlate strongly with geometric similarity. For the $k = 7, m = 2$ configuration: globally correlated data yields $94\% \pm 12\%$ attack transfer (mean $\pm$ std), uncorrelated data shows only $18\% \pm 9\%$ transfer, and paired correlations produce intermediate results of $32\% \pm 20\%$ transfer.

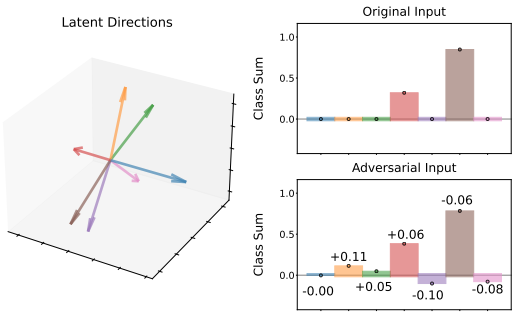


Figure 4: An adversarial attack (from class 5 to class 3) does not perturb the input features for a class represented orthogonally (class 1).

Orthogonal feature experiments. We test whether attacks focus on superposed features by constraining one class's latent vector $\mathbf{v}^{(\perp)}$ to remain orthogonal to all others during training. When generating attacks between classes that remain in superposition (*e.g.* from class a to b where $\mathbf{v}_a \not\perp \mathbf{v}_b$), the inputs corresponding to the orthogonal class remain unperturbed (Fig. 4). This confirms that attacks use their budget to exploit those features in superposition.

Additionally, when $m = k$, the network learns to represent each class direction $\mathbf{v}^{(j)}$ orthogonally. In this configuration, any perturbation that changes the model's prediction also changes the ground truth class. To move a sample from class A to class B requires making the sum of class B features exceed class A's—genuinely transforming it into a class B sample. We find zero successful adversarial examples across 1000 attempts at all ϵ values tested. Furthermore, fixing m and varying k shows that robust accuracy decreases monotonically with superposition pressure k/m (Tab. 5), demonstrating that vulnerability scales with interference degree.

C.4 How the toy model maps to real-world models

How does scaling networks affect adversarial vulnerability? While empirical evidence suggests that larger models tend to be more robust to adversarial attacks, this effect is weak. When adversarial training is employed, clearer scaling trends emerge, but improvements remain largely specific to the attack type used during training rather than conferring general robustness [Howe et al., 2025].

There are two key phenomena at play when scaling up models. On one hand, larger models have more capacity to represent concepts, but on the other hand, there seems to be a long tail of useful concepts for a larger model to capture in general tasks like next token prediction over internet text. This means that despite increased model capacity, superposition appears to be prevalent even in frontier models [Lindsey et al., 2025]. Supporting evidence comes from dictionary learning methods: SAEs require increasingly large dictionaries for larger models [Gao et al., 2025], suggesting that the number of features scales with model size. This suggests that the fundamental tension driving superposition – that models must compress many features into limited dimensions – does not disappear with scale.

Since both superposition and adversarial vulnerability persist in large-scale models, we believe our insights remain relevant across model scales. It is an interesting future avenue to understand how the geometry of superposition changes with scale, potentially helping to mitigate vulnerability.

C.4.1 Separating superposition effects from capacity reduction

Controlling for capacity by keeping the bottleneck dimension fixed to isolate superposition effects controls for the confounding effect of superposition pressure and capacity reduction. We fix bottleneck dimension $m = 2$, and vary number of classes k to isolate superposition pressure. We report the model

accuracy, robust accuracy, and transferability in Tab. 5. These controls demonstrate that increased superposition pressure (k/m), independent of model capacity, drives the adversarial vulnerabilities we observe.

Table 5: Performance metrics across correlation types, perturbation budgets (ε), and number of classes (k)

Correlation Type	**Parameters**		**Performance (%)**		
	ε	k	**Accuracy**	**Robust Acc.**	**Attack Trans.**
Uncorrelated	0.05	2	100.0	100.0	0.0
		4	100.0	94.9	21.2
		6	99.9	87.0	9.8
		8	99.8	77.7	17.5
	0.1	2	100.0	100.0	0.0
		4	100.0	88.1	9.8
		6	99.7	75.4	9.9
		8	99.6	56.3	23.5
	0.5	2	100.0	100.0	0.0
		4	100.0	99.4	0.0
		6	99.9	62.9	30.5
		8	99.5	24.9	39.0
Fully Correlated	0.05	2	100.0	100.0	0.0
		4	100.0	95.8	98.4
		6	97.3	70.6	98.4
		8	93.0	57.4	97.2
	0.1	2	100.0	100.0	0.0
		4	100.0	88.7	98.5
		6	97.7	45.6	99.7
		8	94.2	34.8	98.6
	0.5	2	100.0	100.0	0.0
		4	100.0	90.0	100.0
		6	97.2	34.6	100.0
		8	93.0	32.0	100.0

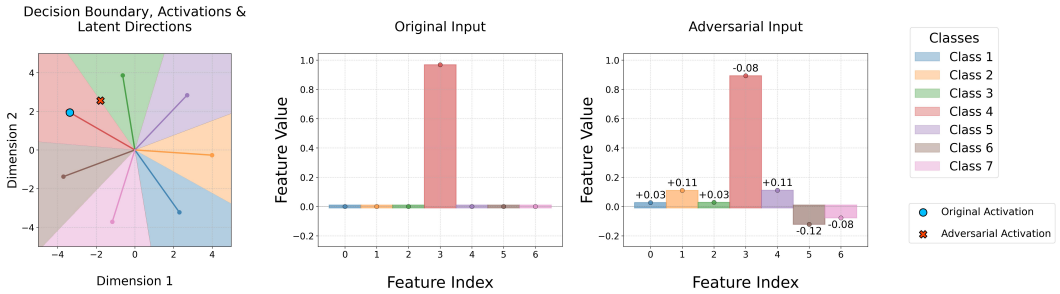
C.4.2 Additional examples of AExs in toy model

Fig. 1 demonstrates how adversarial attacks exploit the interference between latent features in superposition. Here we provide further visual examples (Fig. 5) to reinforce intuition. Specifically, we supplement the main text by showcasing:

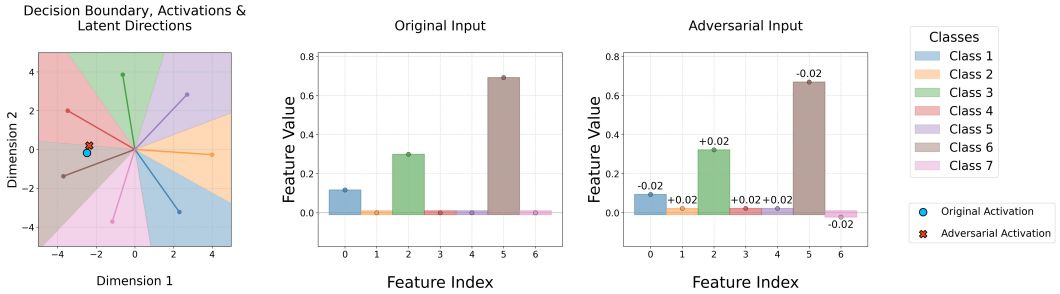
- Fig. 5a shows an additional instance of the setup in **??** ($m = 2$, $k = 7$) with an ℓ_2-norm PGD attack, demonstrating the IPP and latent space manipulations that lead to misclassification.
- Fig. 5b shows a similar setup ($m = 2$, $k = 7$) using an ℓ_∞-norm PGD attack on a less sparse input.
- Fig. 5c shows an example with increased bottleneck dimensionality ($m = 3$, $k = 7$) and accompanying ℓ_2-norm PGD attack. The feature vector similarity matrix used to calculate geometric similarity is also shown.

D CIFAR-10 experiments

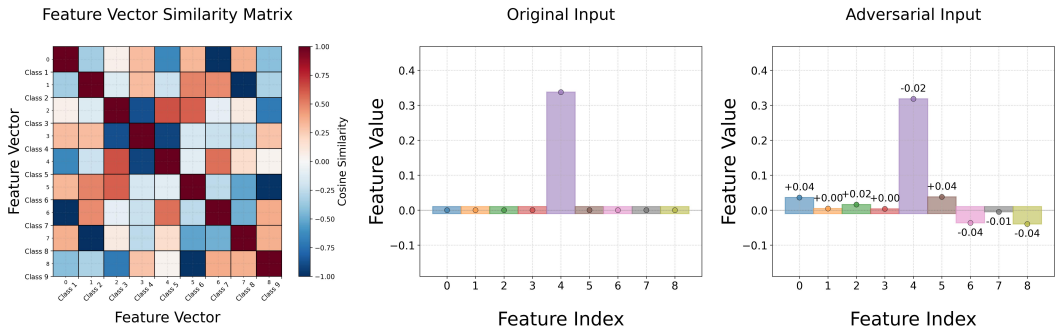
To investigate whether the principles observed in the toy models extend to more complex settings, Sec. 4 introduces experiments trained on CIFAR-10 [Krizhevsky, 2009] with an engineered bottleneck. This appendix section provides further details on this setup and presents extended results.



(a) An ℓ_2-norm attack changing the classification of an input of class 4 to class 3. The left plot shows original and adversarial activations in latent space, along with representation directions. The right plots show original and perturbed input feature values.



(b) An ℓ_∞-norm attack changing the classification of an input of class 6 to class 4. The left plot shows original and adversarial activations in latent space relative to class latent directions. The right plots show original and perturbed input feature values.



(c) An ℓ_2-norm adversarial attack in a 7-class setup with an increased bottleneck dimension $m = 3$. The leftmost plot now shows the cosine similarity matrix between pairs of latent representations for each of the classes.

Figure 5: Visualisations of AExs in the toy model, supplementing Figure 1 from the main paper by illustrating attack mechanisms in activation space and input space under varied conditions.

D.1 Architecture & training information

The base ViT [Dosovitskiy et al., 2020] architecture comprised 6 transformer layers, an embedding dimension (d) of 512, and 8 attention heads in each transformer layer. Input images from the CIFAR-10 dataset, sized at 32×32 pixels, were processed into patches of 4×4 pixels. The Multilayer Perceptron (MLP) within each transformer block had a hidden dimension of 512. Learned positional embeddings were used.

The bottleneck architecture consisted of a linear encoder followed by a linear decoder. The linear encoder projected the pre-classification activations obtained from the ViT backbone into an m-dimensional latent space ($m = \{2, 3, 5, 10\}$). The subsequent linear decoder then mapped these m-dimensional representations back to the $k = 10$ dimensions corresponding to the CIFAR-10 classes.

The base ViT model was trained on the CIFAR-10 dataset for 250 epochs. A learning rate of 0.001 was used with the Adam optimiser using default PyTorch parameters. The batch size was set to 512 and a cosine annealing learning rate scheduler was used. The loss function was CE. Dropout was used. Training was performed across five different random seeds to account for variability. After the base ViT model was trained, its weights were frozen. The bottleneck layer was then trained for 30 epochs, utilising a learning rate of 0.001. As for preprocessing, the images undergo RandomCrop with 4-pixel padding, resize to the target size (32x32 by default), and RandomHorizontalFlip for data augmentation.

D.2 Normalised robust accuracy across perturbation magnitudes

Fig. 3 (right) shows how normalised robust accuracy varies with bottleneck dimension for one value of ϵ. This subsection expands on those findings by detailing the normalised robust accuracy when subjected to PGD attacks of different strengths. Results are presented for both ℓ_2-norm (Tab. 7) and ℓ_∞-norm (Tab. 6) PGD attacks, providing a more comprehensive view of how the degree of superposition interacts with attack strength to affect model robustness.

AExs for these evaluations were generated using PGD with 100 iterations with a step size (α) of 0.01. Robust accuracy was evaluated on 500 samples for each configuration. The mean normalised robust accuracy and standard deviation across five random seeds are reported.

Table 6: Mean normalised robust accuracy ($\pm$ standard deviation across 5 seeds) for ViT models with different bottleneck dimensions (m) on CIFAR-10, subjected to ℓ_∞-norm PGD attacks of varying perturbation magnitudes (ϵ). Robust accuracy is normalised by the clean accuracy of each bottlenecked model. These results support the findings of Sec. 4, demonstrating a similar trend across ϵ.

ϵ	Bottleneck Dimension (m)			
	2	3	5	10
0.001	96.0% $\pm$ 0.4%	97.4% $\pm$ 0.7%	97.9% $\pm$ 0.4%	98.1% $\pm$ 0.3%
0.01	61.7% $\pm$ 3.6%	69.6% $\pm$ 3.5%	77.0% $\pm$ 0.9%	81.8% $\pm$ 3.2%
0.05	4.9% $\pm$ 1.5%	6.7% $\pm$ 1.6%	9.3% $\pm$ 0.6%	10.5% $\pm$ 0.8%
0.1	0.1% $\pm$ 0.2%	0.2% $\pm$ 0.3%	0.2% $\pm$ 0.4%	0.2% $\pm$ 0.4%
0.5	0.0% $\pm$ 0.0%	0.0% $\pm$ 0.0%	0.0% $\pm$ 0.0%	0.0% $\pm$ 0.0%

D.3 Attack transferability across perturbation magnitudes

We here include results on attack transferability across various perturbation magnitudes (ϵ) and bottleneck dimensions (m). Tab. 9 presents the ℓ_2-norm attack transferability and Tab. 8 ℓ_∞-norm attack transferability.

Table 7: Mean normalised robust accuracy ($\pm$ standard deviation across 5 seeds) for ViT models with different bottleneck dimensions (m) on CIFAR-10, subjected to ℓ_2-norm PGD attacks of varying perturbation magnitudes (ϵ). Robust accuracy is normalised by the clean accuracy of each bottlenecked model. These results support the findings in Section 4 of the main paper, showing decreasing robustness with smaller m (increased superposition) and larger ϵ.

ϵ	Bottleneck Dimension (m)			
	2	3	5	10
0.1	90.4% $\pm$ 1.7%	91.8% $\pm$ 0.7%	94.6% $\pm$ 1.2%	95.0% $\pm$ 0.7%
0.5	58.5% $\pm$ 5.0%	60.8% $\pm$ 5.0%	69.2% $\pm$ 1.8%	72.6% $\pm$ 2.8%
1.0	41.7% $\pm$ 4.8%	44.0% $\pm$ 3.0%	50.4% $\pm$ 0.6%	54.9% $\pm$ 1.7%
2.0	34.0% $\pm$ 4.6%	36.2% $\pm$ 3.3%	41.5% $\pm$ 1.3%	47.4% $\pm$ 2.4%
5.0	30.2% $\pm$ 4.1%	32.9% $\pm$ 3.5%	37.3% $\pm$ 1.9%	42.7% $\pm$ 1.0%

D.4 Correlations in class features

In Sec. 3 it was correlations between inputs that drove superposition arrangements. We note that here it is not the correlations between input classes but rather the correlations in the representations at this point in the network that drive these arrangements. At the classification layer this is likely similar to the confusion matrix – *i.e.* how each class is misclassified in relation to the other classes. Classes that cluster together are those the network finds inherently similar and misclassifies together. To test this we repeat the experiment, finetuning the base ViT using timm/vit_base_patch16_384 which has been trained on ImageNet-21k (14 million images, 21,843 classes) and ImageNet (1 million images, 1,000 classes). After fine-tuning on CIFAR-10 and applying the same bottleneck training procedure as Sec. 4, the resulting geometry shows random ordering between initialisations with approximately equal spacing between features (*i.e.* neuron collapse [Kothapalli, 2023]). In this case performance is near 100%, meaning the confusion matrix is the identity, and the superposed loses its structure. In contrast, models trained solely on CIFAR-10 converge to similar geometries because they share the same learned difficulty structure - the same pairs of classes prove challenging to distinguish, leading to consistent superposition patterns.

D.5 ResNet-18

To address concerns on architectural generalisation, we conduct the same experiments using a ResNet-18 [He et al., 2016] architecture as the base model as opposed to a ViT. We achieve a slightly higher 92% clean accuracy (compared to 89% for ViT). We observe the robust accuracy falls slightly faster for ResNet-18 than the ViT at the same ϵ values. Nevertheless, we observe similar declining trends in normalised robust accuracy and increasing transferability as bottleneck pressure increases.

Table 8: Attack transferability (%) for ℓ_∞-norm PGD attacks on CIFAR-10 ViT models. Transferability is shown from a model trained with a specific 'Source Seed' (e.g., Seed 10) to three different target models, each trained with one of the seeds listed in the sub-header (e.g., 'vs. Seeds 20/30/40'). The three slash-separated values in each cell correspond to the transferability to these three target seeds, respectively. All models within a row share the same bottleneck dimension, m. The 'Mean $\pm$ Std' column averages transferability across all 12 source-target seed pairings for each (ϵ, m) configuration. This supports the claim in Section 4 that higher superposition can lead to more consistent latent geometries and thus higher transferability.

ϵ	m	Seed 10 vs. Seeds 20/30/40	Seed 20 vs. Seeds 10/30/40	Seed 30 vs. Seeds 10/20/40	Seed 40 vs. Seeds 10/20/30	Mean $\pm$ Std
0.001	2	77.8/66.7/44.4	25.0/62.5/25.0	72.7/72.7/72.7	58.3/75.0/58.3	59.3 $\pm$ 17.7
	3	57.1/28.6/28.6	14.3/28.6/28.6	63.6/72.7/72.7	42.9/42.9/42.9	43.6 $\pm$ 18.4
	5	70.0/50.0/50.0	57.1/71.4/57.1	50.0/25.0/25.0	16.7/50.0/33.3	46.3 $\pm$ 16.9
	10	55.6/55.6/66.7	12.5/37.5/25.0	16.7/33.3/16.7	28.6/71.4/28.6	37.3 $\pm$ 19.3
0.01	2	60.3/52.6/48.7	63.8/42.0/60.9	53.8/40.9/52.7	51.2/55.8/46.5	52.4 $\pm$ 6.9
	3	42.9/46.4/50.0	46.4/42.9/49.1	44.2/38.9/44.2	42.9/45.1/41.8	44.6 $\pm$ 3.0
	5	43.3/41.1/43.3	39.2/30.4/41.8	46.0/40.2/47.1	41.2/35.3/32.9	40.2 $\pm$ 4.8
	10	42.5/46.0/44.8	38.0/39.4/46.5	43.5/47.8/47.8	32.6/40.0/35.8	42.1 $\pm$ 4.7
0.05	2	35.0/36.4/37.8	38.8/36.0/46.3	37.4/31.3/41.7	42.0/42.0/40.7	38.8 $\pm$ 3.8
	3	29.9/29.9/37.1	30.4/30.1/35.6	30.2/30.2/35.1	30.1/31.9/34.4	32.1 $\pm$ 2.6
	5	26.8/25.3/29.2	25.2/21.5/26.4	28.0/28.0/26.5	26.0/26.6/21.9	25.9 $\pm$ 2.2
	10	27.1/24.9/30.4	21.6/23.0/28.4	25.3/28.1/30.6	21.0/24.9/20.1	25.4 $\pm$ 3.4
0.1	2	34.2/35.6/38.2	36.7/34.9/44.5	37.9/30.0/40.8	41.2/42.0/40.3	38.0 $\pm$ 3.8
	3	29.6/28.9/35.2	28.6/29.2/34.8	29.1/28.2/34.0	28.8/31.1/32.8	30.9 $\pm$ 2.5
	5	25.4/23.3/27.0	23.8/19.6/24.6	25.9/26.7/25.4	24.6/24.3/20.7	24.3 $\pm$ 2.1
	10	24.8/22.6/29.7	20.1/21.6/26.1	22.5/25.5/28.0	19.5/23.3/18.7	23.5 $\pm$ 3.3

E Acronyms

AEx adversarial example

CE cross entropy

LRH linear representation hypothesis

MLP Multilayer Perceptron

MSE mean squared error

NN neural network

PGD projected gradient descent

SAE sparse autoencoder

ViT vision transformer

Table 9: Attack transferability (%) for ℓ_2-norm PGD attacks on CIFAR-10 ViT models. The table format, detailing source model to target model transferability, mirrors that of Tab. 8; please see its caption for a full explanation. These ℓ_2 results further support the claim in Sec. 4 that higher superposition leads to increased attack transferability.

ϵ	m	Seed 10 vs. Seeds 20/30/40	Seed 20 vs. Seeds 10/30/40	Seed 30 vs. Seeds 10/20/40	Seed 40 vs. Seeds 10/20/30	Mean $\pm$ Std
0.1	2	70.6/64.7/41.2	60.0/50.0/60.0	64.3/60.7/57.1	48.1/63.0/51.9	57.6 $\pm$ 8.0
	3	39.1/39.1/47.8	50.0/60.7/50.0	48.1/55.6/51.9	48.0/28.0/40.0	46.5 $\pm$ 8.4
	5	43.5/34.8/43.5	31.6/31.6/42.1	42.9/35.7/21.4	30.4/39.1/34.8	35.9 $\pm$ 6.4
	10	45.5/59.1/45.5	19.0/28.6/28.6	37.5/43.8/43.8	23.8/52.4/38.1	38.8 $\pm$ 11.4
0.5	2	59.6/52.1/48.9	60.8/35.4/60.8	54.1/39.4/51.4	43.8/49.5/46.7	50.2 $\pm$ 7.7
	3	38.7/40.6/48.1	36.1/36.8/41.7	37.4/36.5/43.5	34.4/34.4/36.7	38.7 $\pm$ 3.9
	5	38.4/32.0/37.6	29.4/25.7/32.1	39.1/33.9/38.3	31.8/29.9/32.7	33.4 $\pm$ 4.0
	10	38.3/37.4/37.4	26.5/28.4/31.4	32.4/35.3/35.3	23.4/28.2/25.0	31.6 $\pm$ 5.0
1.0	2	50.8/43.7/44.4	51.7/36.4/50.0	46.1/32.9/44.7	42.7/46.7/45.3	44.6 $\pm$ 5.3
	3	35.7/36.9/41.7	31.8/36.4/42.6	33.7/32.0/37.3	34.9/32.0/35.5	35.9 $\pm$ 3.3
	5	32.1/28.9/33.2	30.9/23.0/30.9	35.9/31.5/33.7	27.7/28.2/27.1	30.3 $\pm$ 3.3
	10	32.3/32.3/38.5	23.7/27.7/29.9	29.1/32.0/34.3	22.2/25.6/21.1	29.1 $\pm$ 5.0
2.0	2	47.0/41.6/44.3	48.2/36.2/48.2	43.9/33.5/42.1	43.4/43.4/42.8	42.9 $\pm$ 4.2
	3	31.9/37.7/40.3	31.1/35.6/40.6	32.6/31.1/36.8	35.4/33.3/34.3	35.1 $\pm$ 3.2
	5	29.9/25.4/32.1	26.3/22.1/26.7	32.3/31.8/29.5	27.9/27.9/23.6	28.0 $\pm$ 3.2
	10	30.8/30.4/35.7	22.5/24.9/28.6	24.8/29.7/32.7	20.4/26.1/19.0	27.1 $\pm$ 4.9
5.0	2	44.7/46.7/40.8	45.3/37.3/48.7	42.7/33.9/41.5	42.3/42.3/40.6	42.2 $\pm$ 3.8
	3	32.3/35.8/40.8	31.3/32.6/39.9	31.0/30.0/33.8	35.0/33.0/33.0	34.1 $\pm$ 3.2
	5	30.6/24.1/28.6	28.3/20.9/24.8	31.7/31.3/27.4	28.1/29.0/23.2	27.3 $\pm$ 3.3
	10	29.8/27.4/32.7	19.7/24.1/28.9	24.9/29.9/31.2	18.8/25.8/19.2	26.0 $\pm$ 4.6