Attacking the Representation Manifold: A Mechanistic Study of Adversarial Robustness in Modular Addition

Edward Stevinson [1] **Lucas Prieto** [1]

Abstract

Neural networks trained on modular addition learn algorithms whose latent representations factor through a torus-to-circle map, providing unusually complete knowledge of the learned algorithm and its representation geometry. We exploit this transparency to show how mechanistic knowledge allows us to predict the form of successful adversarial perturbations and how adversarial training reshapes representations to resist attack. We decompose adversarial perturbations on the embedding torus into phase-shifting and amplitude-changing components, predicting that efficient attacks target the same Fourier features the model uses. We confirm this empirically: the Fourier spectrum of successful PGD perturbations concentrates on the model's frequency features, mechanism-informed attacks restricted to those frequencies are competitive with white-box PGD, and attack transfer between models is predicted by feature overlap. The same mechanistic lens predicts that adversarial training increases robustness by broadening the model's frequency support, linking the representation change to capacity-robustness trade-offs. Modular addition thus provides a case study in which adversarial vulnerability becomes interpretable – vulnerability becomes a targeted failure of the learned algorithm, and robustness becomes a measurable restructuring that algorithm. Code is available at https://github.com/Stevinson/modulo-attack.

1. Introduction

Mechanistic interpretability aims to understand neural networks (NNs) by decomposing them into interpretable components such as features or circuits, and understanding how these interact to implement computation (Bricken et al., 2023; Gao et al., 2025; Lindsey et al., 2025; Templeton et al., 2024). One approach is to study small networks on tasks where the learned algorithm can be faithfully characterised (Elhage et al., 2022; Chughtai et al., 2023; Nanda et al., 2023). One such task is modular arithmetic: NNs trained to solve $(a, b) \mapsto (a + b) \bmod P$ learn sparse Fourier features and select their prediction via constructive interference in logit space (Gromov, 2023; McCracken et al., 2025; Nanda et al., 2023). This circuit is unusually well understood, with both the learned algorithm and representation geometry characterised.

We take advantage of this to ask whether such mechanistic knowledge can be used to predict the adversarial directions that most efficiently cause misclassification (Goodfellow et al., 2015; Szegedy et al., 2014), and how adversarial training (AT) reshapes the learned representations to resist them. These questions are currently intractable in larger models as neither the learned algorithm nor the representation manifold can be fully characterised. Modular addition therefore provides a useful testbed for mechanism-level robustness analysis: we can ask whether gradient-based adversarial perturbations are targeted failures of the learned algorithm rather than uninterpretable high-dimensional directions.

We approach this in three parts. First, we use the learned Fourier geometry to predict phase-shifting and amplitude-changing perturbation effects, and verify these predictions via the Fourier spectra of PGD attacks, mechanism-informed attacks, and transfer between independently trained models. Second, we measure how adversarial training reshapes this Fourier representation, finding broader frequency support subject to a capacity bottleneck. Third, we compare with statistical representation-geometry diagnostics from the robustness literature, showing the diagnostics move consistently with our mechanistic account, while only the mechanistic descriptor – being interpretable at the feature level – supports concrete predictions about transfer and attack structure. Our broader motivation is twofold: to show that mechanistic knowledge can yield insights that guide robustness analysis and that mechanistic interpretability can extend beyond characterising normal operation to predicting and explaining adversarial failure modes.

[1]Department of Computer Science, Imperial College London, London, United Kingdom. Correspondence to: Edward Stevinson <e.stevinson22@imperial.ac.uk>.

Mechanistic Interpretability Workshop at the 43rd International Conference on Machine Learning, Seoul, South Korea, 2026.
Copyright 2026 by the author(s).

Our main contributions are as follows:

1. We give a mechanism-level account of adversarial vulnerability in modular addition by identifying phase-shifting and amplitude-changing perturbation effects on the learned Fourier features.
2. We derive a phase sensitivity matrix mapping input perturbations to first-order feature phase shifts, and use it to construct low-dimensional mechanism-informed attacks.
3. We show empirically that successful PGD perturbations recover the model's key frequencies, that frequency-restricted attacks are competitive with full PGD, and that transfer between independently trained models is predicted by frequency overlap.
4. We show that adversarial training increases robustness by broadening frequency support, making robust algorithmic learning capacity-limited.
5. We compare this mechanistic account with statistical representation-based robustness diagnostics, showing that these agree with our account but on their own do not identify which mechanism produces robustness or what predicts transfer.

Together, these results provide initial support for our claim that understanding the internal mechanisms a model uses can reveal the structure that adversaries exploit.

2. Background

2.1. Modular addition and the learned Fourier algorithm

We study neural networks trained to solve modular addition: given $a, b \in \mathbb{Z}_P$ for prime P, predict $c^* = (a+b) \bmod P$. Each of a, b is represented by a one-hot vector $\boldsymbol{e}_a, \boldsymbol{e}_b \in \mathbb{R}^P$, and a shared embedding matrix $W_E \in \mathbb{R}^{d\times P}$ maps it to $\boldsymbol{h}_a = W_E\, \boldsymbol{e}_a$. The network then combines the embeddings for a and b, and passes them through non-linear layers to produce logits $\boldsymbol{z} \in \mathbb{R}^P$.

Fourier features in the embedding. Prior work has shown that the learned embeddings concentrate energy on a small set of Fourier frequencies, with each frequency contributing an approximately two-dimensional cosine-sine subspace (Gromov, 2023; McCracken et al., 2025; Nanda et al., 2023). For a trained model of the architectures we study, the embedding of a is well approximated by a small set of Fourier modes,

$$W_E \boldsymbol{e}_a \approx \sum_{k\in\mathcal{K}} A_k \left[\cos\left(\frac{2\pi k a}{P}\right)\boldsymbol{v}_{k,c} + \sin\left(\frac{2\pi k a}{P}\right)\boldsymbol{v}_{k,s}\right] \tag{1}$$

where $\mathcal{K}$ is a small set of *key frequencies*, $A_k > 0$ is the amplitude of frequency k, and $\boldsymbol{v}_{k,c}, \boldsymbol{v}_{k,s} \in \mathbb{R}^d$ are approximately orthonormal directions. Thus each key frequency gives a two-dimensional coordinate system in which the integer a is represented at angle $2\pi k a/P$.

Computing addition by phase addition. Once the inputs are represented as sines and cosines, modular addition is implemented using angle-addition identities. For each key frequency k, the hidden layers combine the features for a and b to compute quantities proportional to

$$\cos\left(\frac{2\pi k(a+b)}{P}\right), \quad \sin\left(\frac{2\pi k(a+b)}{P}\right).$$

This angle-addition computation was reverse-engineered by Nanda et al. (2023); McCracken et al. (2025) subsequently showed it is one implementation of a more general coset-intersection algorithm across architectures. In that account, the *simple neuron model* gives a local description of the same computation and clusters of neurons at several frequencies collectively implement an approximate Chinese Remainder Theorem over the modular sum (McCracken et al., 2025).

Constructive interference at the logits. The output layer reads out the per-frequency evidence for each candidate class. The logits are well approximated by (McCracken et al., 2025; Nanda et al., 2023)

$$z_c \approx \beta + \sum_{k\in\mathcal{K}} w_k \cos\left(\frac{2\pi k(a+b-c)}{P}\right), \tag{2}$$

where $w_k > 0$ is the contribution of frequency k. At the correct class $c^* = (a+b) \bmod P$, every cosine term equals 1, so all key frequencies vote in phase. At an incorrect class, the different frequencies generally disagree and partially cancel. The learned algorithm succeeds by *constructive interference*: the correct logit is the unique place where the frequency votes align. If the computation is distributed across many balanced key frequencies (McCracken et al., 2025), this suggests an attacker must disrupt several independent votes rather than a single dominant one.

The representation torus. The Fourier embedding has a direct geometric interpretation. For one frequency, the coordinates $A_k\left(\cos\frac{2\pi k a}{P}, \sin\frac{2\pi k a}{P}\right)$ place the integer a on a circle of radius A_k. Across the key frequencies, the embedding lies on a product of circles,

$$\mathcal{T}_{\mathcal{K}} = S^1(A_{k_1}) \times \cdots \times S^1(A_{k_K}), \quad K = |\mathcal{K}|.$$

In the mechanistic interpretability lens, the coordinate functions of this torus can be viewed as the linear value

features (Prieto et al., 2026): the cosine and sine coordinates in each key-frequency plane. Moisescu-Pareja et al. (2026) show that the embedding manifold takes one of two closed forms: a full torus when per-input phases vary independently, or a disc projection when phases are perfectly correlated. As the architecture we study sums embeddings before the MLP, it lies in the perfectly correlated regime, meaning per-input embeddings lie on $\mathcal{T}_{\mathcal{K}}$ and the post-summation representation is the disc projection.

2.2. Adversarial perturbations

An adversarial perturbation is a small input change that nonetheless causes misclassification. For input $\boldsymbol{x}$ with label y, an adversarial perturbation satisfies $\|\boldsymbol{\delta}\|_p \leq \varepsilon$ and $\operatorname{argmax}_c f(\boldsymbol{x}+\boldsymbol{\delta})_c \neq y$. To generate attacks, we use Projected Gradient Descent (PGD) (Madry et al., 2018),

$$\boldsymbol{x}^{(t+1)} = \Pi_{\mathcal{B}_p(\boldsymbol{x},\varepsilon)}\left(\boldsymbol{x}^{(t)} + \alpha\, \boldsymbol{g}^{(t)}\right), \tag{3}$$

where $\boldsymbol{g}^{(t)}$ is the normalised loss gradient with respect to the input, α is the step size, and $\Pi_{\mathcal{B}_p(\boldsymbol{x},\varepsilon)}$ denotes projection onto the norm ball.

In our setting the perturbed object is the one-hot input vector: $\boldsymbol{e}_a$ is replaced by the dense $\boldsymbol{e}_a + \boldsymbol{\delta}$, instantiating the standard norm-bounded, label-preserving adversarial threat model ("small" here meaning small in input norm rather than perceptually imperceptible). This leaves the discrete one-hot data manifold, analogous to standard image attacks where norm-bounded perturbations typically leave the natural-image manifold (Stutz et al., 2019). In both settings, discrete inputs (one-hot vectors here, pixel grids for images) pass through a learned linear embedding (W_E here, the patch projection in a ViT) into a continuous representation space, and PGD perturbations push the resulting representation off the learned manifold (App. Figure 5). What distinguishes modular addition is that both the task's intrinsic geometry and the learned representation manifold are known explicitly, facilitating our analysis.

3. Adversarial vulnerability of the learned algorithm

We use the description of the learned algorithm from §2 to predict which perturbation directions efficient adversarial attacks use and how AT will change representation geometry. The known torus geometry lets us decompose the embedding-space displacement at each key frequency into tangential motion around the learned circle and normal motion away from it. Our analysis yields three predictions:

1. Successful adversarial perturbations should concentrate Fourier energy at key frequencies.
2. Mechanism-informed attacks restricted to the key frequency subspace should be competitive with full input-space PGD.
3. AT should increase adversarial robustness by increasing the number of key frequencies, requiring more embedding dimensions.

3.1. From input perturbations to feature manipulation

Each clean sample lies on the learned circle at the correct angle for each key frequency. An input perturbation $\boldsymbol{\delta}$ induces an embedding-space displacement $W_E\boldsymbol{\delta}$; only its components in the active sine/cosine planes affect the learned features. Each in-plane component splits into a part that rotates the point along the circle – changing the angle the network reads at that frequency, and therefore the represented value – and a part that pushes it off the circle, changing the feature's amplitude. The first is the more algorithmically meaningful one: it changes the input the algorithm thinks it is processing.

Let an adversarial perturbation $\boldsymbol{\delta} \in \mathbb{R}^P$ modify the one-hot input to $\boldsymbol{x}' = \boldsymbol{e}_a + \boldsymbol{\delta}$. The perturbed embedding is

$$\boldsymbol{h}' = W_E\, \boldsymbol{x}' = \boldsymbol{h}_a + \boldsymbol{\eta}, \qquad \boldsymbol{\eta} := W_E\boldsymbol{\delta}. \tag{4}$$

For each key frequency k, the effect on the learned feature coordinates is

$$\tilde{\boldsymbol{\eta}}_k := V_k^\top \boldsymbol{\eta} = V_k^\top W_E\, \boldsymbol{\delta} \in \mathbb{R}^2, \tag{5}$$

where $V_k = [\boldsymbol{v}_{k,c} \mid \boldsymbol{v}_{k,s}]$ stacks the basis vectors of the two-dimensional cosine-sine feature plane for frequency k. We write φ_k for the induced change in the angular coordinate of this feature, and decompose $\tilde{\boldsymbol{\eta}}_k$ into a tangent component (rotating the feature around its circle) and a normal component (changing its radius).

Lemma 1 (Tangent/normal decomposition)**.** *At frequency k, write*

$$\boldsymbol{\rho}_k(a) = A_k \begin{pmatrix}\cos\theta_k \\ \sin\theta_k\end{pmatrix}, \qquad \theta_k = \frac{2\pi k a}{P},$$

$$\boldsymbol{t}_k = \begin{pmatrix}-\sin\theta_k \\ \cos\theta_k\end{pmatrix}, \qquad \boldsymbol{n}_k = \begin{pmatrix}\cos\theta_k \\ \sin\theta_k\end{pmatrix}.$$

Defining $\eta_k^T := \boldsymbol{t}_k^\top \tilde{\boldsymbol{\eta}}_k$ and $\eta_k^N := \boldsymbol{n}_k^\top \tilde{\boldsymbol{\eta}}_k$,

$$\tilde{\boldsymbol{\eta}}_k = \eta_k^T \boldsymbol{t}_k + \eta_k^N \boldsymbol{n}_k. \tag{6}$$

This is the orthonormal-basis decomposition of $\tilde{\boldsymbol{\eta}}_k \in \mathbb{R}^2$ in the tangent/normal frame at $\boldsymbol{\rho}_k(a)$.

Proposition 1 (First-order phase response)**.** *Let θ_k' denote the angular coordinate of the perturbed point $\boldsymbol{\rho}_k(a) + \tilde{\boldsymbol{\eta}}_k$ in*

the cosine-sine plane of frequency k, and let $\varphi_k := \theta'_k - \theta_k$ be the induced phase shift. For $\|\tilde{\boldsymbol{\eta}}_k\| \ll A_k$,

$$\varphi_k \approx \frac{\eta_k^T}{A_k} = \frac{\boldsymbol{t}_k^\top V_k^\top W_E \boldsymbol{\delta}}{A_k}. \tag{7}$$

Proof. The perturbed angle is $\theta'_k = \arctan(A_k \sin\theta_k + \tilde{\eta}_{k,2}, A_k \cos\theta_k + \tilde{\eta}_{k,1})$. A first-order Taylor expansion at $\tilde{\boldsymbol{\eta}}_k = 0$ gives $\theta'_k - \theta_k \approx \boldsymbol{t}_k^\top \tilde{\boldsymbol{\eta}}_k / A_k$. Substituting $\tilde{\boldsymbol{\eta}}_k = V_k^\top W_E \boldsymbol{\delta}$ yields the equality. □

3.2. Identifying sensitive input directions

We next pull this tangent direction back to input space, since attacks are applied to $\boldsymbol{e}_a$ rather than directly to feature space.

Definition 1 (Phase sensitivity matrix)**.** For each frequency $k \in \mathcal{K}$ and input a, define

$$\boldsymbol{g}_k(a) = \frac{1}{A_k} W_E^\top V_k \boldsymbol{t}_k \in \mathbb{R}^P, \qquad \varphi_k \approx \boldsymbol{g}_k(a)^\top \boldsymbol{\delta}. \tag{8}$$

Stacking these vectors gives the *phase sensitivity matrix*

$$G(a) = [\boldsymbol{g}_{k_1}(a) \mid \cdots \mid \boldsymbol{g}_{k_K}(a)] \in \mathbb{R}^{P \times K}, \qquad \boldsymbol{\varphi} \approx G(a)^\top \boldsymbol{\delta}. \tag{9}$$

$G(a)$ is the pullback of the learned tangent feature directions through the embedding matrix. Writing $\boldsymbol{q}_k(a) = V_k \boldsymbol{t}_k$ for the embedding-space tangent at frequency k, each column of $G(a)$ satisfies

$$\boldsymbol{g}_k(a) = A_k^{-1} W_E^\top \boldsymbol{q}_k(a). \tag{10}$$

We use this map to ask which input directions rotate many frequencies at once, whether PGD perturbations align with these directions, and whether adversarial training reshapes them.

3.3. Frequency redundancy and logit margins

In this Fourier description, increasing the number of key frequencies should improve robustness. In the constructive logit Eq. 2, the correct class $c^* = (a+b) \bmod P$ is the class where all key frequencies vote in phase. The clean margin against a competing class c' is therefore

$$m(c') = z_{c^*} - z_{c'} \approx \sum_{k \in \mathcal{K}} w_k \left[1 - \cos\left(\frac{2\pi k(c^* - c')}{P}\right)\right]. \tag{11}$$

This suggests adding sufficiently weighted key frequencies increases these margins because it adds further support that aligns at c^* and generally misaligns at wrong classes. A new frequency helps only insofar as it contributes independent logit evidence. This separates the vulnerability mechanism into two interpretable factors:

1. *Algorithmic redundancy*: determined by the number and weighting of key frequencies in the logit readout, per Eq. 11. More key frequencies force an attacker to disrupt more to flip the prediction.

2. *Embedding geometry*: determined by W_E, $\{V_k\}$, and $\{A_k\}$ via $G(a)$. Larger amplitudes reduce first-order phase sensitivity through the $1/A_k$ factor, while the singular structure of $G(a)$ controls whether a small input perturbation can rotate many key frequencies at once.

4. Experimental setup

We describe the model architecture, feature-identification procedure, attacks, and representation diagnostics used to test the predictions in §3.

4.1. Architecture and training

Our primary architecture is the MLP-Add (Moisescu-Pareja et al., 2026) for $P = 113$, following standard grokking setups (Nanda et al., 2023; Power et al., 2022): per-input embeddings of dimension $d = 128$ are summed, $\boldsymbol{s} = \boldsymbol{h}_a + \boldsymbol{h}_b$, and passed through a one-hidden-layer MLP ($n = 256$ ReLU units) and linear readout to P logits. We train 20 seeds with CE loss using AdamW (lr 10^{-3}, weight decay 1.0, full-batch) on a 70/30 split for 50,000 epochs, long enough for weight decay to clean up the memorisation circuit (Nanda et al., 2023); all reach $\geq 99.9\%$ test accuracy.

4.2. Attack methodology

We generate PGD attacks under the ℓ_2 norm. Only input a is perturbed, exploiting the symmetry of addition. We use $\varepsilon = 0.5$ with 100 PGD steps and step size $\alpha = 0.01$.

Mechanism-informed attacks. We evaluate *targeted* attacks (each driving the prediction to $c' = c^* + \Delta$); picking a specific wrong class is a strictly harder objective than untargeted misclassification, isolating the predictive power. We compare two attacks of increasing knowledge requirements.

Cosine injection (gradient-free). Using only $\mathcal{K}$ and a target shift Δ, we construct

$$\boldsymbol{\delta}^\Delta(j) = c \cdot \frac{2}{P} \sum_{k \in \mathcal{K}} \cos\left(\frac{2\pi k(j - a - \Delta)}{P}\right), \tag{12}$$

the projection of $\boldsymbol{e}_{a+\Delta}$ onto the input-side DFT basis at $\mathcal{K}$. We sweep c and report the smallest scale at which the model misclassifies to $c' = c^* + \Delta$.

Analytical phase attack. Using the phase sensitivity matrix

Table 1. Robustness diagnostics. Δ is the direction expected under AT (equations in App. C.5).

Diagnostic	What it measures	Δ
Geometric margin, m_{geom}	Logit distance to the decision boundary, accounting for sensitivity. Higher value directly reflects adversarial robustness.	$\uparrow$
Effective rep. dim. $\mathrm{PR}(\mathrm{Cov}(z))$	Participation ratio of the clean-representation covariance.	?
Sensitivity, ADR, $\lVert J_\Phi \rVert_F$, $\lVert J_\Phi \rVert_2$	How much input changes are amplified in representation space. High values mean perturbations can strongly affect internal representations.	$\downarrow$
Alignment, $\lvert\cos(\Delta z, \beta)\rvert$	Whether adversarial displacement points along the decision-changing direction. Perturbations may fail if they move in irrelevant directions.	$\downarrow$
Participation ratio	How low-rank the displacement covariance is across successful attacks, distinguishing diverse from concentrated attacks.	?
Principal-subspace fraction	Fraction of attack displacement lying in the clean-representation subspace. Low values indicate attacks exploit off-manifold directions.	?
Linearisation error	How poorly the Jacobian predicts actual adversarial displacement, revealing use of non-linear paths.	$\downarrow$

and per-frequency logit weights, we solve

$$\begin{aligned} \min_{\boldsymbol{\varphi}\in\mathbb{R}^K} \quad & \boldsymbol{\varphi}^\top (G^\top G)^{-1} \boldsymbol{\varphi} \\ \text{s.t.} \quad & \sum_{k\in\mathcal{K}} w_k \big[\cos\varphi_k - \cos\big(\tfrac{2\pi k \Delta_c}{P} + \varphi_k\big)\big] \le 0, \end{aligned} \tag{13}$$

$\Delta_c = (c^* - c') \bmod P$, for each candidate target class c'. The optimal $\boldsymbol{\varphi}^*$ recovers the input-space perturbation $\boldsymbol{\delta} = G(G^\top G)^{-1}\boldsymbol{\varphi}^*$.

Transfer attacks. For each pair of models, we generate PGD attacks on the source model and evaluate them on the target. The transfer rate is the fraction of successful source attacks that also cause misclassification on the target.

Adversarial training We use PGD-based adversarial training (PGD-AT) (Madry et al., 2018). During each training step, the input $\boldsymbol{e}_a$ is replaced by $\boldsymbol{e}_a + \boldsymbol{\delta}_{\mathrm{AT}}$, where $\boldsymbol{\delta}_{\mathrm{AT}}$ is a PGD adversarial perturbation. We train with ℓ_2-norm $\varepsilon_{\mathrm{AT}} = 0.05 - 0.3$, using 200 steps, $\alpha = 0.01$, for 50,000 epochs, and 10 random seeds.

4.3. Representation analysis

Representation-geometry diagnostics To compare our mechanistic measurements against statistical signatures of robustness, we define a taxonomy of representation-geometry diagnostics (Table 1). Each captures observable structure in how attacks move through the network.

Identifying key frequencies For each trained model, we identify the key frequencies $\mathcal{K}$ by computing the Discrete Fourier Transform (DFT) of the embedding matrix along its column dimension. For each frequency k, we compute the energy $E_k = \lVert F_{:,k} \rVert_2^2$ where $F = \mathrm{DFT}(W_E)$, normalise as $\hat{E}_k = E_k / \sum_j E_j$, and classify frequency k as a key frequency if $\hat{E}_k > \tau \max_{j>0} \hat{E}_j$, with $\tau = 0.1$. Having identified $\mathcal{K}$, we extract $\{\boldsymbol{v}_{k,c}, \boldsymbol{v}_{k,s}\}$ and $\{A_k\}$ by fitting least-squares linear probes $W_k \in \mathbb{R}^{2\times d}$ from $\boldsymbol{h}_a$ to $(\cos\theta_{k,a}, \sin\theta_{k,a})$, $\theta_{k,a} = 2\pi k a / P$, following (Prieto et al., 2026). As sanity checks, key frequencies achieve high held-out R^2 while non-key frequencies do not, and ablating the key-frequency subspace collapses test accuracy while ablating its complement preserves it (App. C.2, App. C.3).

5. Experiments

The analysis in §3 makes a mechanism-level prediction: if attacks exploit the learned modular-addition algorithm, their perturbations should be structured by the same Fourier features that implement clean computation. We evaluate this prediction by testing (1) whether *PGD rediscovers the model's key frequencies*, (2) *whether those frequencies support competitive low-dimensional attacks and predict transfer*, (3) *how adversarial training reparameterises the Fourier mechanism*, and (4) *whether statistical representation-level diagnostics recover the same explanation*.

5.1. Attacks utilise the learned frequency structure

Our framework predicts that efficient input perturbations need to change the phase of Fourier features to disrupt the constructive-interference in logits, whilst perturbation energy outside this subspace should be much less useful in modifying the network's prediction. We indeed see this Fourier concentration when running PGD attacks on all P^2 input pairs. For each successful perturbation $\boldsymbol{\delta}$, we compute $\lvert \mathrm{DFT}(\boldsymbol{\delta})_k \rvert$ and compare the attack spectrum with the model's key frequencies, as per §4.3 (see Figure 1). The attacks succeed on 100% of inputs at $\varepsilon = 0.5$ with 95.7% of perturbation Fourier energy concentrated at the key

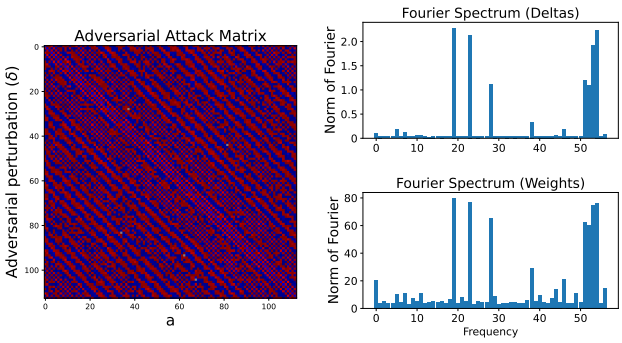

Figure 1. Mean successful perturbations $\boldsymbol{\delta}$ to input a across samples. The sinusoidal structure indicates that attacks concentrate on the Fourier frequencies.

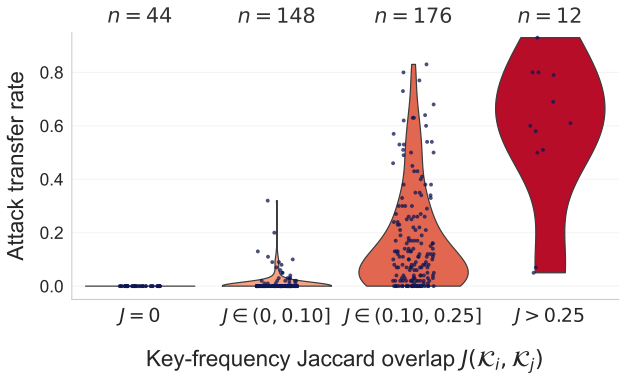

Figure 2. Attack transfer rate vs key-frequency Jaccard overlap. Each sample is a source/target model pair; violins show the distribution within each Jaccard bucket. Transfer is near-zero for pairs with no shared key frequencies and rises sharply, supporting that vulnerability is controlled by shared computational structure.

frequencies, far above chance ($\sim 17\%$). PGD concentrates perturbations on the features used by the model.

Mechanism-informed attacks succeed at comparable ε. We test whether mechanism knowledge enables targeted attacks competitive with PGD. Fixing $(a, b) = (0, 0)$ and sweeping over wrong classes c', the median minimum ε that lands on c' is: PGD 0.25, cosine injection (12) 0.33, analytical phase (13) 0.37; the Gaussian-noise baseline (1.10) solves a strictly easier untargeted problem. The ordering is expected: PGD's iterative gradient ascent exploits nonlinearities and non-phase effects the mechanism attacks ignore; cosine injection installs a clean target signal directly in the key-frequency subspace; the analytical phase attack pays a small further cost for solving the linearised surrogate.

Takeaway. The Fourier concentration and success of mechanism-informed attacks point to adversarial perturbations preferentially exploiting the features implementing the learned algorithm not arbitrary high-dimensional directions.

5.2. Attack transfer is predicted by shared frequencies

If adversarial vulnerability is tied to the learned frequencies, then attacks should transfer between models that share key frequencies and fail to transfer between models that implement the task using disjoint frequency sets. For each pair of models trained with different seeds, we generate PGD attacks on a source model and evaluate them on a target model. We compare the transfer rate with the Jaccard overlap of the key-frequency sets, $J(\mathcal{K}_i, \mathcal{K}_j) = {}^{|\mathcal{K}_i \cap \mathcal{K}_j|}/_{|\mathcal{K}_i \cup \mathcal{K}_j|}$. Across 380 model pairs in Figure 2, key-frequency overlap predicts transfer with a sharp threshold. Transfer is essentially zero below $J \approx 0.10$ and rises sharply above it, reaching a mean of 0.58 for the strongly overlapping pairs ($J > 0.25$). The threshold pattern supports the prediction that adversarial vulnerability is controlled by shared computational structure: a perturbation crafted against one model's Fourier features does not transfer to a second model unless the two share enough of those features.

Takeaway. The same task can be implemented by different frequencies, and adversarial perturbations transfer primarily when they overlap.

5.3. Adversarial training learns more frequencies, capacity-permitting

We next ask what happens when models are made robust to ℓ_2-norm attacks using PGD-AT. Do they preserve the trigonometric algorithm, and if so, does it change the frequency parameterisation?

Adversarial training broadens the frequency basis. We compare standard models with AT models at increasing $\varepsilon_{\mathrm{AT}}$-budgets in Figure 3, revealing three trends. First, AT broadens the Fourier support: $|\mathcal{K}|$ rises from 7 to 17 as $\varepsilon_{\mathrm{AT}}$ increases. Second, the embedding's effective rank and effective dimension rise in lockstep, with their ratio to $|\mathcal{K}|$ remaining ~ 2, as expected since each key frequency occupies a two-dimensional subspace. Most diagnostically, mean probe R^2 stays ≥ 0.9 for $\varepsilon_{\mathrm{AT}} \leq 0.25$ but collapses to 0.23 at $\varepsilon_{\mathrm{AT}} = 0.30$: $|\mathcal{K}|$ keeps growing but those frequencies are no longer cleanly linearly recoverable as $(\cos, \sin)$ pairs. The algorithm reparameterises consistently up to $\varepsilon_{\mathrm{AT}} \approx 0.25$, then transitions out of the cleanly trigonometric regime our framework describes.

Capacity determines robustness The above results suggest that adversarially robust representations demand additional capacity. We train models at five scales, $(d, n) = (32, 64) \rightarrow (512, 1024)$ crossed with four training regimes (standard plus $\varepsilon_{\mathrm{AT}} \in \{0.1, 0.2, 0.3\}$). The robustness frontier shifts upward with scale, with larger models learning more key frequencies.

$\varepsilon_{\mathrm{AT}}$	$\lvert\mathcal{K}\rvert$	$\bar{R^2}(\mathcal{K})$	Effective rank(W_E)	Effective dim$(\boldsymbol{z})$
0.00	7	0.92	17.7	17.9
0.05	9	0.95	18.8	19.0
0.10	10	0.87	20.3	20.7
0.15	11	0.87	23.3	24.9
0.20	15	0.94	27.6	30.5
0.25	16	0.90	27.3	31.3
0.30	17	**0.23**	28.3	31.3

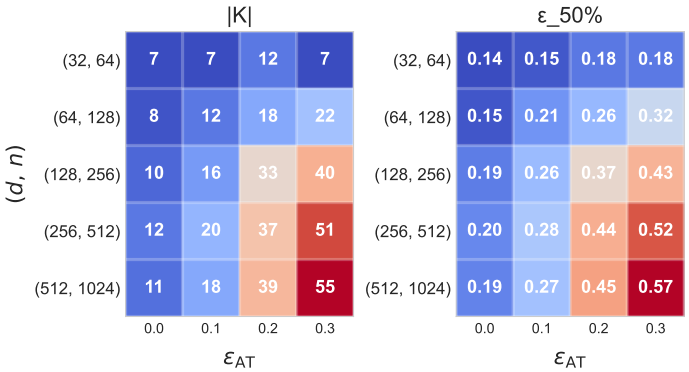

Figure 3. **Effect of adversarial training on representation geometry and capacity scaling.** *Left:* As the adversarial training budget $\varepsilon_{\mathrm{AT}}$ increases, $(d, n) = (32, 256)$, the number of key frequencies $|\mathcal{K}|$ grows and the embedding effective rank/dimension rise in lockstep, indicating that adversarial training (AT) reparameterises the algorithm rather than abandoning it, confirmed by probe R^2. R^2 collapses at $\varepsilon_{\mathrm{AT}} = 0.30$, signalling a transition out of the cleanly trigonometric regime. *Right:* The number of key frequencies $|\mathcal{K}|$ and empirical robustness threshold $\varepsilon_{50\%}$ as a function of model size (d, n) and AT budget. Increasing $\varepsilon_{\mathrm{AT}}$ broadens the Fourier support and shifts robustness upward. Larger models continue to add robustness, suggesting capacity is the bottleneck.

Takeaway. Below an ε-threshold, AT preserves the algorithm but reparameterises it with broader frequency support and larger effective dimension. Robust algorithmic learning via adversarial training is capacity-limited. Scaling the embedding dimension allows the model to preserve the algorithm while spreading it across more frequency dimensions.

5.4. Robustness diagnostic axes do not capture mechanism

We instrument each model with the taxonomy of representation-geometry diagnostics from Table 1 and run the same AT sweep. Figure 4 summarises all seven axes; every axis the framework predicts to move does so, in the predicted direction.

The axes move as expected. Encoder sensitivity falls, alignment with the decision gradient drops, robust accuracy and the geometric margin rise, and linearisation error falls modestly. This is consistent with the literature – AT creates less sensitive, more locally-linear encoders.

Effective dimension and participation ratio rise in step with $|\mathcal{K}|$. AT spreads the algorithm across more dimensions, and successful attacks span more independent directions because more frequencies do computational work. Each new key frequency contributes its own roughly two-dimensional subspace, forcing attacks to spread across them – exactly the framework's prediction.

What the mechanistic account adds. The taxonomy and the mechanistic descriptor agree that AT changes the representation; the mechanistic descriptor provides additional information, identifying what AT does to the algorithm itself – recruiting more key frequencies, redistributing amplitude across them, and reweighting their contributions to the constructive-interference logit. This added specificity supports three concrete predictions the diagnostics on their own do not make. *Transfer*: shared $\mathcal{K}$ predicts whether source-model attacks fool target models (Figure 2). *Attack structure*: successful PGD perturbations should concentrate Fourier energy at $\mathcal{K}$ (Figure 1). *Capacity scaling*: admissible AT budgets scale with d via $|\mathcal{K}| \leq d/2$ (Figure 3).

6. Related Work

Mechanistic interpretability of modular addition. Modular addition is a standard testbed for mechanistic interpretability. Nanda et al. (2023) reverse-engineered the learned trigonometric algorithm and used it to define progress measures for grokking (Power et al., 2022). Gromov (2023) and Morwani et al. (2024) analyse why sinusoidal representations emerge, while McCracken et al. (2025) show that multiple architectures converge to an approximate Chinese Remainder Theorem. Moisescu-Pareja et al. (2025) give a geometric picture, showing that these architectures factor the same torus-to-circle map through different intermediate representations. Related Fourier and cyclic-arithmetic structure has also been identified in large pretrained models (Feucht et al., 2026; Lindsey et al., 2025). We build on this understanding to characterise how the mechanism fails under adversarial perturbations.

Capacity and robustness. Bereska et al. (2025) use an information-theoretic measure to argue that the effect of AT depends on task complexity relative to model capacity, placing modular addition in an abundance regime. Scherlis et al. (2025) argue that features are represented polysemantically only when shared features have substantial marginal value. Our AT results are consistent with this: robustness is increased by increasing orthogonal Fourier features rather than by introducing superposition.

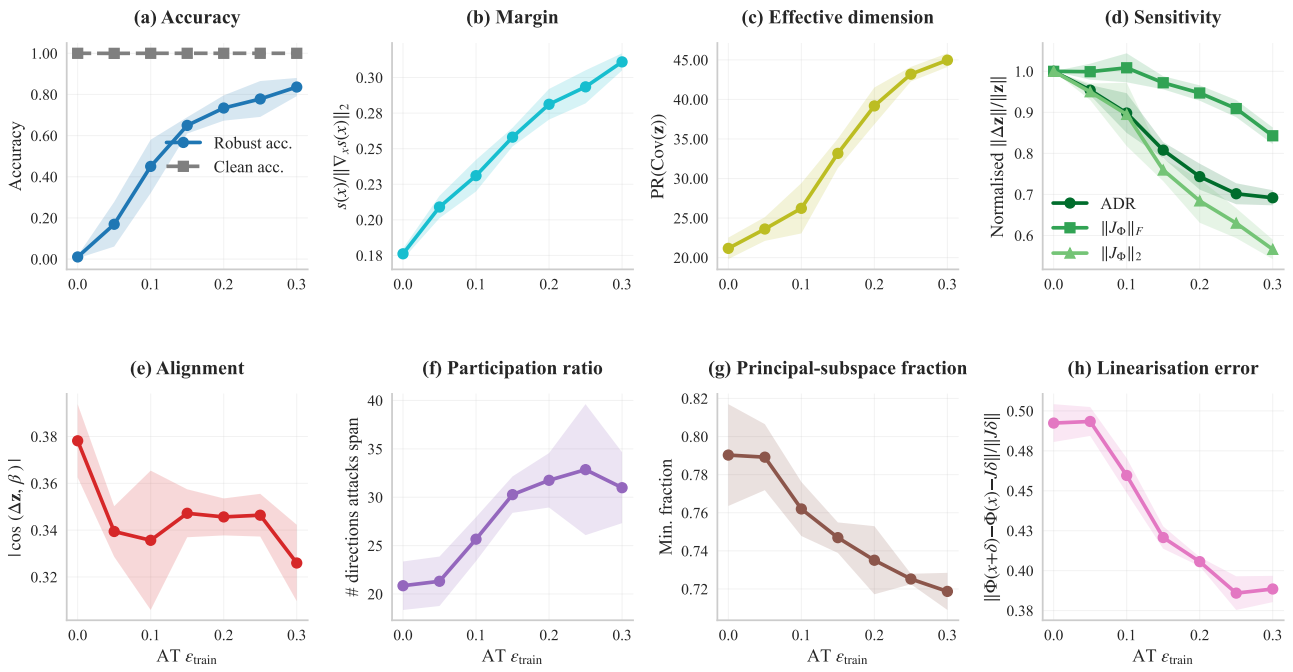

Figure 4. Adversarial robustness taxonomy (P=53, d=32, n=256) under untargeted ℓ_2 PGD at $\varepsilon = 0.3$, plotted against the AT budget ε_{AT} (mean across seeds, bands ± 1 s.d.). Under adversarial training, robust accuracy, geometric margin and effective representation dimension all rise, while encoder sensitivity and linearisation error fall, consistent with the algorithm being redistributed across more, less individually-sensitive frequencies.

On- vs. off-manifold adversarial examples. Stutz et al. (2019) distinguish on-manifold adversarial examples, which remain within the data manifold and correspond to meaningful input changes, from off-manifold examples, which exploit directions away from the manifold; they find that AT primarily defends against the off-manifold component. More recent work finds that on-manifold adversarial examples can have higher attack success rates than off-manifold ones (Xiao et al., 2025). We provide a concrete, interpretable instantiation of these questions.

7. Discussion, limitations & future work

To our knowledge, this is the first end-to-end mechanistic account of adversarial vulnerability in terms of the learned features implementing a reverse-engineered circuit. We find that adversarial perturbations are not arbitrary high-dimensional directions, but preferentially act on the Fourier features that implement the clean computation: successful PGD perturbations concentrate on the model's key frequencies, frequency-restricted attacks are competitive with full input-space PGD, and transfer is strongest between models sharing frequency support. Robustness is associated with broader frequency support, increased effective rank, and less concentrated attack displacements; AT increases robustness by reshaping the representation, but leaves the same frequency-based failure mode in place, suggesting a defence that is broad rather than mechanism-targeted.

Limitations A limitation is the narrowness of the setting. Whilst modular addition makes it possible to study adversarial vulnerability with an explicitly known task, data manifold, and learned circuit, our results should be read as a controlled case study rather than as evidence that the same diagnostics apply unchanged to models and tasks more generally. Our theoretical analysis focuses on a first-order phase response, and does not fully account for non-uniform phase shifts, normal-direction perturbations, or nonlinear effects.

Future Work One immediate direction is to extend the adversarial analysis to other architectures that have been studied (Moisescu-Pareja et al., 2025). Another is to design mechanism-targeted defences, such as frequency-domain adversarial training. Such defences would test whether mechanistic knowledge can guide robustness research rather than only explain attacks post hoc. More broadly, testing whether attacks exploit related Fourier structure identified in large models (Feucht et al., 2026; Lindsey et al., 2025) would clarify whether this account extends beyond shallow algorithmic models.

Ethics Statement

Due to its investigative nature and generality, our mechanistic interpretability framework is not anticipated to have any specific negative impact or immediate societal consequences. Our experiments utilise only toy models, involving no human subjects or personally identifiable information. The gradient-based attack methods used in the paper are already well-established in the literature. Our work does not introduce novel attack capabilities but rather provides theoretical understanding of existing phenomena. We have no conflicts of interest to declare. By elucidating the relationship between interpretability and adversarial vulnerability, we hope to inform future research directions on robustness, ultimately contributing to more trustworthy AI systems that can be safely deployed in critical applications.

Reproducibility Statement

We make source code for all experiments available at https://github.com/Stevinson/modulo-attack. Implementation details are referenced by experiment: architecture and training hyperparameters in Section 4.1; key-frequency identification and validation in Section 4.3, Section C.1, Section C.2, and Section C.3; attack methodology (PGD, mechanism-informed cosine injection, analytical phase attack, transfer protocol) and adversarial training schedule in Section 4.2, with logit-weight estimation in Section C.4 and the end-to-end attack flow in Appendix A; representation-geometry taxonomy definitions in Section C.5; and the phase-shift derivation in Section B.1. Experiments were conducted on an Apple M4 Max (16-core CPU, 40-core GPU, 128 GB unified memory) using PyTorch's MPS backend.

LLM Usage Statement

We used LLMs as assistive tools for generating experiment code, searching for related work, and proof-reading the manuscript. Generated code was verified through unit tests, manual review, and replication of results across multiple experimental setups. Suggested references were checked against the original publications. Editorial suggestions were reviewed by the authors before incorporation. All scientific claims, experimental design, and final text are the authors' own, and we take full responsibility for the content.

References

Melih Barsbey, Antônio H Ribeiro, Umut Şimşekli, and Tolga Birdal. On the Interaction of Compressibility and Adversarial Robustness. In *International Conference on Learning Representations, ICLR*, 2026.

Leonard Bereska, Zoe Tzifa-Kratira, Reza Samavi, and Efstratios Gavves. Superposition as Lossy Compression: Measure with Sparse Autoencoders and Connect to Adversarial Vulnerability. 2025.

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.

Bilal Chughtai, Lawrence Chan, and Neel Nanda. A Toy Model of Universality: Reverse Engineering How Networks Learn Group Operations. In *International Conference on Machine Learning*, pages 6243–6267. PMLR, 2023.

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.

Sheridan Feucht, Tal Haklay, Usha Bhalla, Daniel Wurgaft, Can Rager, Raphaël Sarfati, Jack Merullo, Thomas McGrath, Owen Lewis, Ekdeep Singh Lubana, Thomas Fel, and Atticus Geiger. Arithmetic in the Wild: Llama uses Base-10 Addition to Reason About Cyclic Concepts, 2026.

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.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR*, 2015.

Andrey Gromov. Grokking Modular Arithmetic. *arXiv preprint arXiv:2301.02679*, 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.pub/2025/attribution-graphs/biology.html, 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.

Gavin McCracken, Gabriela Moisescu-Pareja, Vincent Letourneau, Doina Precup, and Jonathan Love. Uncovering a Universal Abstract Algorithm for Modular Addition in Neural Networks. *arXiv preprint arXiv:2505.18266*, 2025.

Gabriela Moisescu-Pareja, Gavin McCracken, Harley Wiltzer, Vincent Létourneau, Colin Daniels, and Jonathan Love. On the Geometry and Topology of Neural Circuits for Modular Addition. In *NeurIPS 2025 Workshop on Mechanistic Interpretability*, 2025.

Gabriela Moisescu-Pareja, Gavin McCracken, Harley Wiltzer, Vincent Létourneau, Colin Daniels, Doina Precup, and Jonathan Love. On the Geometry and Topology of Representations: The Manifolds of Modular Addition. In *International Conference on Learning Representations (ICLR)*, 2026.

Depen Morwani, Benjamin L. Edelman, Costin-Andrei Oncescu, Rosie Zhao, and Sham M. Kakade. Feature Emergence via Margin Maximization: Case Studies in Algebraic Tasks. In *The Twelfth International Conference on Learning Representations*, 2024.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress Measures for Grokking via Mechanistic Interpretability. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, 2023.

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization Beyond Overfitting on Small Algorithmic Datasets, 2022.

Lucas Prieto, Edward Stevinson, Melih Barsbey, Tolga Birdal, and Pedro A. M. Mediano. From Data Statistics to Feature Geometry: How Correlations Shape Superposition, 2026.

Adam Scherlis, Kshitij Sachan, Adam S. Jermyn, Joe Benton, and Buck Shlegeris. Polysemanticity and Capacity in Neural Networks, 2025.

David Stutz, Matthias Hein, and Bernt Schiele. Disentangling Adversarial Robustness and Generalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6976–6987, 2019.

C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, I. Goodfellow, and R. Fergus. Intriguing Properties of Neural Networks. In *International Conference on Learning Representations*, 2014.

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.

Jiancong Xiao, Liusha Yang, Yanbo Fan, Jue Wang, and Zhi-Quan Luo. Understanding Adversarial Robustness against On-Manifold Adversarial Examples. *Pattern Recognition*, 159:111071, 2025.

A. End-to-end Attack Flow

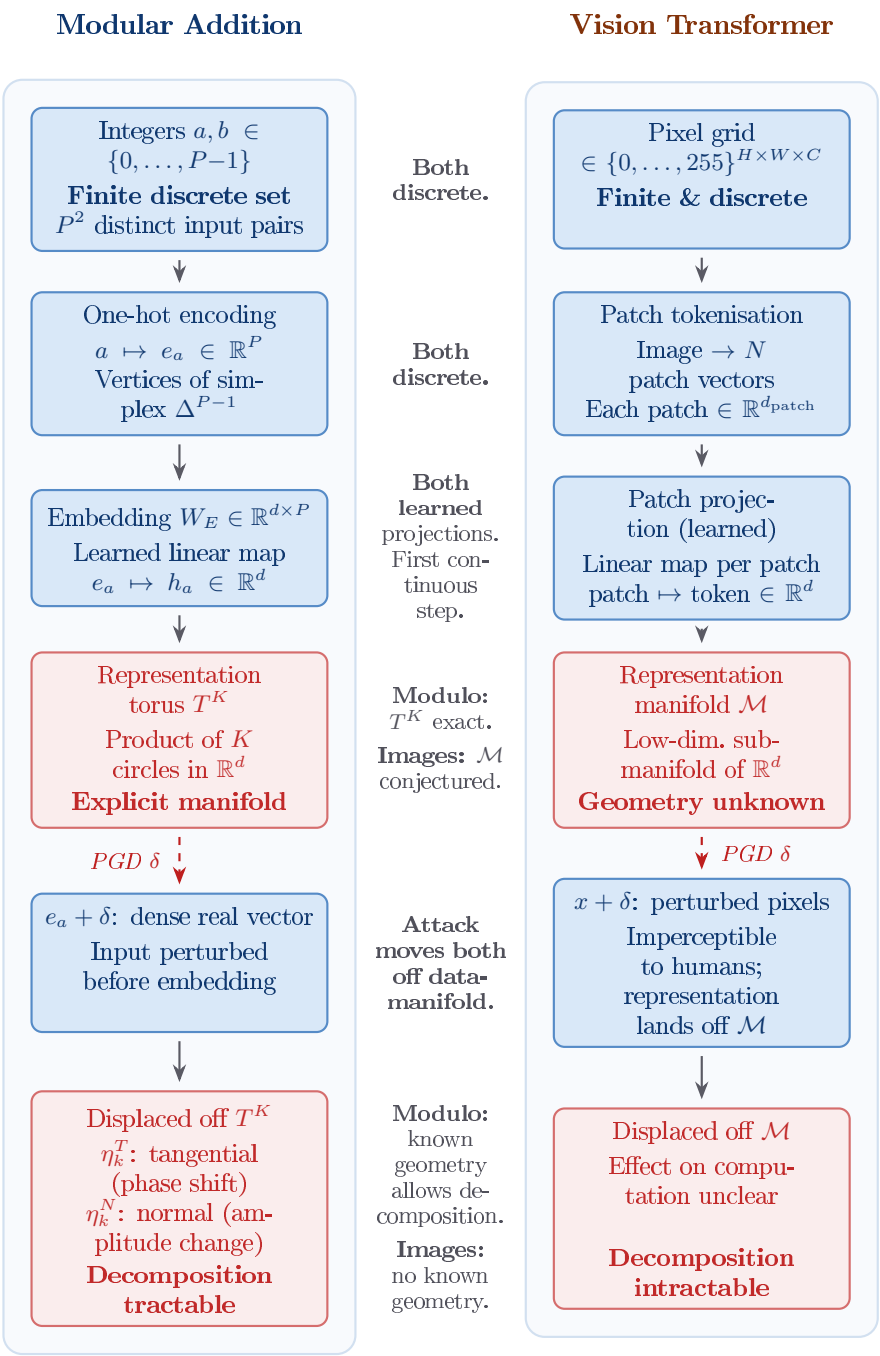

Figure 5. PGD attacks in modular addition and vision both perturb inputs off the data manifold before mapping them into representation space. In modular addition, the data and representation manifolds are explicit, allowing the off-manifold displacement to be analysed mechanistically.

This section summarises the representation-to-representation flow of an attack on the MLP-Add architecture. The clean computation can be read as

$$(\boldsymbol{e}_a, \boldsymbol{e}_b) \xrightarrow{W_E} (\boldsymbol{h}_a, \boldsymbol{h}_b) \xrightarrow{+} \boldsymbol{s}_{ab} \xrightarrow{W_1, \mathrm{ReLU}} \boldsymbol{r}_{ab} \xrightarrow{W_{\text{out}}} \boldsymbol{z}_{ab}, \tag{14}$$

where the embedding representations lie on the learned key-frequency torus, the forced summation maps the pair representation into a disc-like post-embedding representation, the hidden layer rectifies projections of this disc point, and the output layer combines the resulting features into Fourier-structured logits.

1. Input simplex to embedding torus. A clean token a is represented by the one-hot vertex $\boldsymbol{e}_a$ and embedded as $\boldsymbol{h}_a = W_E \boldsymbol{e}_a$. In the active frequency planes this is approximately

$$V_k^\top \boldsymbol{h}_a \approx A_k \begin{pmatrix} \cos(2\pi k a/P) \\ \sin(2\pi k a/P) \end{pmatrix}, \qquad k \in \mathcal{K}. \tag{15}$$

Thus the clean representation is a point on the product of active-frequency circles. The input perturbation changes the one-hot vector to $\boldsymbol{e}_a + \boldsymbol{\delta}$, generally leaving the input simplex. After the embedding, the perturbation becomes

$$\boldsymbol{h}'_a = W_E(\boldsymbol{e}_a + \boldsymbol{\delta}) = \boldsymbol{h}_a + \boldsymbol{\eta}, \qquad \boldsymbol{\eta} = W_E \boldsymbol{\delta}. \tag{16}$$

2. Embedding displacement to feature displacement. Only the component of $\boldsymbol{\eta}$ that lands in the active sine-cosine planes directly manipulates the learned trigonometric features. For each active frequency,

$$\tilde{\boldsymbol{\eta}}_k = V_k^\top \boldsymbol{\eta} = V_k^\top W_E \boldsymbol{\delta}. \tag{17}$$

At the clean feature angle $\theta_k = 2\pi k a/P$, this displacement decomposes into a tangential component, which rotates the represented feature value, and a normal component, which changes its radius:

$$\tilde{\boldsymbol{\eta}}_k = \eta_k^T \boldsymbol{t}_k + \eta_k^N \boldsymbol{n}_k, \qquad \varphi_k \approx \frac{\eta_k^T}{A_k} = \frac{\boldsymbol{t}_k^\top V_k^\top W_E \boldsymbol{\delta}}{A_k}. \tag{18}$$

The phase sensitivity vector $\boldsymbol{g}_k(a) = A_k^{-1} W_E^\top V_k \boldsymbol{t}_k$ therefore maps input perturbations to feature rotations, $\varphi_k \approx \boldsymbol{g}_k(a)^\top \boldsymbol{\delta}$.

3. Perturbed embedding torus to perturbed sum disc. With b unperturbed, the summed representation becomes

$$\boldsymbol{s}'_{ab} = \boldsymbol{h}'_a + \boldsymbol{h}_b = \boldsymbol{s}_{ab} + \boldsymbol{\eta}. \tag{19}$$

In the single-frequency picture, summing two circle points produces a disc point whose direction is the half-answer angle and whose radius depends on $|a - b|$. A tangential attack on a rotates the a clock hand by φ, so the disc direction shifts by approximately $\varphi/2$ and the answer angle read by the circuit shifts by approximately φ. In the multi-frequency case the same statement holds frequency-by-frequency: the attack induces the phase vector $\boldsymbol{\varphi} \approx G(a)^\top \boldsymbol{\delta}$.

4. Perturbed sum disc to hidden activations. The first hidden layer projects the summed representation onto learned preferred directions and applies ReLU:

$$\boldsymbol{r}'_{ab} = \mathrm{ReLU}(W_1 \boldsymbol{s}'_{ab} + \boldsymbol{b}_1)\,. \tag{20}$$

In the idealised single-frequency construction, these pre-activations are a cosine over preferred direction, and ReLU clips the negative half. Opposite or phase-shifted groups of hidden neurons then provide signed readouts from which the output layer can reconstruct the answer-angle features. The attack changes which preferred-direction neurons fire and changes their amplitudes, thereby corrupting the hidden representation used for the final Fourier readout.

5. Hidden representation to corrupted logits. The output layer converts the hidden representation into logits that are well approximated by active-frequency cosine votes:

$$z_c \approx \beta + \sum_{k \in \mathcal{K}} w_k \cos\left(\frac{2\pi k(a + b - c)}{P}\right). \tag{21}$$

After the attack, the phase-shifted surrogate is

$$z'_c \approx \beta + \sum_{k \in \mathcal{K}} w_k \cos\left(\frac{2\pi k(a + b - c)}{P} + \varphi_k\right). \tag{22}$$

The clean class $c^* = (a + b) \bmod P$ wins because all active-frequency votes constructively interfere there. The attack succeeds when the induced phase and amplitude errors reduce this constructive interference enough that some competing class c' has logit at least as large as c^*.

Summary. The attack is therefore not an arbitrary movement in input space. Its efficient pathway is

$$\boldsymbol{\delta} \;\mapsto\; W_E\boldsymbol{\delta} \;\mapsto\; \{\tilde{\boldsymbol{\eta}}_k\}_{k\in\mathcal{K}} \;\mapsto\; \boldsymbol{\varphi} \;\mapsto\; \boldsymbol{s}'_{ab} \;\mapsto\; \boldsymbol{r}'_{ab} \;\mapsto\; \boldsymbol{z}'_{ab}, \tag{23}$$

where the key mechanistic step is the conversion of a small dense input perturbation into rotations and radius changes of the learned Fourier representations. This is why mechanism-informed attacks restrict $\boldsymbol{\delta}$ to the active input DFT span: those directions are the input-space directions that most directly steer the representations used by the learned algorithm.

B. Proof Details

B.1. Derivation of the Phase Shift

We provide a more detailed derivation of the first-order phase shift from proposition 1.

Let the clean embedding at frequency k be $\boldsymbol{p}_k = A_k(\cos\theta_k, \sin\theta_k)^\top$ and the perturbation be $\tilde{\boldsymbol{\eta}}_k = (\eta_1, \eta_2)^\top$. The perturbed point is $\boldsymbol{p}'_k = \boldsymbol{p}_k + \tilde{\boldsymbol{\eta}}_k = (A_k\cos\theta_k + \eta_1, A_k\sin\theta_k + \eta_2)^\top$. The perturbed angle is

$$\theta'_k = \arctan\left(\frac{A_k\sin\theta_k + \eta_2}{A_k\cos\theta_k + \eta_1}\right). \tag{24}$$

Using the identity $\arctan(y/x) = \theta_k + \arctan\left(\frac{y\cos\theta_k - x\sin\theta_k}{x\cos\theta_k + y\sin\theta_k}\right)$ with $x = A_k\cos\theta_k + \eta_1$ and $y = A_k\sin\theta_k + \eta_2$:

$$\begin{aligned} y\cos\theta_k - x\sin\theta_k &= (A_k\sin\theta_k + \eta_2)\cos\theta_k - (A_k\cos\theta_k + \eta_1)\sin\theta_k \\ &= \eta_2\cos\theta_k - \eta_1\sin\theta_k = \boldsymbol{t}_k^\top\tilde{\boldsymbol{\eta}}_k, \end{aligned} \tag{25}$$

$$\begin{aligned} x\cos\theta_k + y\sin\theta_k &= (A_k\cos\theta_k + \eta_1)\cos\theta_k + (A_k\sin\theta_k + \eta_2)\sin\theta_k \\ &= A_k + \eta_1\cos\theta_k + \eta_2\sin\theta_k = A_k + \boldsymbol{n}_k^\top\tilde{\boldsymbol{\eta}}_k. \end{aligned} \tag{26}$$

Therefore,

$$\theta'_k - \theta_k = \arctan\left(\frac{\boldsymbol{t}_k^\top\tilde{\boldsymbol{\eta}}_k}{A_k + \boldsymbol{n}_k^\top\tilde{\boldsymbol{\eta}}_k}\right). \tag{27}$$

For $\|\tilde{\boldsymbol{\eta}}_k\| \ll A_k$, the first-order expansion gives

$$\varphi_k = \theta'_k - \theta_k \approx \frac{\boldsymbol{t}_k^\top\tilde{\boldsymbol{\eta}}_k}{A_k}, \tag{28}$$

which is the result stated in proposition 1.

The *exact* (non-linearised) phase shift is

$$\varphi_k^{\text{exact}} = \arctan\left(\frac{\boldsymbol{t}_k^\top\tilde{\boldsymbol{\eta}}_k}{A_k + \boldsymbol{n}_k^\top\tilde{\boldsymbol{\eta}}_k}\right). \tag{29}$$

The experiments use the linearised phase map $\boldsymbol{\varphi} \approx G(a)^\top\boldsymbol{\delta}$, not this exact nonlinear expression.

C. Additional Experimental Details

C.1. Key Frequency Identification Algorithm

We identify key frequencies from the embedding matrix W_E as follows:

1. Compute $F = \mathrm{DFT}(W_E) \in \mathbb{C}^{d\times P}$, where the DFT is applied along the column index (over integers $0, \ldots, P-1$).

2. For each frequency $k \in \{0, \ldots, \lfloor P/2 \rfloor\}$, compute the energy $E_k = \|F_{:,k}\|_2^2$.

3. Normalise: $\hat{E}_k = E_k / \sum_j E_j$.

4. Identify key frequencies as those with $\hat{E}_k > \tau$, where τ is set adaptively as $2/P$ (twice the uniform baseline).

This procedure is validated against per-neuron DFT analysis of pre-activation matrices, following McCracken et al. (2025). We find consistent key frequency identification across both methods.

C.2. Linear probe validation of Fourier coordinates

Following the use of linear probes to validate coordinate features in Prieto et al. (2026), we test whether the expected Fourier coordinates are linearly represented in the embedding table. After training the modular-addition model, we freeze W_E and collect embeddings $\boldsymbol{h}_a = W_E \boldsymbol{e}_a$ for $a = 0, \ldots, P-1$. For each candidate frequency k, we train a linear probe $P_k \in \mathbb{R}^{2\times d}$ to predict

$$\boldsymbol{y}_k(a) = \begin{pmatrix} \cos(2\pi ka/P) \\ \sin(2\pi ka/P) \end{pmatrix}, \qquad \hat{\boldsymbol{y}}_k(a) = P_k \boldsymbol{h}_a. \tag{30}$$

The probe is trained on a subset of integers and evaluated on held-out integers. We report the validation coefficient of determination

$$R_k^2 = 1 - \frac{\sum_{a\in\mathcal{V}} \|\boldsymbol{y}_k(a) - \hat{\boldsymbol{y}}_k(a)\|_2^2}{\sum_{a\in\mathcal{V}} \|\boldsymbol{y}_k(a) - \bar{\boldsymbol{y}}_k\|_2^2}, \qquad \bar{\boldsymbol{y}}_k = \frac{1}{|\mathcal{V}|} \sum_{a\in\mathcal{V}} \boldsymbol{y}_k(a), \tag{31}$$

where $\mathcal{V}$ is the validation set. High R_k^2 for $k \in \mathcal{K}$ confirms that the identified key frequencies are linearly readable from the learned embeddings, while low R_k^2 for non-key frequencies provides a control showing that the probes are not simply recovering arbitrary Fourier coordinates.

C.3. Subspace ablation validation

As a functional validation that the identified Fourier features are used by the model, we ablate either the key-frequency subspace or its orthogonal complement in the embedding representation. Let

$$\mathcal{V}_\mathcal{K} = \mathrm{span}\{\boldsymbol{v}_{k,c}, \boldsymbol{v}_{k,s} : k \in \mathcal{K}\}, \tag{32}$$

and let $\Pi_\mathcal{K}$ be the orthogonal projection onto this subspace. For each clean embedding $\boldsymbol{h}_a$, we compare the model's accuracy after replacing the embedding by either

$$\boldsymbol{h}_{a,\mathrm{keep}} = \Pi_\mathcal{K} \boldsymbol{h}_a + \boldsymbol{\mu}_\perp, \qquad \boldsymbol{h}_{a,\mathrm{remove}} = (I - \Pi_\mathcal{K}) \boldsymbol{h}_a + \boldsymbol{\mu}_\parallel, \tag{33}$$

where $\boldsymbol{\mu}_\perp$ and $\boldsymbol{\mu}_\parallel$ are the dataset means of the ablated orthogonal-complement and key-frequency components, respectively. Keeping only the key-frequency subspace preserves most of the model's test accuracy, while removing that subspace collapses performance, replicating the modular-addition ablation logic of Nanda et al. (2023).

C.4. Logit-weight estimation

To estimate the surrogate logit weights $\{w_k\}$, we evaluate the model on all inputs $(a, 0)$ and align each logit vector by the offset from the correct answer $c^* = a$. That is, for offset $\delta = c^* - c \pmod P$, we average the aligned profile

$$\bar{z}_\delta = \frac{1}{P} \sum_{a=0}^{P-1} z_{a,(a-\delta) \bmod P}. \tag{34}$$

Under the phase-only surrogate

$$z_\delta \approx \beta + \sum_{k\in\mathcal{K}} w_k \cos\left(\frac{2\pi k\delta}{P}\right), \tag{35}$$

so the Fourier magnitude of $\bar{z}$ at frequency k provides a heuristic estimate of w_k.

C.5. Representation-geometry taxonomy metrics

This section defines the generic robustness-taxonomy metrics introduced in Section 4.3; these metrics are motivated by representation-geometry views of adversarial robustness (Barsbey et al., 2026). For the MLP-Add architecture, let Φ denote the map from the input pair through the shared embedding, embedding summation, and hidden ReLU layer to the representation used by the final linear readout. Let $z = \Phi(x)$ be the clean representation, $J_\Phi(x) = \partial\Phi/\partial x$ the corresponding representation Jacobian, and $x_{\mathrm{adv}} = x + \delta$ a PGD adversarial example. Let $\Delta z = \Phi(x_{\mathrm{adv}}) - \Phi(x)$ be the adversarial representation displacement. The taxonomy combines these input-to-representation quantities with full-logit quantities such as robust accuracy, margins, and decision-gradient alignment. For a correctly classified input with label y, define the confidence margin

$$m_{\mathrm{conf}}(x) = f_y(x) - \max_{c \neq y} f_c(x), \tag{36}$$

and let $\beta(x) = \nabla_z m_{\mathrm{conf}}(x)$ be the representation-space decision gradient.

Robust accuracy. RA is the fraction of evaluation examples that remain correctly classified after PGD attack at the chosen perturbation budget. It is the direct robustness outcome that the other metrics aim to explain.

Confidence margin. m_{conf} is the clean logit gap between the correct class and the strongest competitor. It captures how confidently the model classifies the input, but by itself does not account for how quickly perturbations can erode that gap.

Geometric margin. m_{geom} is the local linearised distance to the decision boundary, computed as the confidence margin divided by the input-space norm of its gradient,

$$m_{\mathrm{geom}}(x) = \frac{m_{\mathrm{conf}}(x)}{\|\nabla_x m_{\mathrm{conf}}(x)\|_2}. \tag{37}$$

It captures whether a given logit gap corresponds to a genuinely large local perturbation budget or merely to rescaled logits.

Representation sensitivity. $\|J_\Phi\|_F$ and $\|J_\Phi\|_2$ are the Frobenius and spectral norms of the representation Jacobian. They measure average and worst-case local amplification from input space to representation space; lower values indicate that small input perturbations produce smaller changes in the network representation.

Adversarial displacement ratio. ADR measures the representation displacement realised by the actual attack, normalised by the clean representation scale:

$$\mathrm{ADR}(x) = \frac{\|\Delta z\|_2}{\|z\|_2}. \tag{38}$$

It captures how much representational change the successful perturbation achieves, as opposed to the worst-case gain measured by the Jacobian norms.

Decision-gradient alignment. $|\cos|$ is the absolute cosine similarity between Δz and the representation-space decision gradient $\beta(x)$,

$$|\cos|(x) = \frac{|\Delta z^\top \beta(x)|}{\|\Delta z\|_2 \, \|\beta(x)\|_2}. \tag{39}$$

High alignment means the attack displacement is directed efficiently toward reducing the decision margin; low alignment means much of the displacement is spent in directions less relevant to the current classifier.

Attack concentration. top-3 is the fraction of adversarial-displacement covariance captured by its three largest eigenvalues,

$$\text{top-3} = \frac{\lambda_1 + \lambda_2 + \lambda_3}{\sum_j \lambda_j}, \qquad \{\lambda_j\} = \mathrm{eig}(\mathrm{Cov}(\Delta z)). \tag{40}$$

It captures whether attacks exploit a small number of dominant representation directions or are distributed across many directions.

Principal-subspace fraction. $f_{\|}$ measures the fraction of Δz lying in the principal subspace of clean representations. If P_{clean} projects onto the top clean-activation PCA subspace, then

$$f_{\|}(x) = \frac{\|P_{\text{clean}}\Delta z\|_2^2}{\|\Delta z\|_2^2}. \tag{41}$$

It is intended to distinguish attacks that move along directions of natural representation variation from attacks that exploit directions outside the principal clean-representation subspace.

Effective representation dimension. $\mathrm{PR}(\mathrm{Cov}(z))$ is the participation ratio of the clean-representation covariance,

$$\mathrm{PR}(\mathrm{Cov}(z)) = \frac{\left(\sum_j \lambda_j\right)^2}{\sum_j \lambda_j^2}, \qquad \{\lambda_j\} = \mathrm{eig}(\mathrm{Cov}(z)). \tag{42}$$

It captures the effective number of dimensions the encoder uses to represent natural data; under our framework it should rise under AT as the algorithm recruits additional key frequencies.

Linearisation error. lin. err. compares the actual adversarial representation displacement with the first-order Jacobian prediction:

$$\mathrm{linerr}(x) = \frac{\|\Phi(x+\delta) - \Phi(x) - J_\Phi(x)\delta\|_2}{\|J_\Phi(x)\delta\|_2}. \tag{43}$$

Low values indicate that the attack is well described by local linear sensitivity, while high values indicate that nonlinear effects such as ReLU gating changes or curvature are important.

D. Embedding Fourier and singular spectra under adversarial training

Figure 3 reports per-row summary statistics for each AT budget. Figure 6 appendix gives the underlying spectra so the reader can see the shape, not just the summary.

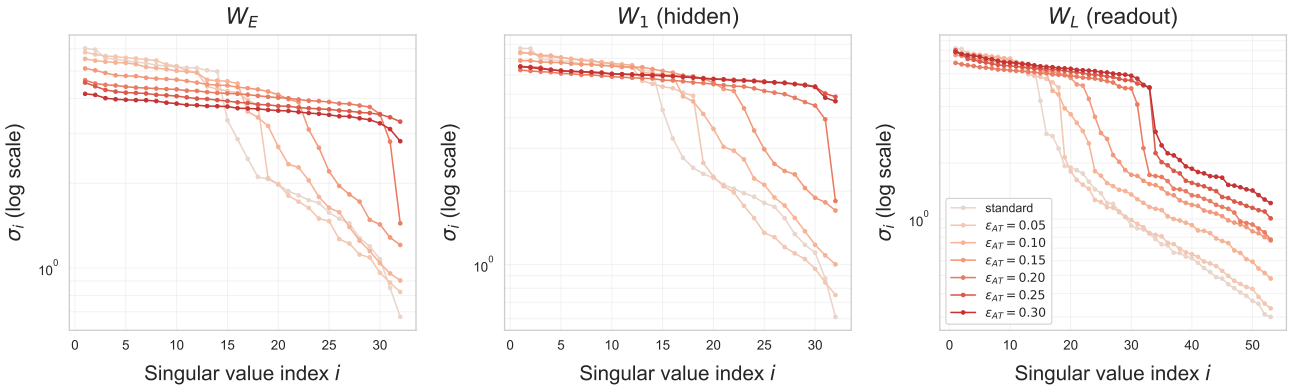

Figure 6. Singular value spectra of the three weight matrices (W_E, W_1, W_L) under adversarial training. Each curve is one AT-trained model on a log-y axis; warmer colours correspond to larger ε_{AT}, matching the heatmap palette in Figure 3. Reading across panels: AT visibly extends the singular tail of W_E (left) without much change to its dominant directions, consistent with the embedding recruiting more frequencies rather than redistributing weight.