The Canonical Representation of a Task

Anonymous authors
Paper under double-blind review

Abstract

Generalization in deep learning remains poorly understood, as neural networks fall outside the framework of classical statistical learning theory. To make progress on understanding generalization, research has focused on controlled tasks such as modular arithmetic, as a testbed. On these tasks, models exhibit grokking, i.e., a delayed onset of generalization after training loss has converged. Prior work has identified empirical regularities in the learned representations associated with this transition, but the mapping between representation structure and generalization behavior remains empirical and descriptive. We lack a predictive theory of why and when generalization occurs. In this work, we provide such a predictive theory for modular arithmetic tasks including addition, subtraction, multiplication, and division. We introduce the notion of *canonical representation* of a task: the representation determined by the target function prior to training which is needed for perfect generalization. For modular arithmetic, the canonical representation can be derived from the group structure of the task. We then define *representational deviation* as the discrepancy between the learned representation and the canonical representation which meets a specified target loss. From this, we derive that reaching a prescribed level of generalization requires the representational deviation to fall below a threshold. We finally provide a set of reproducible experiments which empirically confirm the above findings and offer a regularizer to accelerate the grokking transition.

1 Introduction

Generalization has been a central question in deep learning theory. Why and how over-parametrized neural networks generalize remains unexplained within the framework of traditional statistical learning theory (Vapnik, 2013; Mohri et al., 2018; Hu et al., 2024). Neural networks learn representations jointly with predictors, a setting that falls outside the fixed-representation and fixed-hypothesis-class assumptions of statistical learning theory. This gap has significant implications, both for theoretical understanding and for the deployment of deep learning models in increasingly complex settings. Although several approximations have been proposed in restricted regimes, we still lack a theory of generalization usable in real world settings (Jacot et al., 2020; Tishby & Zaslavsky, 2015; Watanabe, 2009).

To make progress on this question, research has focused on small and controlled tasks, where generalization can be directly observed and tested. A prominent example is grokking (Power et al., 2022), in which instances of the same model trained on the same task achieve similar training loss yet generalize at different rates or not at all. Due to this discrepancy, grokking is thereby used as a testbed for theories of generalization in deep learning.

Recent work on grokking has uncovered patterns and regularities underlying these dynamics (Nanda et al., 2023a; Liu et al., 2022a; Kumar et al., 2023; Ruppik et al., 2025). In particular, several studies emphasize the role of the learned representation space, showing that shifts in representation geometry are closely associated with the onset of generalization. These include the emergence of computational circuits during training (Nanda et al., 2023a), phase-transition–like behavior in representation space (Liu et al., 2022a; Žunkovič & Ilievski, 2024), and convergence toward low-dimensionality representations (Ruppik et al., 2025). However, while these works identify correlates of grokking in the learned representations, the results remain empirical

and descriptive. To tackle the problem of generalization in this restricted case, a predictive theory would explain these observations from first principles and explain why and when generalization emerges.

In this paper, we develop such a theory for modular arithmetic tasks, one of the canonical settings exhibiting grokking. We introduce the notion of the *canonical representation* of a task: the intrinsic representation structure induced by the target function itself. To each canonical representation is associated a *sufficient representation*, corresponding to the minimal representational structure required to achieve a target loss. We argue that generalization emerges when the learned representation converges toward this sufficient representation.

Our contributions are as follows. (i) We formalize the *canonical representation* of a task and its associated sufficient subspace, determined directly by the target function prior to training. (ii) We define *representational deviation* as the residual representation energy outside the sufficient subspace, and show that generalization up to a chosen margin requires this deviation to fall below a critical threshold. (iii) From the geometry of the sufficient representation and its associated optimization dynamics, we derive predictive conditions for the grokking transition. (iv) We provide reproducible experiments validating the theoretical predictions and observed transition behavior, as well as a regularizer to accelerate this transition. This offers a first-principles account of grokking in modular arithmetic tasks.

2 The Canonical Representation of a Task

We begin by formalizing the concepts of task, model, and generalization. In our setup, we focus on deterministic classification tasks with a finite label set. We then introduce the notion of the *canonical representation* of a task and establish its connection to generalization. Finally, we derive the canonical representation explicitly for modular arithmetic.

2.1 Concepts

Task. The first step is to formalize the concept of task. Intuitively, a task specifies what must be learned independently of any particular model or implementation (Mitchell, 1997; Goodfellow, 2016). Following statistical learning theory, we formalize this idea by taking a task T to be the tuple $T = (X, Y, P, L)$ where X denotes the input space, Y is a finite label space, P is the unknown probability distribution over $X \times Y$, and L is a loss function (Vapnik, 2013). For a finite set Y, we write $\mathbb{R}^Y$ for the vector space of logits indexed by labels $y \in Y$. The goal of learning is to approximate f^* given the loss function and finite samples. In this work, we restrict attention to deterministic classification tasks, where the distribution P is induced by a target function $f^* : X \to Y$, such that $P(y \mid x) = \mathbf{1}[y = f^*(x)]$.

Generalization. Generalization is understood relative to T. For any measurable predictor $f : X \to Y$, the *population risk* $R(f)$ is the expected loss under P. It is a deterministic functional of (f, P, L), and is fully specified by these variables. However, the underlying distribution P is typically unkown. We thereby estimate population risk $R(f)$ with an estimator. The *empirical risk* $\hat{R}_S(f)$ is its sample counterpart given the test set S, and is a random variable whose value depends on the particular realization S. Generalization is thereby defined as the discrepancy $R(f) - \hat{R}_S(f)$. For deep neural networks, however, obtaining *non-vacuous* bounds on the generalization gap remains difficult (Valiant, 1984; Chatterjee & Zielinski, 2022). A model generalizes well when it achieves small excess risk compared to the Bayes optimal predictor (Shalev-Shwartz & Ben-David, 2014).

Model and Instantiation. A task T is not well-defined without restricting to a hypothesis class of admissible functions $\mathcal{H}$. A model defines this hypothesis class and selects the best approximation within it given a finite sample drawn from P. Formally, a model instantiates the task by returning the learned function $\hat{f} \in \mathcal{H}$. The generalization gap of $\hat{f}$ is then the discrepancy between its empirical risk and the population risk for the task.

2.2 Canonical Representation of a Task

A task T implicitly encodes structure through its target function $f^* : X \to Y$. Before any model is trained, f^* determines which distinctions in the input space are relevant to prediction: distinctions in X matter only insofar as they result in different labels in Y. For a classification task, this means f^* partitions X into equivalence classes of inputs sharing the same label. We can thereby define the relation $x \sim x' \iff f^*(x) = f^*(x')$. This relation defines an equivalence class of x denoted $[x] = \{x' \in X : f^*(x') = f^*(x)\}$. We call the quotient set $X/\sim$ the collection of all such equivalence classes on X. By construction, f^* factors through $X/\sim$: it decomposes into a projection that groups inputs into their equivalence classes, followed by an injective map from classes to labels.

Definition 1 (Canonical Decomposition)**.** *Let $T = (X, Y, P, L)$ be a task with target function $f^* : X \to Y$. The **canonical decomposition** of T is the factorization $f^* = \bar{f}^* \circ \pi$, where $\pi : X \to X/\sim$ is the canonical projection $x \mapsto [x]$, and $\bar{f}^* : X/\sim \to Y$ is the induced injective map on equivalence classes.*

This decomposition depends only on f^*, not on P or any model. It is the coarsest partition of X consistent with the correct labeling, and inherits whatever additional structure f^* possesses: if f^* is a group homomorphism, the equivalence classes carry the algebraic structure of the quotient group.

A deep learning model instantiates T by selecting a function $\hat{f} \in \mathcal{H}$. For classification tasks, the model first maps inputs to features through $\phi : X \to \mathbb{R}^d$, then applies a linear readout $W \in \mathbb{R}^{|Y| \times d}$ to produce logits $W\phi(x) \in \mathbb{R}^{|Y|}$. The output is obtained by a final decision map $\rho : \mathbb{R}^{|Y|} \to Y$, so that $\hat{f} = \rho \circ W \circ \phi$. Assuming $f^* \in \mathcal{H}$, we can identify the constraints under which $\hat{f}$ represents f^*.

Since f^* factors through the quotient $X/\sim$, any composition implementing it must also be constant on equivalence classes. This is a condition on $W \circ \phi$ jointly: since W is linear, $\phi(x) - \phi(x') \in \ker(W)$ for all $x \sim x'$, so ϕ already encodes the quotient $X/\sim$ up to a linear residual in $\ker(W)$. In the canonical case where this linear residual vanishes, ϕ factors as $\phi = \tilde{\phi} \circ \pi$ with $\tilde{\phi} : X/\sim \to \mathbb{R}^d$ injective, so that distinct equivalence classes, which carry distinct labels by definition of $\sim$, receive distinct coordinates in $\mathbb{R}^d$. These are requirements for the composition $W \circ \phi$ to represent f^*. Under these conditions, the learned predictor realizes the canonical decomposition. This allows us to identify the constraints on the representational space $\mathbb{R}^d$ needed to instantiate f^*.

Definition 2 (Canonical Representation)**.** *A **canonical representation** of a task T is a faithful embedding $\tilde{\phi} : X/\sim \to \mathbb{R}^d$ equipped with a readout W such that $\rho \circ W \circ \tilde{\phi} = \bar{f}^*$.*

Canonical representations therefore characterize the class of embeddings compatible with the quotient structure induced by the target function f^*. A model $W \circ \phi$ correctly represents the task insofar as its learned representation realizes a canonical representation. Furthermore, the decision map ρ is fixed and plays no role in the representation geometry analyzed below, we suppress it in what follows and focus on the logit map $W \circ \phi$ directly.

2.3 Sufficient Representation of a Task

The canonical representation characterizes the minimal geometry required to realize the target function f^*. However, realizing the quotient structure alone is not necessarily sufficient to minimize the loss function L. Different losses can impose additional geometric constraints on the representation space, such as margin separation for cross-entropy loss.

Definition 3 (Sufficient Representation)**.** *Let L be a loss and let $\delta > 0$. A **sufficient representation** for a task T is a canonical representation whose task loss under L is at most δ. It is **minimal** if no lower-dimensional canonical representation achieves task loss at most δ.*

The sufficient representations therefore characterize the set of canonical representations capable of achieving a loss below threshold δ. By construction, any logit map $W\tilde{\phi}(\pi(x))$ achieving this loss must realize a sufficient representation. In deterministic settings, the limit $\delta \to 0$ corresponds to convergence toward the minimal achievable population risk for the task.

The family of sufficient representations induces a nested hierarchy indexed by the loss threshold. If $\delta_1 < \delta_2$ then any representation sufficient for δ_1 is also sufficient for δ_2. Accordingly $\mathcal{V}_{\delta_1}^L \subseteq \mathcal{V}_{\delta_2}^L$, where $\mathcal{V}_\delta^L$ denotes the set of representations sufficient at level δ. Lower loss thresholds therefore impose increasingly restrictive constraints on the representation space.

A model instantiates the task through a representation $\phi : X \to \mathbb{R}^d$. In general, the learned geometry $\phi(X)$ does not exactly coincide with the sufficient representation. Since V_δ^L is a linear subspace, we can measure the discrepancy between the learned representation and the sufficient representation by orthogonal projection.

Definition 4 (Representational Deviation)**.** *Let $V \in \mathcal{V}_\delta^L$ be a chosen sufficient subspace for task T at loss level δ. The **representational deviation** of a sample x_i relative to V is $h_i^{(V)} = \Pi_{V^\perp}\phi(x_i)$, where $\Pi_{V^\perp}$ denotes the orthogonal projection onto the complement of V. For a readout W, the induced logit deviation is $H_i^{(V)} = Wh_i^{(V)}$.*

The magnitude $\|h_i^{(V)}\|^2$ measures the extent to which the learned representation deviates from the chosen sufficient subspace. A model whose representation lies entirely in V satisfies $h_i^{(V)} = 0$ for all i. Representational deviation induces a loss contribution relative to the sufficient component, denoted $d_L^{(V)}(H_i)$.

2.4 Sufficient Representations and Generalization for Cross-entropy Loss

In practice, deep learning models are trained with cross-entropy loss. We therefore specialize the notion of sufficient representation to the cross-entropy setting and derive the corresponding geometric constraints on the representation space.

For a task $T = (X, Y, P, L)$, a model produces logits $z(x) \in \mathbb{R}^Y$. We can define the margin against label $y \neq \bar{f}^*(x)$ as $\gamma_y(x) = z_{f^*(x)}(x) - z_y(x)$, the loss rewrites as $\mathcal{L}(x) = \log\left(1 + \sum_{y \neq f^*(x)} e^{-\gamma_y(x)}\right)$. The loss depends directly on the margin between the correct class logit and the incorrect class logits. The sufficient representation for a task with cross-entropy loss can therefore be characterized as a canonical representation with margins large enough to meet a loss threshold δ.

Proposition 1 (Sufficient Representation for Cross Entropy Loss)**.** *Let $(\tilde{\phi}, W)$ be a canonical representation of a deterministic classification task, with logits $z(x) = W\tilde{\phi}(\pi(x))$. Then $(\tilde{\phi}, W)$ is δ-sufficient for cross-entropy loss iff*

$$\mathbb{E}_{x \sim P_X} \log\left(1 + \sum_{y \neq f^*(x)} e^{-\gamma_y(x)}\right) \leq \delta.$$

This provides a link from the canonical representation of a task to generalization via the geometry of the representation space. Let $V \in \mathcal{V}_\delta^L$ be a chosen sufficient subspace. For each sample x_i, decompose the logits as $z_i = z_i^* + H_i^{(V)}$, where z_i^* is the sufficient component and $H_i^{(V)}$ is the logit deviation induced by the component of the representation outside V.

Proposition 2 (Deviation Loss Contribution)**.** *For cross-entropy loss, where $y_i^* = f^*(x_i)$, the per-sample loss contribution of the representational deviation is*

$$d_{\mathcal{L}}^{(V)}(H_i) = \log \sum_{y \in Y} e^{z_{iy}^* + H_{iy}^{(V)}} - \log \sum_{y \in Y} e^{z_{iy}^*} - H_{i,y_i^*}^{(V)},$$

Thus, once V is fixed, any excess loss relative to the sufficient representation is entirely determined by the deviation $H_i^{(V)}$. This gives a direct route from representational alignment to population risk: if the learned representation concentrates near V, then the deviation contribution is small.

Theorem 1 (Population Risk from Representational Deviation)**.** *Let $V \in \mathcal{V}_\delta^L$ be a chosen sufficient subspace, and decompose the logits as $z = z^* + H^{(V)}$. Then*

$$R(f) \leq \delta + \mathbb{E}_{x \sim P_X}\left[d_{\mathcal{L}}^{(V)}(H(x))\right].$$

This result identifies representational deviation as the quantity that remains after the task-aligned component has been fixed. The sufficient component z^* accounts for the loss achievable inside V, while $d_{\mathcal{L}}^{(V)}(H)$ measures the additional loss induced by off-subspace structure. The corresponding generalization gap is obtained by subtracting empirical risk from both sides of the exact decomposition $R(f) = R(z^*) + \mathbb{E}[d_{\mathcal{L}}^{(V)}(H)]$ so it depends on the sampling error of the sufficient component and of the deviation term.

3 Canonical Representations of Modular Arithmetic Tasks

A central challenge in machine learning is to learn the canonical representations. Since f^* is typically unknown, the partition $X/\sim$ must be inferred from finite samples drawn from P and cannot be specified before training. For certain structured tasks, however, f^* is known a priori, and the canonical decomposition can be derived exactly. In such cases, the canonical representation of the task can be specified before any model is trained and provide a target against which learned representations can be measured.

3.1 Modular Arithmetic Tasks

We study modular arithmetic tasks of the form $f^*(a, b)$ over $\mathbb{Z}_p$ for prime p, including modular addition $f^*(a, b) = (a + b) \bmod p$, subtraction $f^*(a, b) = (a - b) \bmod p$, multiplication $f^*(a, b) = (ab) \bmod p$, and division $f^*(a, b) = (a/b) \bmod p$. For addition and subtraction we take $X = \mathbb{Z}_p \times \mathbb{Z}_p$ and $Y = \mathbb{Z}_p$; for multiplication and division we take $X = \mathbb{Z}_p^\times \times \mathbb{Z}_p^\times$ and $Y = \mathbb{Z}_p^\times$. In all cases P is uniform over the corresponding domain and L is cross-entropy loss. These tasks form a family of deterministic problems widely studied in the grokking literature (Power et al., 2022; Nanda et al., 2023b; Kumar et al., 2023).

For each task, the target function $f^* : X \to Y$ induces an equivalence relation on X as per Definition 1. In modular arithmetic, this quotient structure is determined exactly by the underlying cyclic group operation defining the task. For addition and subtraction, the relevant structure is the additive cyclic group $(\mathbb{Z}_p, +)$, where the task is induced by additive group operations. For multiplication and division, the relevant structure is the multiplicative cyclic group $(\mathbb{Z}_p^\times, \cdot)$ over nonzero elements, where the task is induced by multiplicative group operations. The quotient classes are thereby fully determined by the finite cyclic group structure, allowing the canonical decomposition of the task to be derived exactly before training.

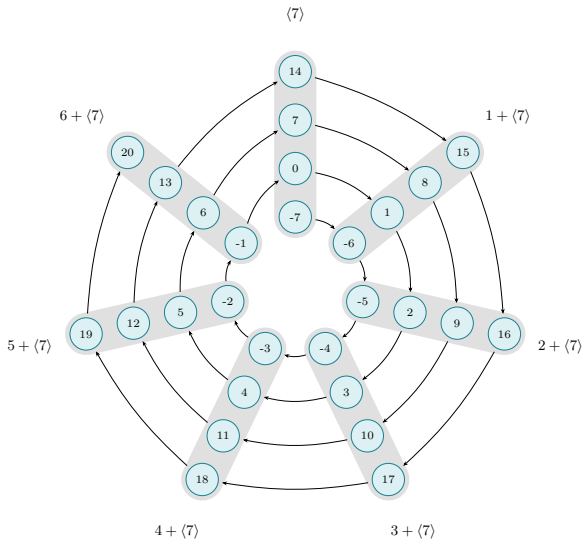
Proposition 3 (Canonical Decomposition of Modular Arithmetic Tasks)**.** *For modular arithmetic tasks, the target function f^* admits a canonical decomposition $f^* = \bar{f}^* \circ \pi$, where π is the minimal quotient map induced by the task's underlying cyclic group operation, and $\bar{f}^*$ identifies each quotient class with its corresponding label in Y.*

Figure 1 illustrates this mapping into equivalence classes induced by $\bar{f}^*$ for modular addition. The canonical decompositions for all other modular arithmetic tasks considered are provided in Appendix B.

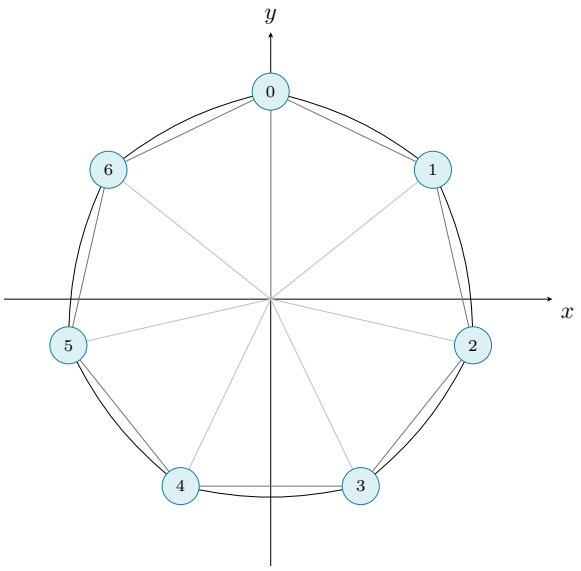
3.2 Canonical Representation of Modular Arithmetic

We now turn to how modular arithmetic tasks can be represented by a deep learning model. A model solving such a task instantiates a feature map $\phi : X \to \mathbb{R}^d$, before a linear readout. Under Definition 2, the central question is therefore how the quotient structure identified in Proposition 3 can be faithfully represented in $\mathbb{R}^d$.

Because the canonical quotient of modular arithmetic tasks is fully specified by a finite cyclic group structure, this problem is governed by representation theory (Fulton & Harris, 2013). In particular, a canonical representation corresponds to a group representation of the relevant quotient group as linear transformations on $\mathbb{R}^d$. For additive modular tasks this is a homomorphism $\rho : \mathbb{Z}_p \to GL(V)$; for multiplicative tasks, after choosing a generator of $\mathbb{Z}_p^\times$, it is a homomorphism $\rho : \mathbb{Z}_{p-1} \to GL(V)$, where $V \subseteq \mathbb{R}^d$. By Maschke's theorem, every finite-group representation over $\mathbb{R}$ decomposes uniquely into a direct sum of linear transformations called irreducible representations (irreps) (Fulton & Harris, 2013; Serre et al., 1977). Thus, once the canonical quotient structure of the task is known, its minimal realizable vector-space structure is given by the irreps of its cyclic group. The canonical representation of modular arithmetic tasks is therefore determined by the irreducible structure of their underlying quotient group.



(a) **Canonical quotient structure for modular addition with $p = 7$.** The integers are arranged into cosets of $\langle 7 \rangle \subset \mathbb{Z}$.



(b) **Canonical representation of the quotient $\mathbb{Z}_7$ in $\mathbb{R}^2$.** Under the non-trivial irreducible representation ρ_1, the generator of $\mathbb{Z}_7$ acts as rotation by $2\pi/7$.

Figure 1: **Canonical decomposition and representation of modular addition for $p = 7$.** Left: the quotient partition induced by modular equivalence, showing how $\mathbb{Z}$ decomposes into cosets of $\langle 7 \rangle$. Right: a canonical representation of the quotient group $\mathbb{Z}_7$ via its non-trivial irreducible representation in $\mathbb{R}^2$.

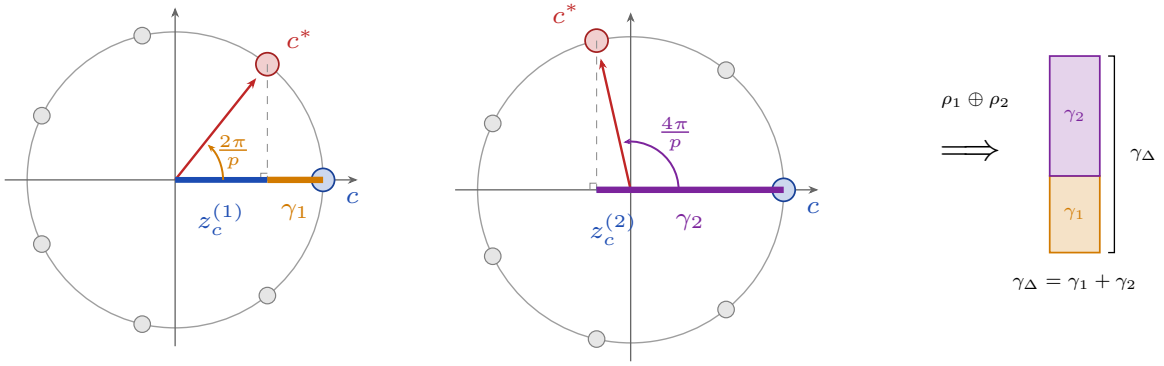
Proposition 4 (Canonical Representation of Modular Arithmetic Tasks)**.** *Canonical representations of modular arithmetic tasks are given by faithful non-trivial rotation irreps of the relevant cyclic group: $\mathbb{Z}_p$ for additive tasks, and $\mathbb{Z}_{p-1}$ for multiplicative tasks after choosing a generator of $\mathbb{Z}_p^\times$, along with the assocaited readout W. Direct sums of such irreps are also canonical.*

For $\mathbb{Z}_p$, the irreducible representations over $\mathbb{R}$ consist of one trivial representation and $\frac{p-1}{2}$ faithful non-trivial two-dimensional irreps, each embedding the quotient group as rotations on the unit circle. For cyclic groups, each non-trivial irrep corresponds to a rotational Fourier mode of frequency k. The frequency determines the periodic structure encoded by the representation and the amplitude determines the magnitude with which that mode contributes to the embedding and therefore to the resulting logits. Each irrep has a different frequency and amplitude. Since p is prime, every non-trivial irrep is a faithful representation of the group structure. A single non-trivial irrep suffices to recover f^* and provides a minimal canonical representation of the task. Therefore, any direct sum of non-trivial irreps yields an extended canonical representation: additional irreps increase the dimensionality of the embedding in $\mathbb{R}^d$ but do not change the quotient structure. For $\mathbb{Z}_{p-1}$, the faithful irreps are exactly the rotation blocks whose frequency k is coprime to $p - 1$. The canonical representations for all selected tasks are given in Appendix B.

3.3 Sufficient Representation of Modular Arithmetic

We now turn to the sufficient representation of our tasks. The loss function L in our setup is cross-entropy loss, which cares not only about correct classification but also about the margins between the correct and incorrect class logits.

Once the quotient structure is embedded through irreps, the task becomes a geometric classification problem in representation space. The readout computes a dot product between the representation and the class weight vectors, producing the class logits. This is illustrated in Figure 2a and Figure 2b, where the class logit is the projection of the correct-class representation onto each class direction.



(a) **Irrep ρ_1 ($k = 1$).** The class logit is the projection of c^* onto the class direction c; yielding margin contribution γ_1.

(b) **Irrep ρ_2 ($k = 2$),** which yields margin contribution γ_2.

(c) **Additive margin.** The direct sum $\rho_1 \oplus \rho_2$ yields additive margin contributions.

Figure 2: **Additive margin contributions from irreducible blocks for modular addition with $p = 7$.** In each irrep block k, the class logit is the horizontal projection $z_c^{(k)} = \alpha_k \cos(2\pi k\Delta/p)$. The colored segment is the gap to perfect alignment in that block, $\gamma_k = \alpha_k - z_c^{(k)} = \alpha_k(1 - \cos(2\pi k\Delta/n))$.

Let $n = |X/\!\sim| = |Y|$ denote the number of quotient classes; thus $n = p$ for additive tasks and $n = p - 1$ for multiplicative tasks. We define the *margin at offset* Δ as the logit gap $\gamma_\Delta = z_{c^*} - z_{c^*+\Delta}$, where $\Delta \in \{1, \ldots, n-1\}$. The per-input cross-entropy loss is then $\mathcal{L} = \log\left(1 + \sum_{\Delta=1}^{n-1} e^{-\gamma_\Delta}\right)$. From Proposition 4, we know how to extend the canonical representation to a higher dimension while preserving the quotient structure, i.e., via direct sum of irreps. We can therefore express the full margin spectrum in terms of the active irreps K and their amplitudes $\{\alpha_k\}_{k\in K}$. Proofs are given in Appendix A.

Corollary 1 (Irrep Margin Spectrum)**.** *Let K be a set of active non-trivial irreps of the quotient group with amplitudes $\alpha_k \geq 0$. For the irrep logit representation over n quotient classes, the margin against the class at offset $\Delta \in \{1, \ldots, n-1\}$ is*

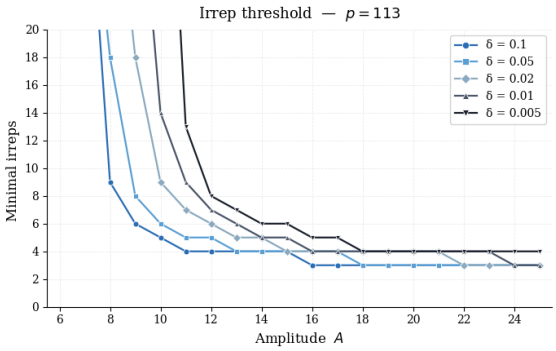
$$\gamma_\Delta(K, \alpha) = \sum_{k\in K} \alpha_k \left(1 - \cos\frac{2\pi k\Delta}{n}\right).$$

Each active irrep contributes a nonnegative term to every γ_Δ, and these margin contributions add across orthogonal subspaces via the direct sum structure. However, not all irreps contribute equally to all offsets. For a given k, the term $1 - \cos(2\pi k\Delta/n)$ may be small for some Δ, meaning that a single irrep cannot uniformly separate all incorrect classes. This leads to a geometric obstruction: unless the set of active irreps covers all offsets Δ, some incorrect predictions remain with small margin. Since the margins translate directly to the cross-entropy loss, we can characterize the sufficient irrep subspaces exactly by substituting the margin decomposition into the loss.

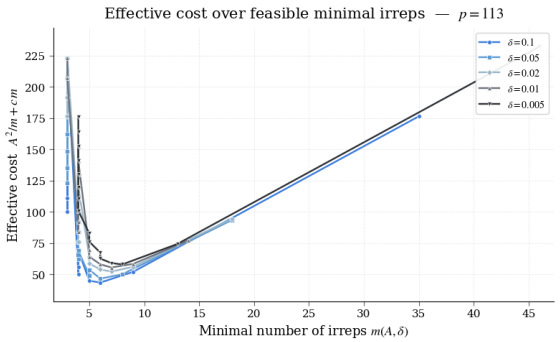
Corollary 2 (Sufficient Representations for Modular Arithmetic Tasks)**.** *Let $V_K = \bigoplus_{k\in K} V_k$. Then the δ-sufficient irrep subspaces are exactly*

$$\mathcal{V}_\delta = \left\{ V_K : \exists\, \alpha_k \geq 0 \text{ such that } \log\left(1 + \sum_{\Delta=1}^{n-1} e^{-\gamma_\Delta(K,\alpha)}\right) \leq \delta \right\}.$$

Corollary 2 characterizes the irrep subspaces capable of achieving loss δ. For a given loss target δ, any active set $K = \{k_1, \ldots, k_s\}$ satisfying the corollary defines a sufficient subspace $V_K = V_{k_1} \oplus \cdots \oplus V_{k_s} \subset \mathbb{R}^n$ spanned by the Fourier modes corresponding to K. The full logit space decomposes orthogonally as $\mathbb{R}^n = V_K \oplus V_K^\perp$, where $V_K^\perp$ contains all logit directions carrying task-irrelevant information. Following Proposition 2, we can calculate the loss contribution of the representational deviation.



(a) Minimal number of irreps required to achieve a target loss δ under the amplitude budget used for the numerical evaluation of Corollary 2



(b) Corresponding effective cost $\mathcal{C}(A, K)$, showing the existence of an optimal number of irreps selected by training. For $p = 113$, 6–7 is typically the minimum.

Figure 3: Effective cost $\mathcal{C}(A, K)$ over feasible irreps for a given loss δ and amplitude A.

3.4 Optimization and Convergence

Corollary 2 characterizes the sufficient irrep subspaces capable of achieving loss at most δ. We now ask which of these sufficient representations gradient-based optimization is biased to select. Under weight decay, optimization favors solutions that achieve the target loss with smaller parameter norm. For modular arithmetic tasks, margins can be increased in two ways: by increasing the amplitude of the active irreps or by activating additional irreps, which improves separation across offsets. These correspond to distinct parameter norms. Higher parameter norms yield higher costs.

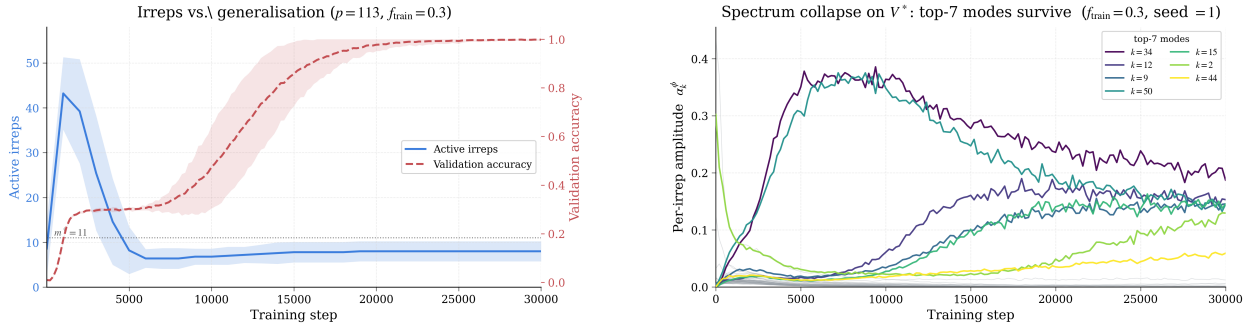
Consider a representation $\phi(x) = \sum_{k \in K} \alpha_k \rho_k(x)$, where K denotes the active irreps and α_k their amplitudes. Under isotropic weight decay, the readout contribution scales with the squared norm of the coefficients, $\sum_{k \in K} \alpha_k^2$. For fixed total amplitude $A = \sum_{k \in K} \alpha_k$, this quantity is minimized when amplitudes are evenly distributed, yielding $\sum_{k \in K} \alpha_k^2 \approx A^2/|K|$. Activating additional irreps also increases parameter norm. Each non-trivial irrep introduces a two-dimensional rotational subspace in the residual stream and requires corresponding upstream circuitry in the attention and MLP layers. We therefore model the architectural cost of activating irreps as increasing approximately linearly with $|K|$. This yields an approximation of the total parameter norm $\mathcal{C}(A, K) = \frac{A^2}{|K|} + c|K|$, where the first term captures coefficient norm under weight decay and the second captures the dimensional and architectural cost of realizing additional irreps.

Under our assumptions, training is therefore biased toward the feasible sufficient representation minimizing $\mathcal{C}(A, K)$ subject to the loss constraint of Corollary 2. The resulting optimum predicts a preferred number of active irreps, shown in Figure 3b.We analyze different parameter norm regimes and the corresponding representational costs in Appendix B.4. This provides a theoretical account of the representations empirically observed by previous work. (Nanda et al., 2023a; Chughtai et al., 2023; Stander et al., 2023; Beck et al., 2024).

This framework also characterizes the transition from memorization to generalization. A sufficient solution occupies only the task-aligned subspace V^*, whereas a memorizing solution additionally retains a component $W_\perp$ orthogonal to V^*. Since $W_\perp$ does not reduce population task loss, weight decay penalizes it without corresponding generalization benefit. The optimizer therefore prefers the sufficient solution whenever the remaining empirical loss is smaller than the regularization cost of the off-subspace component, $\bar{\mathcal{L}}^*(S) < \lambda \|W_\perp\|_F^2$.

4 Experiments

We validate our theory on modular arithmetic tasks (Power et al., 2022; Ruppik et al., 2025) with modulo $p = 113$, over five random seeds. Unless otherwise stated, we fix the target loss to $\delta = 0.02$, corresponding to



(a) Irreps and Loss.

(b) Irrep amplitude spectrum.

Figure 4: **Training dynamics in the irrep basis for f=0.3**. The model progressively concentrates mass on a small subset of irreps, approaching the minimal canonical representation. This is visible as a collapse of the spectrum onto a few dominant modes.

approximately 99% accuracy. Corollary 2 gives the family of sufficient irrep subspaces, and the cost model in Section 3.3 selects the minimal sufficient subspace V^* used for the predictions. The corresponding theoretical predictions are reported in Appendix B.3. More information on the setup is given in Appendix C.

4.1 Sufficient Representations

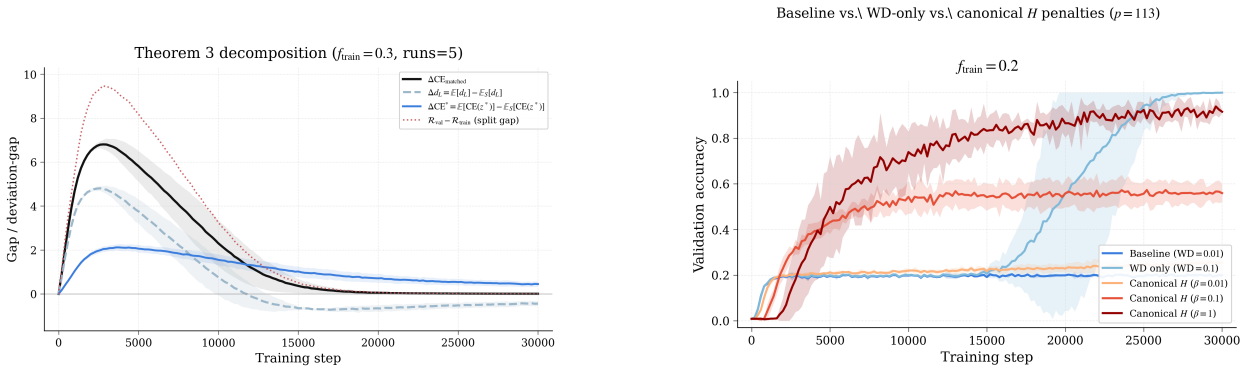
We first test the prediction that a model reaches loss level δ only once its logits are well aligned with a δ-sufficient representation. By Corollary 2, each candidate irrep subspace induces an exact margin spectrum and therefore an exact cross-entropy loss. We enumerate the sufficient representations for each modular arithmetic task at the target loss δ, restricting the amplitude budget to $A \leq 30$ for tractability. We then select the cheapest sufficient representation under the cost model of Section 3.3; this defines the predicted subspace V^*. For modular addition with $p = 113$, the resulting predictions are shown in Figure 3, with further predictions in Appendix B.3.

During training, we cache the final-layer residual stream activations and project the induced logits onto V^*. We measure both the energy contained in V^* and the representational deviation outside it. If the theory is correct, validation loss should drop only when the learned representation has activated the irreps spanning V^*, and the remaining off-subspace energy should decrease.

This is what we observe. Across all four tasks, generalization begins when the predicted sufficient modes become active and gain amplitude. For $p = 113$ and $\delta = 0.02$, the cost model predicts a cheapest sufficient representation using six irreps. As shown in Figure 4, validation loss decreases when these six modes are learned. The spectrum simultaneously concentrates on the predicted subspace (Figure 4b), indicating a reallocation of norm toward V^*. Although the amplitude tends to be distributed evenly amongst irreps, this is not a strict requirement, and we observe dominant irreps in the selected subspace. Grokking thereby corresponds to convergence toward the cost-selected sufficient representation, which guarantees targeted levels of population loss.

4.2 Representational Deviation and Population Loss

We next validate Proposition 2, which states that the loss contribution is determined by the representational deviation H. For each sample, we decompose the logits as $z_i = z_i^* + H_i$, where z_i^* is the projection onto the sufficient subspace V^* identified in Section 3.3. Using this decomposition, we compute the predicted loss $d_{\mathcal{L}}(H_i)$ and compare it to the observed cross-entropy loss. We find near-perfect agreement, with correlation close to 1, confirming that the deviation from the sufficient subspace fully characterizes the loss. Experiments are reported in Appendix D.



(a) Loss and Representational Deviation

(b) Regularization.

Figure 5: **Generalization is governed by representational alignment.** Left: the deviation term tracks excess loss and the empirical generalization gap. Right: adding a canonical deviation penalty accelerates convergence relative to standard weight decay.

We then test the prediction that population loss and generalization is governed by the distribution of the representational deviation, as predicted by Theorem 1. We compute the average deviation cost $d_{\mathcal{L}}(H)$ on both the training set and the full input space and use it to estimate the generalization gap.

The relationship holds across our four selected tasks. Figure 5a illustrates this relationship for modular addition. The black curve shows the predicted excess loss from the representational deviation, while the red curve shows the empirical generalization gap $R(f) - \hat{R}_S(f)$. The two curves closely track each other throughout training: both rise sharply in the early phase, peak at similar training steps, and decay together as training progresses. While the validation gap is estimated from a finite sample and is therefore noisy, the two curves can differ. However the predicted and empirical curves remain closely aligned throughout training. The gap narrows after training.

4.3 Regularization

Finally, we test whether explicitly penalizing representational deviation accelerates convergence. If our hypothesis holds, a regularizer that fits the canonical geometry should speed up training compared to weight decay. We introduce a regularization term $\lambda\|h\|^2$, computed by projecting the representation onto $(V^*)^\perp$. Unlike weight decay, this directly penalizes deviation from the sufficient subspace and enforces alignment with the minimal sufficient representation. We evaluate several strengths of this regularizer, with $\lambda \in \{0.001, 0.01, 0.1, 1\}$, and compare against standard weight decay. Importantly, no weight decay is applied in conjunction with the canonical regularizer, isolating its effect.

Figure 5(b) shows that this approach significantly accelerates generalization for modular addition. Alignment with the sufficient subspace occurs earlier, and validation performance improves correspondingly. Higher sufficient-subspace regularization also leads to faster convergence, while small amounts (0.001) yield little results.

Compared to standard weight decay, the canonical regularizer consistently speeds up convergence and enables generalization at lower data fractions. Notably, settings with $f_{\text{train}} = 0.1$ generalize under this regularizer, whereas they fail to do so with the standard weight decay. For larger amounts of training data, the canonical regularizer remains effective but can slow optimization slightly, reflecting the trade-off between alignment and flexibility. These results confirm that representational deviation is a key quantity governing generalization and that controlling this deviation directly affects generalization and convergence speed. The rest of the experiments can be found in Appendix D.

5 Related Work

Generalization and Generalization Bounds How gradient-based training restricts the parameter space to a much smaller set of realized functions is widely viewed as a core open problem in deep learning theory (Roberts et al., 2022). Classical PAC bounds (Valiant, 1984; Vapnik, 2013), PAC-Bayes approaches (McAllester, 1998), and more recent analyses provide partial explanations in restricted regimes (Watanabe, 2009; Jacot et al., 2020; Tishby & Zaslavsky, 2015). Despite this progress, a general, predictive characterization of generalization that accounts for representation learning remains unresolved.

Universality Hypothesis As training minimizes empirical risk, it shapes the model's internal feature map to organize the data in ways that make the task easier to solve (LeCun et al., 2015; Goodfellow, 2016; Whiteley et al., 2025). A recurring empirical observation is that, for a fixed task and sufficiently large models, optimization often discovers similar internal mechanisms across runs and architectures. This has been termed the universality hypothesis (Olah et al., 2020). Recent work has given empirical evidence of converging representations in large language models (Jha et al., 2025; Kaushik et al., 2025). The challenge that research needs to address is to turn these regularities into predictive theory.

Grokking Toy setups have been widely used to identify properties of generalization (Power et al., 2022). Our theory explains empirical results obtained in previous research (Nanda et al., 2023b), where training moves from a memorizing regime to a richer regime in which the learned representation collapses onto a smaller task-aligned subspace (Kumar et al., 2023). Interventions that amplify the effective weight of low-frequency components, such as GrokFast (Lee et al., 2024) or related gradient-filtering schemes (Xu et al., 2025; Beck et al., 2024), can be understood as artificially reducing representation deviation early in training (Liu et al., 2022b).

6 Discussion

Contributions This work derives a predictive theory of grokking from the algebraic structure of the target function. It explains a set of independently observed empirical phenomena in modular arithmetic and provides an account of generalization in this restricted setting. The central bridge is geometric: generalization occurs when the learned representation aligns with the canonical subspace determined by the task. This also provides a concrete basis for the universality hypothesis, since the **canonical representation of a task** is derivable from the task alone and is independent of any particular model instantiation.

Limitations Our approach faces limitations. It relies on knowing f^* a priori. Extending this approach beyond the toy setup presents substantial challenges. The derivation of the canonical geometry is straightforward given the simple structure of f^* in modular addition but is bound to be significantly more complicated for real-world tasks. Despite these limitations, this theory still provides an initial step towards a general theory of generalization and feature learning in deep learning models.

References

Alon Beck, Noam Levi, and Yohai Bar-Sinai. Grokking at the edge of linear separability. *arXiv preprint arXiv:2410.04489*, 2024.

Satrajit Chatterjee and Piotr Zielinski. On the Generalization Mystery in Deep Learning, June 2022. arXiv:2203.10036 [cs].

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*, pp. 6243–6267. PMLR, 2023.

William Fulton and Joe Harris. *Representation theory: a first course*. Springer Science & Business Media, 2013.

Ian Goodfellow. Deep learning, 2016.

Zhengyu Hu, Linxin Song, Jieyu Zhang, Zheyuan Xiao, Jingang Wang, Zhenyu Chen, Jieyu Zhao, and Hui Xiong. Rethinking LLM-based Preference Evaluation. *arXiv preprint arXiv:2407.01085*, 2024.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural Tangent Kernel: Convergence and Generalization in Neural Networks, February 2020. arXiv:1806.07572.

Rishi Jha, Collin Zhang, Vitaly Shmatikov, and John X. Morris. Harnessing the Universal Geometry of Embeddings, June 2025. arXiv:2505.12540 [cs].

Prakhar Kaushik, Shravan Chaudhari, Ankit Vaidya, Rama Chellappa, and Alan Yuille. The Universal Weight Subspace Hypothesis, December 2025. arXiv:2512.05117 [cs].

Tanishq Kumar, Blake Bordelon, Samuel J Gershman, and Cengiz Pehlevan. Grokking as the transition from lazy to rich training dynamics. In *The twelfth international conference on learning representations*, 2023.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

Jaerin Lee, Bong Gyun Kang, Kihoon Kim, and Kyoung Mu Lee. Grokfast: Accelerated grokking by amplifying slow gradients. *arXiv preprint arXiv:2405.20233*, 2024.

Ziming Liu, Ouail Kitouni, Niklas S Nolte, Eric Michaud, Max Tegmark, and Mike Williams. Towards understanding grokking: An effective theory of representation learning. *Advances in Neural Information Processing Systems*, 35:34651–34663, 2022a.

Ziming Liu, Eric J Michaud, and Max Tegmark. Omnigrok: Grokking beyond algorithmic data. *arXiv preprint arXiv:2210.01117*, 2022b.

David A McAllester. Some pac-bayesian theorems. In *Proceedings of the eleventh annual conference on Computational learning theory*, pp. 230–234, 1998.

Tom M. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability, October 2023a. arXiv:2301.05217 [cs].

Neel Nanda, Andrew Lee, and Martin Wattenberg. Emergent Linear Representations in World Models of Self-Supervised Sequence Models, September 2023b. arXiv:2309.00941.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 5(3):e00024–001, 2020.

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*, 2022.

Daniel A Roberts, Sho Yaida, and Boris Hanin. *The principles of deep learning theory*, volume 46. Cambridge University Press Cambridge, MA, USA, 2022.

Benjamin Matthias Ruppik, Julius von Rohrscheidt, Carel van Niekerk, Michael Heck, Renato Vukovic, Shutong Feng, Hsien-chin Lin, Nurul Lubis, Bastian Rieck, Marcus Zibrowius, et al. Less is more: Local intrinsic dimensions of contextual language models. *arXiv preprint arXiv:2506.01034*, 2025.

Jean-Pierre Serre et al. *Linear representations of finite groups*, volume 42. Springer, 1977.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Dashiell Stander, Qinan Yu, Honglu Fan, and Stella Biderman. Grokking group multiplication with cosets. *arXiv preprint arXiv:2312.06581*, 2023.

Naftali Tishby and Noga Zaslavsky. Deep Learning and the Information Bottleneck Principle, March 2015. arXiv:1503.02406.

Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

Sumio Watanabe. *Algebraic geometry and statistical learning theory*, volume 25. Cambridge university press, 2009.

Nick Whiteley, Annie Gray, and Patrick Rubin-Delanchy. Statistical exploration of the Manifold Hypothesis, March 2025. arXiv:2208.11665 [stat].

Zhiwei Xu, Zhiyu Ni, Yixin Wang, and Wei Hu. Let me grok for you: Accelerating grokking via embedding transfer from a weaker model. *arXiv preprint arXiv:2504.13292*, 2025.

Bojan Žunkovič and Enej Ilievski. Grokking phase transitions in learning local rules with gradient descent. *Journal of Machine Learning Research*, 25(199):1–52, 2024.

A Proofs

A.1 Proof of Corrolary 1

Corrolary 1 (Irrep Margin Spectrum). *Let K be a set of active non-trivial irreps of the quotient group with amplitudes $\alpha_k \geq 0$. For the irrep logit representation over n quotient classes, the margin against the class at offset $\Delta \in \{1, \ldots, n-1\}$ is*

$$\gamma_\Delta(K, \alpha) = \sum_{k \in K} \alpha_k \left(1 - \cos \frac{2\pi k \Delta}{n}\right).$$

Proof. By construction, the contribution of irrep k to the logit of class c is

$$z_c^{(k)} = \alpha_k \cos\left(\frac{2\pi k(c - c^*)}{n}\right), \tag{A.1}$$

where c^* is the correct class. Summing over all active irreps gives

$$z_c = \sum_{k \in K} \alpha_k \cos\left(\frac{2\pi k(c - c^*)}{n}\right). \tag{A.2}$$

For an offset $\Delta \in \{1, \ldots, n-1\}$, the margin is

$$\begin{aligned} \gamma_\Delta(K, \alpha) &= z_{c^*} - z_{c^*+\Delta} && \text{(A.3)} \\ &= \sum_{k \in K} \alpha_k - \sum_{k \in K} \alpha_k \cos\left(\frac{2\pi k \Delta}{n}\right) && \text{(A.4)} \\ &= \sum_{k \in K} \alpha_k \left(1 - \cos \frac{2\pi k \Delta}{n}\right). && \text{(A.5)} \end{aligned}$$

Equation equation A.4 follows from evaluating equation A.2 at $c = c^*$ and $c = c^* + \Delta$, using $\cos(0) = 1$. Equation equation A.5 is obtained by factoring the sum. □

A.2 Proof of Corollary 2

Corollary 2 (Sufficient Representations for Modular Arithmetic Tasks). *Let $V_K = \bigoplus_{k\in K} V_k$. Then the δ-sufficient irrep subspaces are exactly*

$$\mathcal{V}_\delta = \left\{ V_K : \exists\, \alpha_k \geq 0 \text{ such that } \log\left(1 + \sum_{\Delta=1}^{n-1} e^{-\gamma_\Delta(K,\alpha)}\right) \leq \delta \right\}.$$

Proof. For a sample whose correct class is c^*, every incorrect class is represented by an offset $\Delta \in \{1, \ldots, n-1\}$. The cross-entropy loss of the irrep representation (K, α) is therefore

$$\mathcal{L}(K,\alpha) = \log\left(1 + \sum_{\Delta=1}^{n-1} e^{-\gamma_\Delta(K,\alpha)}\right). \tag{A.6}$$

By Corrolary 1, the margins $\gamma_\Delta(K,\alpha)$ are exactly the margins induced by the active irrep subspace V_K with amplitudes α. Hence a fixed pair (K,α) achieves cross-entropy loss at most δ if and only if $\mathcal{L}(K,\alpha) \leq \delta$.

The subspace V_K is δ-sufficient precisely when there exists at least one amplitude assignment $\alpha_k \geq 0$ inside that subspace achieving loss at most δ. Therefore the family of δ-sufficient irrep subspaces is exactly

$$\mathcal{V}_\delta = \left\{ V_K : \exists\, \alpha_k \geq 0 \text{ such that } \log\left(1 + \sum_{\Delta=1}^{n-1} e^{-\gamma_\Delta(K,\alpha)}\right) \leq \delta \right\}.$$

□

A.3 Proof of Proposition 2

Proposition 2 (Deviation Loss Contribution). *Let $V \in \mathcal{V}_\delta^L$ be a chosen sufficient subspace and decompose the logits for sample x_i as $z_i = z_i^* + H_i^{(V)}$. For cross-entropy loss, where $y_i^* = f^*(x_i)$, the per-sample loss contribution of the representational deviation is*

$$d_{\mathcal{L}}^{(V)}(H_i) = \log \sum_{y\in Y} e^{z_{iy}^* + H_{iy}^{(V)}} - \log \sum_{y\in Y} e^{z_{iy}^*} - H_{i,y_i^*}^{(V)}.$$

Proof. For sample i, let $y_i^* = f^*(x_i)$ denote the correct label. The cross-entropy loss of the model logits $z_i = z_i^* + H_i^{(V)}$ is

$$\mathcal{L}_i = -z_{i,y_i^*} + \log \sum_{y\in Y} e^{z_{iy}} \tag{A.7}$$

$$= -\left(z_{i,y_i^*}^* + H_{i,y_i^*}^{(V)}\right) + \log \sum_{y\in Y} e^{z_{iy}^* + H_{iy}^{(V)}}. \tag{A.8}$$

Likewise, the loss of the sufficient logits alone is

$$\mathcal{L}_i^* = -z_{i,y_i^*}^* + \log \sum_{y\in Y} e^{z_{iy}^*}. \tag{A.9}$$

Subtracting equation A.9 from equation A.8 gives the excess loss

$$\mathcal{L}_i - \mathcal{L}_i^* = \log \sum_{y\in Y} e^{z_{iy}^* + H_{iy}^{(V)}} - \log \sum_{y\in Y} e^{z_{iy}^*} - H_{i,y_i^*}^{(V)}. \tag{A.10}$$

By definition,

$$d_{\mathcal{L}}^{(V)}(H_i) = \mathcal{L}_i - \mathcal{L}_i^*$$

which is exactly the claimed expression. □

A.4 Proof of Theorem 1

Theorem 1 (Population Risk from Representational Deviation). *Let $V \in \mathcal{V}_\delta^L$ be a chosen sufficient subspace, and decompose the logits as $z = z^* + H^{(V)}$. Then*

$$R(f) \leq \delta + \mathbb{E}_{x \sim P_X}\left[d_{\mathcal{L}}^{(V)}(H(x))\right].$$

Proof. By definition of $d_{\mathcal{L}}^{(V)}$,

$$d_{\mathcal{L}}^{(V)}(H(x)) = \mathcal{L}(z^*(x) + H^{(V)}(x)) - \mathcal{L}(z^*(x)).$$

Taking expectation over the input distribution gives

$$R(f) = \mathbb{E}_{x \sim P_X}[\mathcal{L}(z^*(x))] + \mathbb{E}_{x \sim P_X}\left[d_{\mathcal{L}}^{(V)}(H(x))\right].$$

Since $V \in \mathcal{V}_\delta^L$ is δ-sufficient, $\mathbb{E}_{x \sim P_X}[\mathcal{L}(z^*(x))] \leq \delta$. Substituting this bound gives the result. □

B Canonical Decompositions and Representations

All modular arithmetic tasks considered in this work admit the same general structure. Given a task $f^* : X \to Y$, the canonical decomposition is $f^* = \bar{f}^* \circ \pi$, where $\pi : X \to X/\sim$ maps each input to its equivalence class under $x \sim x'$ iff $f^*(x) = f^*(x')$, and $\bar{f}^*$ is the induced injective map on the quotient.

B.1 Additive Tasks

For modular addition and subtraction over $\mathbb{Z}_p$, the target functions are $f^*(a,b) = a+b \pmod p$ and $f^*(a,b) = a - b \pmod p$. Both induce quotient structures over the additive cyclic group $(\mathbb{Z}_p, +)$.

For addition, the equivalence relation is $(a,b) \sim (a',b')$ iff $a + b \equiv a' + b' \pmod p$, yielding quotient map $\pi_+(a,b) = a + b \pmod p$. For subtraction, the quotient map is $\pi_-(a,b) = a - b \pmod p$.

In both cases, the quotient space is isomorphic to $\mathbb{Z}_p$, and the canonical decomposition is $f^* = \mathrm{id}_{\mathbb{Z}_p} \circ \pi$.

Canonical representations are therefore given by the irreducible representations of $\mathbb{Z}_p$. Over $\mathbb{R}$, the non-trivial irreducible blocks are the two-dimensional rotations

$$\rho_k(t) = \begin{pmatrix} \cos(2\pi kt/p) & -\sin(2\pi kt/p) \\ \sin(2\pi kt/p) & \cos(2\pi kt/p) \end{pmatrix}, \qquad k = 1, \ldots, \frac{p-1}{2}. \tag{B.1}$$

The canonical representations are thus

$$\phi(a,b) = \rho_k(a \pm b), \tag{B.2}$$

or more generally

$$\phi(a,b) = \bigoplus_{k \in K} \rho_k(a \pm b). \tag{B.3}$$

B.2 Multiplicative Tasks

For modular multiplication and division, the target functions are $f^*(a,b) = ab \pmod p$ and $f^*(a,b) = ab^{-1} \pmod p$. These induce quotient structures over the multiplicative cyclic group $(\mathbb{Z}_p^\times, \cdot)$.

The equivalence relations are $(a,b) \sim (a',b')$ iff $ab \equiv a'b' \pmod p$ for multiplication, and $(a,b) \sim (a',b')$ iff $ab^{-1} \equiv a'(b')^{-1} \pmod p$ for division.

Since $\mathbb{Z}_p^\times$ is cyclic, let g be a generator and write $a = g^u$ and $b = g^v$. Multiplication and division then reduce to additive structure in exponent space: $ab = g^{u+v}$ and $ab^{-1} = g^{u-v}$.

The quotient maps therefore become $\pi_\times(g^u, g^v) = u + v \pmod{p-1}$ and $\pi_\div(g^u, g^v) = u - v \pmod{p-1}$.

The canonical quotient is thus isomorphic to $\mathbb{Z}_{p-1}$, and canonical representations are given by faithful irreducible representations of $\mathbb{Z}_{p-1}$. Over $\mathbb{R}$, the faithful two-dimensional rotation blocks are

$$\rho_k^\times(t) = \begin{pmatrix} \cos(2\pi kt/(p-1)) & -\sin(2\pi kt/(p-1)) \\ \sin(2\pi kt/(p-1)) & \cos(2\pi kt/(p-1)) \end{pmatrix}, \qquad 1 \le k \le \frac{p-2}{2}, \qquad \gcd(k, p-1) = 1. \tag{B.4}$$

The canonical representations are therefore

$$\phi(a, b) = \rho_k^\times(\log_g a \pm \log_g b), \tag{B.5}$$

or more generally

$$\phi(a, b) = \bigoplus_{k \in K} \rho_k^\times(\log_g a \pm \log_g b). \tag{B.6}$$

The zero element is treated separately, since $0 \notin \mathbb{Z}_p^\times$.

B.3 Theoretical Representations

The result of a greedy approximation of the theoretical quantity of the canonical representations

	$\delta = 0.01$			$\delta = 0.02$			$\delta = 0.05$			$\delta = 0.1$		
A	$m^\dagger$	α	Loss	$m^\dagger$	α	Loss	$m^\dagger$	α	Loss	$m^\dagger$	α	Loss
8	*unf.*	*unf.*	*unf.*	*unf.*	*unf.*	*unf.*	14	0.57	0.0484	7	1.14	0.0961
9	*unf.*	*unf.*	*unf.*	14	0.64	0.0199	7	1.29	0.0458	5	1.80	0.0927
10	12	0.83	0.0098	8	1.25	0.0161	5	2.00	0.0498	5	2.00	0.0498
11	8	1.38	0.0073	6	1.83	0.0170	5	2.20	0.0268	4	2.75	0.0663
12	6	2.00	0.0085	5	2.40	0.0149	4	3.00	0.0404	4	3.00	0.0404
13	5	2.60	0.0082	5	2.60	0.0082	4	3.25	0.0246	4	3.25	0.0246
14	5	2.80	0.0045	4	3.50	0.0150	4	3.50	0.0150	4	3.50	0.0150
15	4	3.75	0.0092	4	3.75	0.0092	4	3.75	0.0092	3	5.00	0.0852
16	4	4.00	0.0056	4	4.00	0.0056	4	4.00	0.0056	3	5.33	0.0640
17	4	4.25	0.0035	4	4.25	0.0035	3	5.67	0.0482	3	5.67	0.0482
18	4	4.50	0.0021	4	4.50	0.0021	3	6.00	0.0363	3	6.00	0.0363
19	4	4.75	0.0013	4	4.75	0.0013	3	6.33	0.0274	3	6.33	0.0274
20	4	5.00	0.0008	4	5.00	0.0008	3	6.67	0.0207	3	6.67	0.0207

Table 1: Predicted minimal feasible representations for $p = 97$. Each block reports the greedy exact-loss approximation of the minimal number of irreps $m^\dagger$, the corresponding uniform amplitude $\alpha = A/m^\dagger$, and the achieved Loss. Entries marked *unf.* indicate that no feasible representation satisfying the target loss was found.

	$\delta = 0.01$			$\delta = 0.02$			$\delta = 0.05$			$\delta = 0.1$		
A	$m^\dagger$	α	Loss	$m^\dagger$	α	Loss	$m^\dagger$	α	Loss	$m^\dagger$	α	Loss
8	*unf.*	*unf.*	*unf.*	*unf.*	*unf.*	*unf.*	18	0.44	0.0496	9	0.89	0.0869
9	*unf.*	*unf.*	*unf.*	18	0.50	0.0198	8	1.12	0.0477	6	1.50	0.0785
10	14	0.71	0.0099	9	1.11	0.0179	6	1.67	0.0400	5	2.00	0.0765
11	9	1.22	0.0081	7	1.57	0.0139	5	2.20	0.0443	4	2.75	0.0900
12	7	1.71	0.0067	6	2.00	0.0104	5	2.40	0.0259	4	3.00	0.0558
13	6	2.17	0.0053	5	2.60	0.0153	4	3.25	0.0346	4	3.25	0.0346
14	5	2.80	0.0091	5	2.80	0.0091	4	3.50	0.0214	4	3.50	0.0214
15	5	3.00	0.0055	4	3.75	0.0133	4	3.75	0.0133	4	3.75	0.0133
16	4	4.00	0.0083	4	4.00	0.0083	4	4.00	0.0083	3	5.33	0.0873
17	4	4.25	0.0051	4	4.25	0.0051	4	4.25	0.0051	3	5.67	0.0659
18	4	4.50	0.0032	4	4.50	0.0032	3	6.00	0.0498	3	6.00	0.0498
19	4	4.75	0.0020	4	4.75	0.0020	3	6.33	0.0376	3	6.33	0.0376
20	4	5.00	0.0013	4	5.00	0.0013	3	6.67	0.0284	3	6.67	0.0284

Table 2: Predicted minimal feasible representations for $p = 113$. Each block reports the greedy exact-loss approximation of the minimal number of irreps $m^\dagger$, the corresponding uniform amplitude $\alpha = A/m^\dagger$, and the achieved Loss. Entries marked *unf.* indicate that no feasible representation satisfying the target loss was found.

	$\delta = 0.01$			$\delta = 0.02$			$\delta = 0.05$			$\delta = 0.1$		
A	$m^\dagger$	α	Loss	$m^\dagger$	α	Loss	$m^\dagger$	α	Loss	$m^\dagger$	α	Loss
8	*unf.*	*unf.*	*unf.*	*unf.*	*unf.*	*unf.*	*unf.*	*unf.*	*unf.*	18	0.44	0.0976
9	*unf.*	*unf.*	*unf.*	*unf.*	*unf.*	*unf.*	14	0.64	0.0496	9	1.00	0.0899
10	48	0.21	0.0099	16	0.62	0.0189	9	1.11	0.0438	7	1.43	0.0717
11	14	0.79	0.0094	10	1.10	0.0177	7	1.57	0.0373	6	1.83	0.0600
12	10	1.20	0.0084	7	1.71	0.0194	6	2.00	0.0338	5	2.40	0.0622
13	8	1.62	0.0070	6	2.17	0.0191	5	2.60	0.0375	5	2.60	0.0375
14	7	2.00	0.0053	6	2.33	0.0109	5	2.80	0.0227	4	3.50	0.0794
15	6	2.50	0.0062	5	3.00	0.0137	5	3.00	0.0137	4	3.75	0.0546
16	5	3.20	0.0084	5	3.20	0.0084	4	4.00	0.0376	4	4.00	0.0376
17	5	3.40	0.0051	5	3.40	0.0051	4	4.25	0.0259	4	4.25	0.0259
18	5	3.60	0.0031	4	4.50	0.0179	4	4.50	0.0179	4	4.50	0.0179
19	5	3.80	0.0028	4	4.75	0.0124	4	4.75	0.0124	4	4.75	0.0124
20	4	5.00	0.0085	4	5.00	0.0085	4	5.00	0.0085	4	5.00	0.0085

Table 3: Predicted minimal feasible representations for $p = 197$. Each block reports the greedy exact-loss approximation of the minimal number of irreps $m^\dagger$, the corresponding uniform amplitude $\alpha = A/m^\dagger$, and the achieved Loss. Entries marked *unf.* indicate that no feasible representation satisfying the target loss was found.

B.4 Theoretical Cost Curves

	$\delta = 0.01$					$\delta = 0.02$				
A	$m^\dagger$	$\mathcal{C}_{c=4}$	$\mathcal{C}_{c=5}$	$\mathcal{C}_{c=6}$	$\mathcal{C}_{c=8}$	$m^\dagger$	$\mathcal{C}_{c=4}$	$\mathcal{C}_{c=5}$	$\mathcal{C}_{c=6}$	$\mathcal{C}_{c=8}$
8	*unf.*	*unf.*	*unf.*	*unf.*	*unf.*	*unf.*	*unf.*	*unf.*	*unf.*	*unf.*
9	*unf.*	*unf.*	*unf.*	*unf.*	*unf.*	14	61.79	75.79	89.79	117.79
10	12	56.33	68.33	80.33	104.33	8	44.50	52.50	60.50	76.50
11	8	47.12	55.12	63.12	79.12	6	44.17	50.17	56.17	68.17
12	6	48.00	54.00	60.00	72.00	5	48.80	53.80	58.80	68.80
13	5	53.80	58.80	63.80	73.80	5	53.80	58.80	63.80	73.80
14	5	59.20	64.20	69.20	79.20	4	65.00	69.00	73.00	81.00
15	4	72.25	76.25	80.25	88.25	4	72.25	76.25	80.25	88.25
16	4	80.00	84.00	88.00	96.00	4	80.00	84.00	88.00	96.00
17	4	88.25	92.25	96.25	104.25	4	88.25	92.25	96.25	104.25
18	4	97.00	101.00	105.00	113.00	4	97.00	101.00	105.00	113.00
19	4	106.25	110.25	114.25	122.25	4	106.25	110.25	114.25	122.25
20	4	116.00	120.00	124.00	132.00	4	116.00	120.00	124.00	132.00

Table 4: Effective costs for predicted minimal feasible representations for $p = 97$ and $\delta \in \{0.01, 0.02\}$. Costs are reported for $c \in \{4, 5, 6, 8\}$ using $\mathcal{C}(A, K) = A^2/|K| + c|K|$. Entries marked *unf.* indicate that no feasible representation satisfying the target loss was found.

	$\delta = 0.05$					$\delta = 0.1$				
A	$m^\dagger$	$\mathcal{C}_{c=4}$	$\mathcal{C}_{c=5}$	$\mathcal{C}_{c=6}$	$\mathcal{C}_{c=8}$	$m^\dagger$	$\mathcal{C}_{c=4}$	$\mathcal{C}_{c=5}$	$\mathcal{C}_{c=6}$	$\mathcal{C}_{c=8}$
8	14	60.57	74.57	88.57	116.57	7	37.14	44.14	51.14	65.14
9	7	39.57	46.57	53.57	67.57	5	36.20	41.20	46.20	56.20
10	5	40.00	45.00	50.00	60.00	5	40.00	45.00	50.00	60.00
11	5	44.20	49.20	54.20	64.20	4	46.25	50.25	54.25	62.25
12	4	52.00	56.00	60.00	68.00	4	52.00	56.00	60.00	68.00
13	4	58.25	62.25	66.25	74.25	4	58.25	62.25	66.25	74.25
14	4	65.00	69.00	73.00	81.00	4	65.00	69.00	73.00	81.00
15	4	72.25	76.25	80.25	88.25	3	87.00	90.00	93.00	99.00
16	4	80.00	84.00	88.00	96.00	3	97.33	100.33	103.33	109.33
17	3	108.33	111.33	114.33	120.33	3	108.33	111.33	114.33	120.33
18	3	120.00	123.00	126.00	132.00	3	120.00	123.00	126.00	132.00
19	3	132.33	135.33	138.33	144.33	3	132.33	135.33	138.33	144.33
20	3	145.33	148.33	151.33	157.33	3	145.33	148.33	151.33	157.33

Table 5: Effective costs for predicted minimal feasible representations for $p = 97$ and $\delta \in \{0.05, 0.1\}$. Costs are reported for $c \in \{4, 5, 6, 8\}$ using $\mathcal{C}(A, K) = A^2/|K| + c|K|$. Entries marked *unf.* indicate that no feasible representation satisfying the target loss was found.

	$\delta = 0.01$					$\delta = 0.02$				
A	$m^\dagger$	$\mathcal{C}_{c=4}$	$\mathcal{C}_{c=5}$	$\mathcal{C}_{c=6}$	$\mathcal{C}_{c=8}$	$m^\dagger$	$\mathcal{C}_{c=4}$	$\mathcal{C}_{c=5}$	$\mathcal{C}_{c=6}$	$\mathcal{C}_{c=8}$
8	*unf.*	*unf.*	*unf.*	*unf.*	*unf.*	*unf.*	*unf.*	*unf.*	*unf.*	*unf.*
9	*unf.*	*unf.*	*unf.*	*unf.*	*unf.*	18	76.50	94.50	112.50	148.50
10	14	63.14	77.14	91.14	119.14	9	47.11	56.11	65.11	83.11
11	9	49.44	58.44	67.44	85.44	7	45.29	52.29	59.29	73.29
12	7	48.57	55.57	62.57	76.57	6	48.00	54.00	60.00	72.00
13	6	52.17	58.17	64.17	76.17	5	53.80	58.80	63.80	73.80
14	5	59.20	64.20	69.20	79.20	5	59.20	64.20	69.20	79.20
15	5	65.00	70.00	75.00	85.00	4	72.25	76.25	80.25	88.25
16	4	80.00	84.00	88.00	96.00	4	80.00	84.00	88.00	96.00
17	4	88.25	92.25	96.25	104.25	4	88.25	92.25	96.25	104.25
18	4	97.00	101.00	105.00	113.00	4	97.00	101.00	105.00	113.00
19	4	106.25	110.25	114.25	122.25	4	106.25	110.25	114.25	122.25
20	4	116.00	120.00	124.00	132.00	4	116.00	120.00	124.00	132.00

Table 6: Effective costs for predicted minimal feasible representations for $p = 113$ and $\delta \in \{0.01, 0.02\}$. Costs are reported for $c \in \{4, 5, 6, 8\}$ using $\mathcal{C}(A, K) = A^2/|K| + c|K|$. Entries marked *unf.* indicate that no feasible representation satisfying the target loss was found.

	$\delta = 0.05$					$\delta = 0.1$				
A	$m^\dagger$	$\mathcal{C}_{c=4}$	$\mathcal{C}_{c=5}$	$\mathcal{C}_{c=6}$	$\mathcal{C}_{c=8}$	$m^\dagger$	$\mathcal{C}_{c=4}$	$\mathcal{C}_{c=5}$	$\mathcal{C}_{c=6}$	$\mathcal{C}_{c=8}$
8	18	75.56	93.56	111.56	147.56	9	43.11	52.11	61.11	79.11
9	8	42.12	50.12	58.12	74.12	6	37.50	43.50	49.50	61.50
10	6	40.67	46.67	52.67	64.67	5	40.00	45.00	50.00	60.00
11	5	44.20	49.20	54.20	64.20	4	46.25	50.25	54.25	62.25
12	5	48.80	53.80	58.80	68.80	4	52.00	56.00	60.00	68.00
13	4	58.25	62.25	66.25	74.25	4	58.25	62.25	66.25	74.25
14	4	65.00	69.00	73.00	81.00	4	65.00	69.00	73.00	81.00
15	4	72.25	76.25	80.25	88.25	4	72.25	76.25	80.25	88.25
16	4	80.00	84.00	88.00	96.00	3	97.33	100.33	103.33	109.33
17	4	88.25	92.25	96.25	104.25	3	108.33	111.33	114.33	120.33
18	3	120.00	123.00	126.00	132.00	3	120.00	123.00	126.00	132.00
19	3	132.33	135.33	138.33	144.33	3	132.33	135.33	138.33	144.33
20	3	145.33	148.33	151.33	157.33	3	145.33	148.33	151.33	157.33

Table 7: Effective costs for predicted minimal feasible representations for $p = 113$ and $\delta \in \{0.05, 0.1\}$. Costs are reported for $c \in \{4, 5, 6, 8\}$ using $\mathcal{C}(A, K) = A^2/|K| + c|K|$. Entries marked *unf.* indicate that no feasible representation satisfying the target loss was found.

	$\delta = 0.01$					$\delta = 0.02$				
A	$m^\dagger$	$\mathcal{C}_{c=4}$	$\mathcal{C}_{c=5}$	$\mathcal{C}_{c=6}$	$\mathcal{C}_{c=8}$	$m^\dagger$	$\mathcal{C}_{c=4}$	$\mathcal{C}_{c=5}$	$\mathcal{C}_{c=6}$	$\mathcal{C}_{c=8}$
8	*unf.*	*unf.*	*unf.*	*unf.*	*unf.*	*unf.*	*unf.*	*unf.*	*unf.*	*unf.*
9	*unf.*	*unf.*	*unf.*	*unf.*	*unf.*	*unf.*	*unf.*	*unf.*	*unf.*	*unf.*
10	48	194.08	242.08	290.08	386.08	16	70.25	86.25	102.25	134.25
11	14	64.64	78.64	92.64	120.64	10	52.10	62.10	72.10	92.10
12	10	54.40	64.40	74.40	94.40	7	48.57	55.57	62.57	76.57
13	8	53.12	61.12	69.12	85.12	6	52.17	58.17	64.17	76.17
14	7	56.00	63.00	70.00	84.00	6	56.67	62.67	68.67	80.67
15	6	61.50	67.50	73.50	85.50	5	65.00	70.00	75.00	85.00
16	5	71.20	76.20	81.20	91.20	5	71.20	76.20	81.20	91.20
17	5	77.80	82.80	87.80	97.80	5	77.80	82.80	87.80	97.80
18	5	84.80	89.80	94.80	104.80	4	97.00	101.00	105.00	113.00
19	5	92.20	97.20	102.20	112.20	4	106.25	110.25	114.25	122.25
20	4	116.00	120.00	124.00	132.00	4	116.00	120.00	124.00	132.00

Table 8: Effective costs for predicted minimal feasible representations for $p = 197$ and $\delta \in \{0.01, 0.02\}$. Costs are reported for $c \in \{4, 5, 6, 8\}$ using $\mathcal{C}(A, K) = A^2/|K| + c|K|$. Entries marked *unf.* indicate that no feasible representation satisfying the target loss was found.

	$\delta = 0.05$					$\delta = 0.1$				
A	$m^\dagger$	$\mathcal{C}_{c=4}$	$\mathcal{C}_{c=5}$	$\mathcal{C}_{c=6}$	$\mathcal{C}_{c=8}$	$m^\dagger$	$\mathcal{C}_{c=4}$	$\mathcal{C}_{c=5}$	$\mathcal{C}_{c=6}$	$\mathcal{C}_{c=8}$
8	*unf.*	*unf.*	*unf.*	*unf.*	*unf.*	18	75.56	93.56	111.56	147.56
9	14	61.79	75.79	89.79	117.79	9	45.00	54.00	63.00	81.00
10	9	47.11	56.11	65.11	83.11	7	42.29	49.29	56.29	70.29
11	7	45.29	52.29	59.29	73.29	6	44.17	50.17	56.17	68.17
12	6	48.00	54.00	60.00	72.00	5	48.80	53.80	58.80	68.80
13	5	53.80	58.80	63.80	73.80	5	53.80	58.80	63.80	73.80
14	5	59.20	64.20	69.20	79.20	4	65.00	69.00	73.00	81.00
15	5	65.00	70.00	75.00	85.00	4	72.25	76.25	80.25	88.25
16	4	80.00	84.00	88.00	96.00	4	80.00	84.00	88.00	96.00
17	4	88.25	92.25	96.25	104.25	4	88.25	92.25	96.25	104.25
18	4	97.00	101.00	105.00	113.00	4	97.00	101.00	105.00	113.00
19	4	106.25	110.25	114.25	122.25	4	106.25	110.25	114.25	122.25
20	4	116.00	120.00	124.00	132.00	4	116.00	120.00	124.00	132.00

Table 9: Effective costs for predicted minimal feasible representations for $p = 197$ and $\delta \in \{0.05, 0.1\}$. Costs are reported for $c \in \{4, 5, 6, 8\}$ using $\mathcal{C}(A, K) = A^2/|K| + c|K|$. Entries marked *unf.* indicate that no feasible representation satisfying the target loss was found.

C Experimental Setup

Task. We study grokking on modular arithmetic over the cyclic group $\mathbb{Z}_p$. Each example is a token sequence $(a, \circ, b, =)$ with target $y = a \circ b \pmod p$, where $\circ$ corresponds to addition, substraction, division and multiplication. The vocabulary has size $p + 2$ (two special tokens $\circ$ and $=$, plus the p group elements). Following Power et al. (2022), we sample a fraction $\rho \in [0.1, 0.5]$ of the p^2 pairs uniformly at random as the training set and use the remaining pairs as the held-out validation set. Unless stated otherwise we use $p = 113$, $\rho = 0.3$, and global seed 42. The full sweep over ρ is run on $\rho \in \{0.1, 0.15, 0.2, 0.25, 0.3, 0.4, 0.5\}$.

Model. We use a standard decoder-only transformer with pre-norm: $L = 2$ blocks, $h = 4$ heads, hidden width $d = 128$, per-head attention width $d_{\text{attn}} = 32$, feed-forward width $d_{\text{ff}} = 512$, dropout 0.1, learned

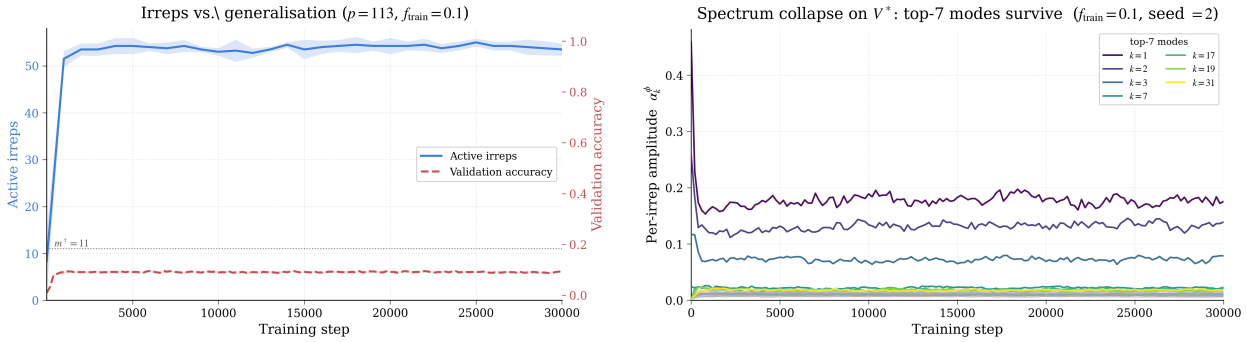
token and positional embeddings, and a tied-vocabulary linear readout $W_L \in \mathbb{R}^{p \times d}$. Maximum sequence length is 5. This is the same architecture used by Power et al. (2022) and matches the re-implementation in (Ruppik et al., 2025).

Optimisation. All models are trained with AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 10^{-6}$), constant learning rate $\eta = 10^{-3}$ after 50 steps of linear warmup, gradient-norm clipping at 1.0, batch size 512, and weight decay $\lambda_{\text{wd}} = 0.01$ unless otherwise specified. We train for $T = 30{,}000$ steps, which is enough to reach $> 99\%$ validation accuracy in every condition where grokking occurs. Validation and geometry metrics are evaluated every 200 steps.

Reproducibility. Each condition is repeated for 5 seeds; we report the mean and a ± 1 s.d. band across seeds (shaded region in all curves on our graphs). All experiments fit on a single Nvidia A100. The code is available on Github.

D Additional Experiments

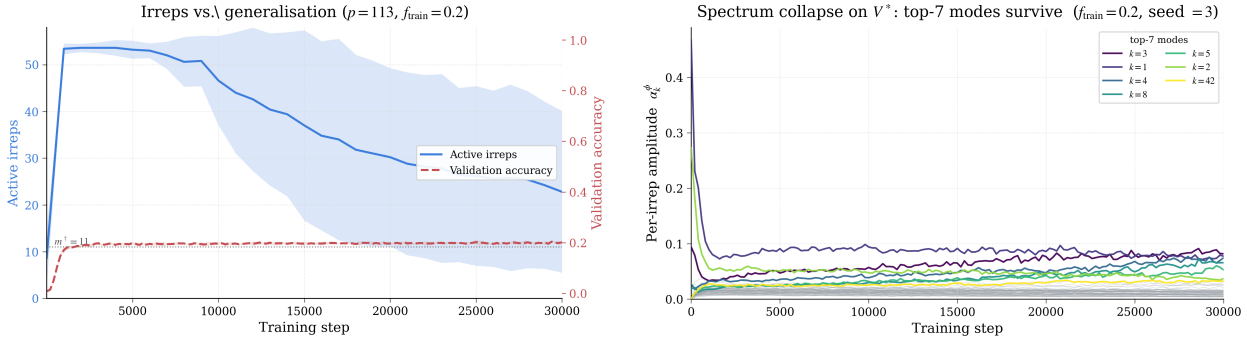
D.1 Experiment 1



(a) Active irreps

(b) Spectrum collapse

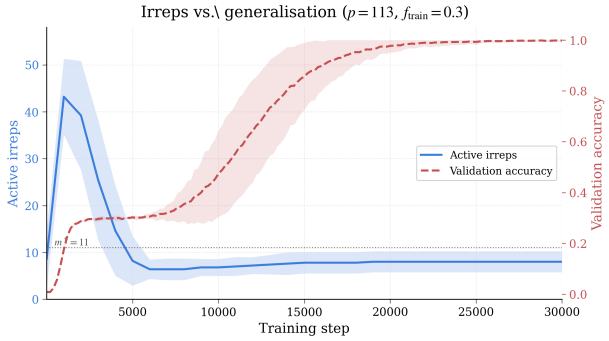
Experiment 1 at training fraction 0.1.



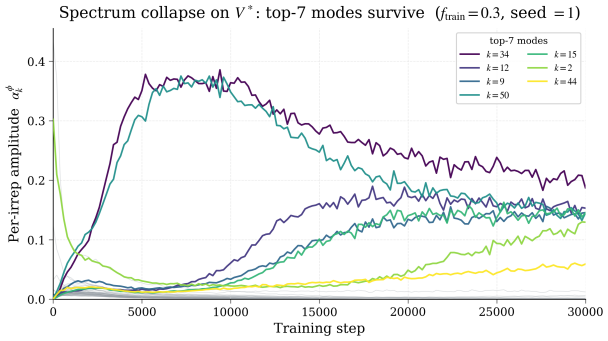
(a) Active irreps

(b) Spectrum collapse

Experiment 1 at training fraction 0.2.

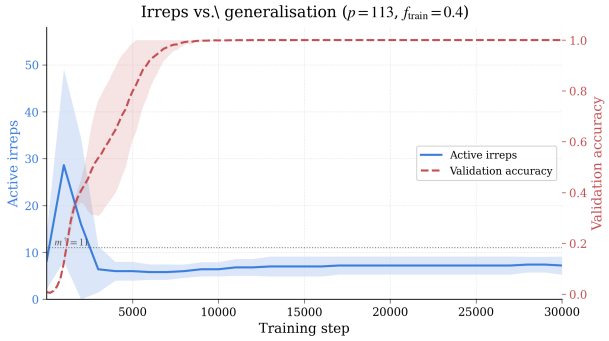


(a) Active irreps

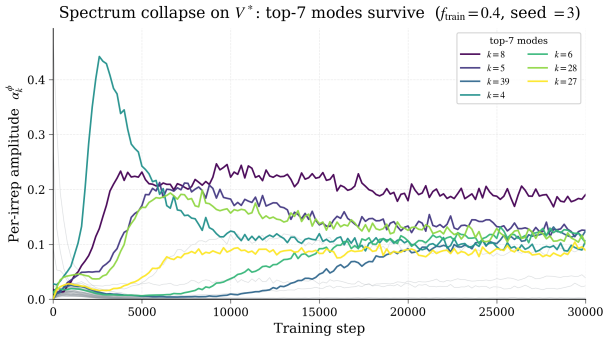


(b) Spectrum collapse

Experiment 1 at training fraction 0.3.

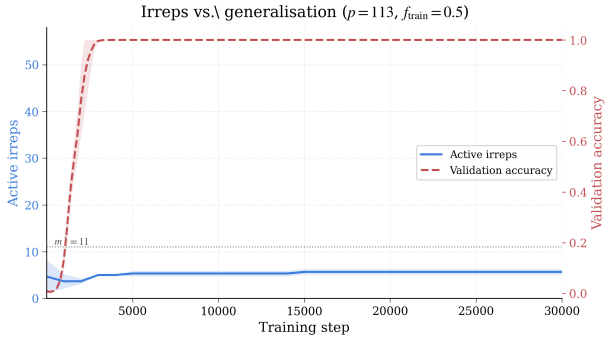


(a) Active irreps

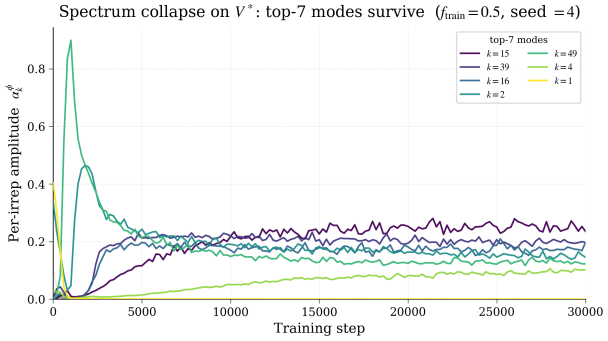


(b) Spectrum collapse

Experiment 1 at training fraction 0.4.

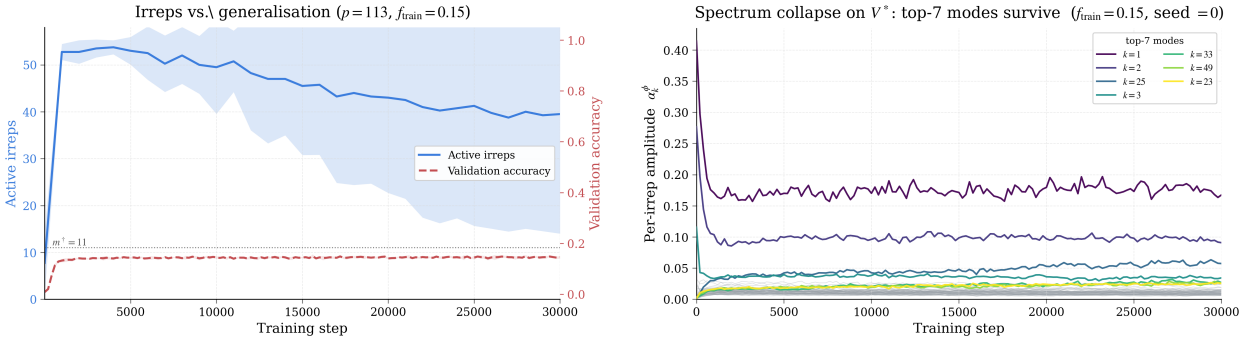


(a) Active irreps



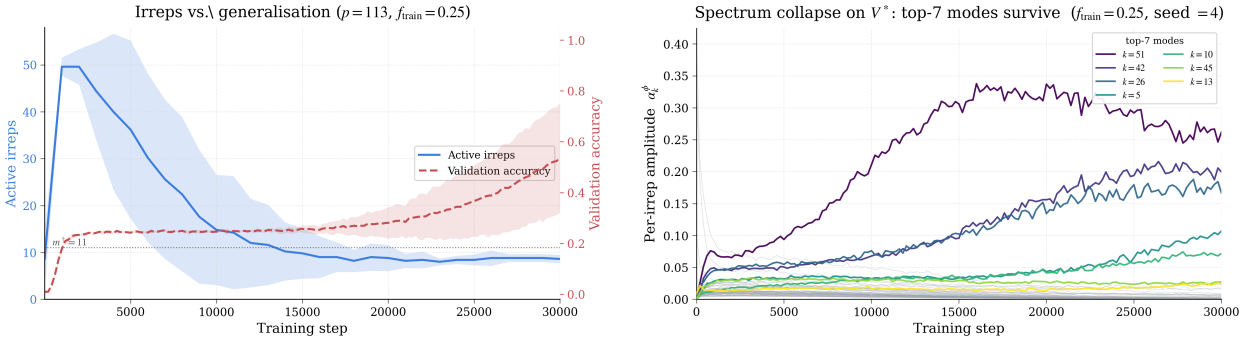
(b) Spectrum collapse

Experiment 1 at training fraction 0.5.



(a) Active irreps (b) Spectrum collapse

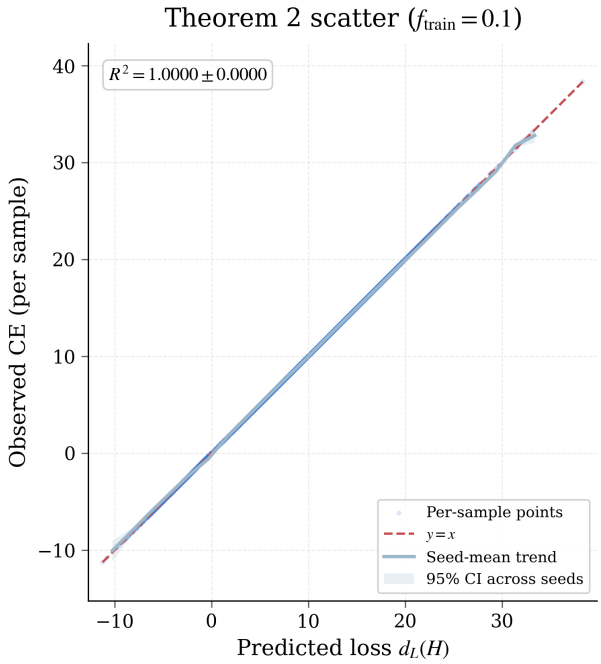
Experiment 1 at training fraction 0.15.



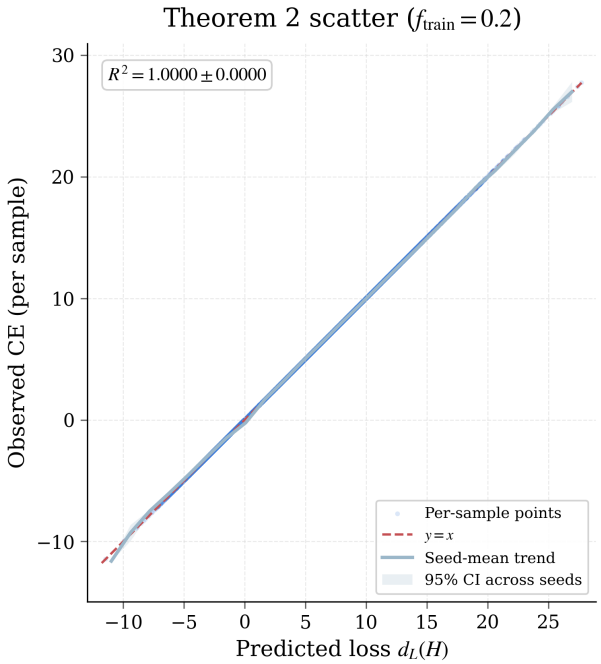
(a) Active irreps (b) Spectrum collapse

Experiment 1 at training fraction 0.25.

D.2 Experiment 2

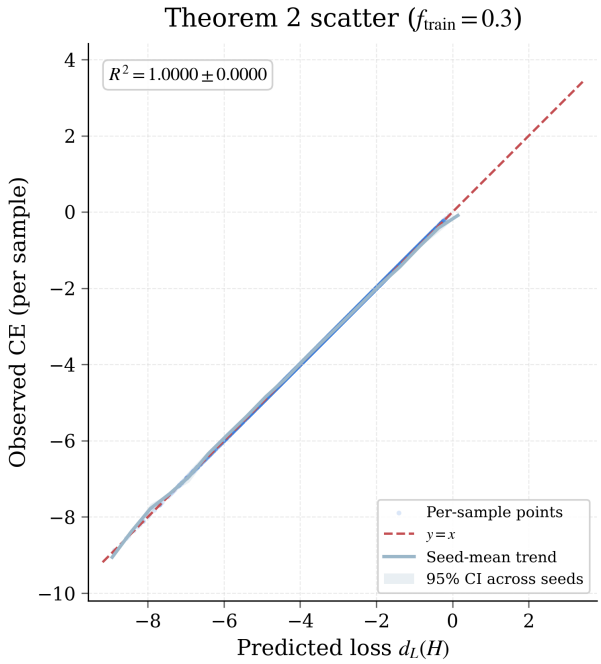


(a) Training fraction 0.1

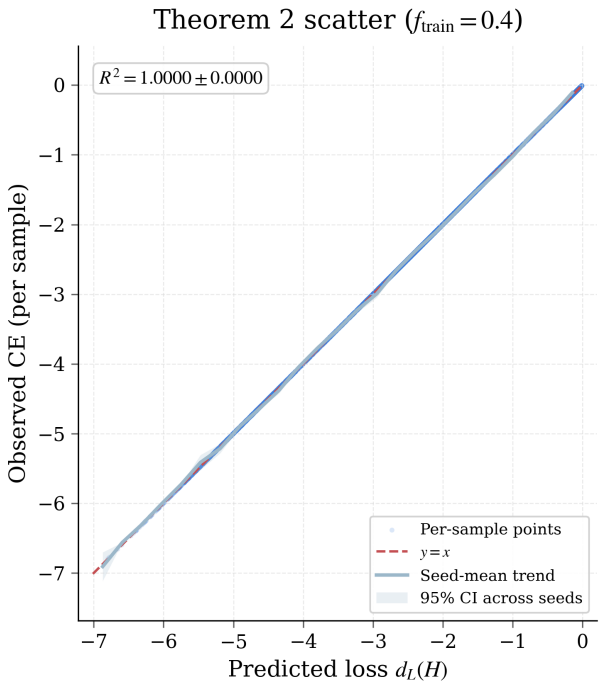


(b) Training fraction 0.2

Experiment 2 scatter plots with confidence intervals.

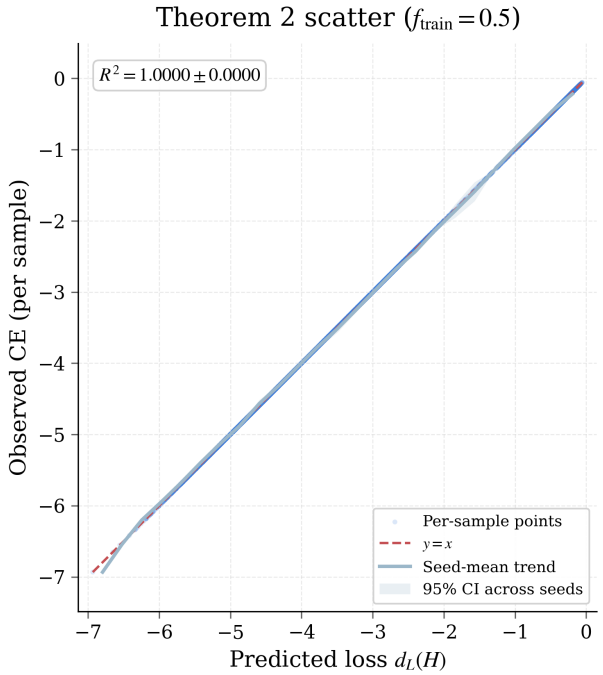


(a) Training fraction 0.3

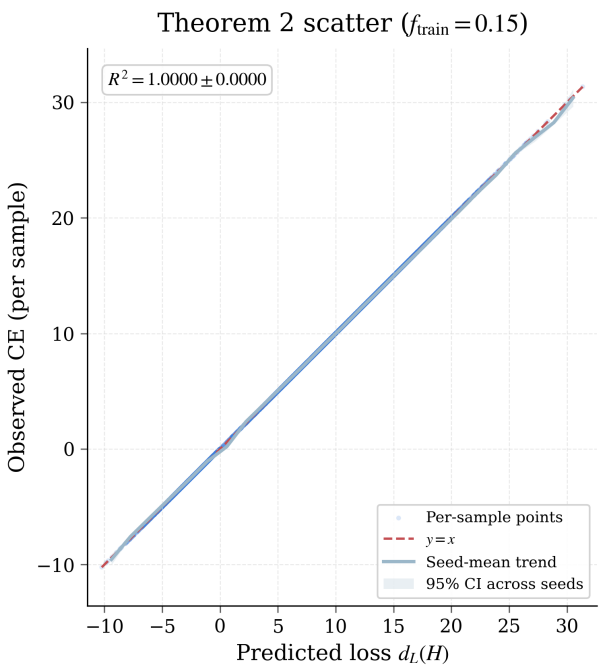


(b) Training fraction 0.4

Experiment 2 scatter plots with confidence intervals.

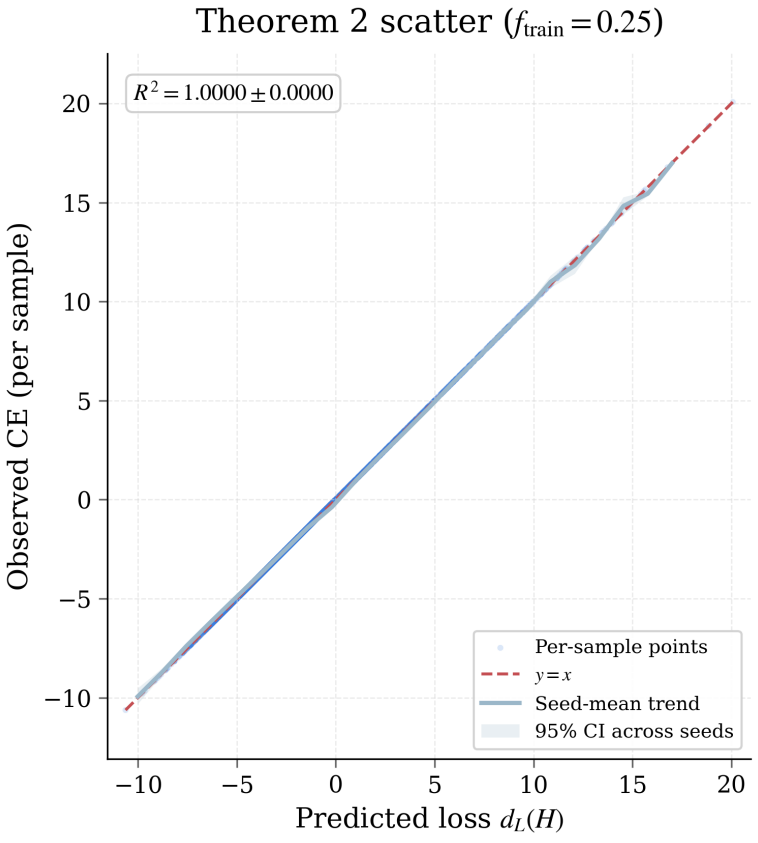


(a) Training fraction 0.5



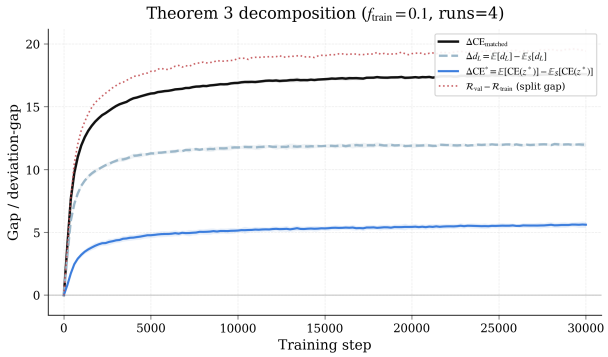
(b) Training fraction 0.15

Experiment 2 scatter plots with confidence intervals.

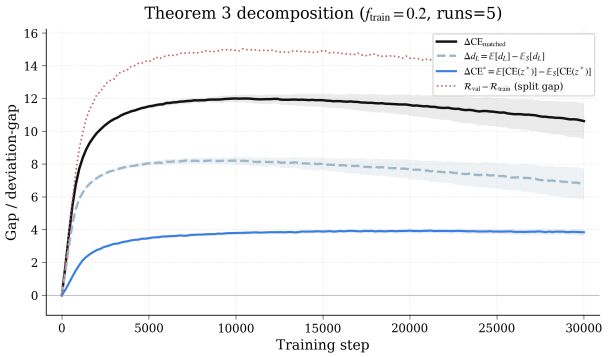


Experiment 2 scatter plot with confidence intervals at training fraction 0.25.

D.3 Experiment 3

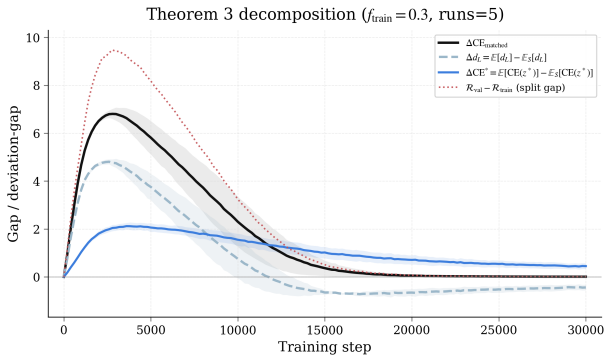


(a) Training fraction 0.1

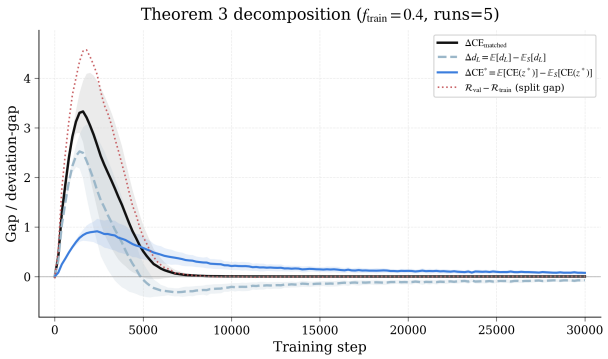


(b) Training fraction 0.2

Theorem 3 decomposition plots for Experiment 3.

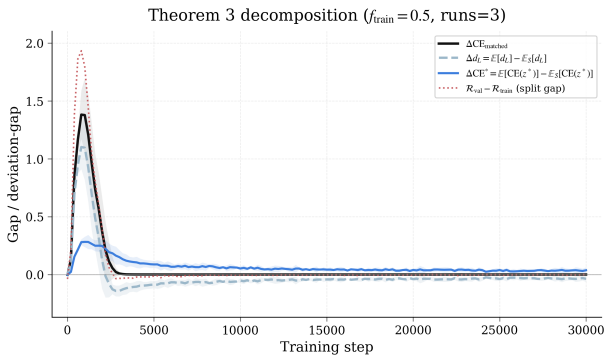


(a) Training fraction 0.3

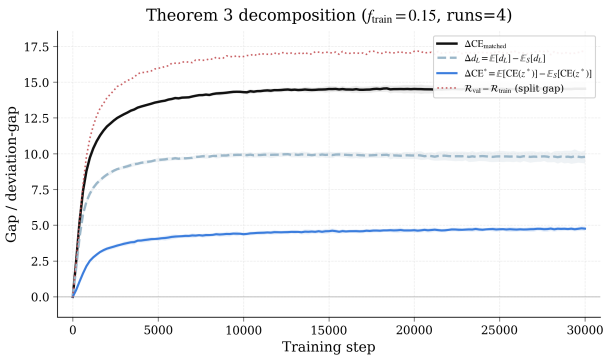


(b) Training fraction 0.4

Theorem 3 decomposition plots for Experiment 3.

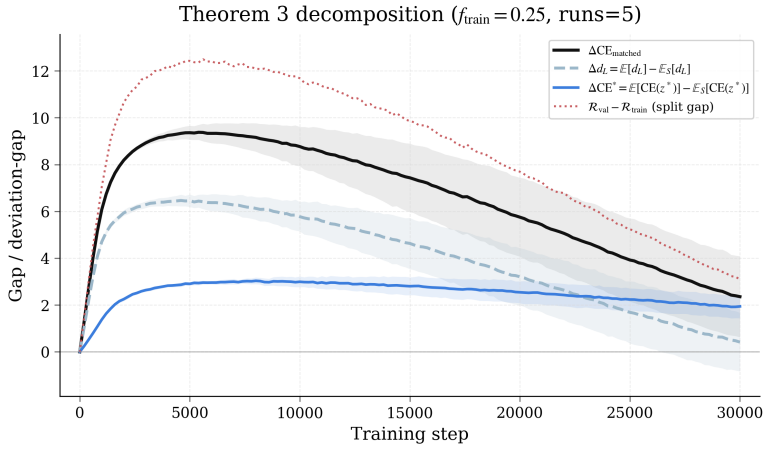


(a) Training fraction 0.5



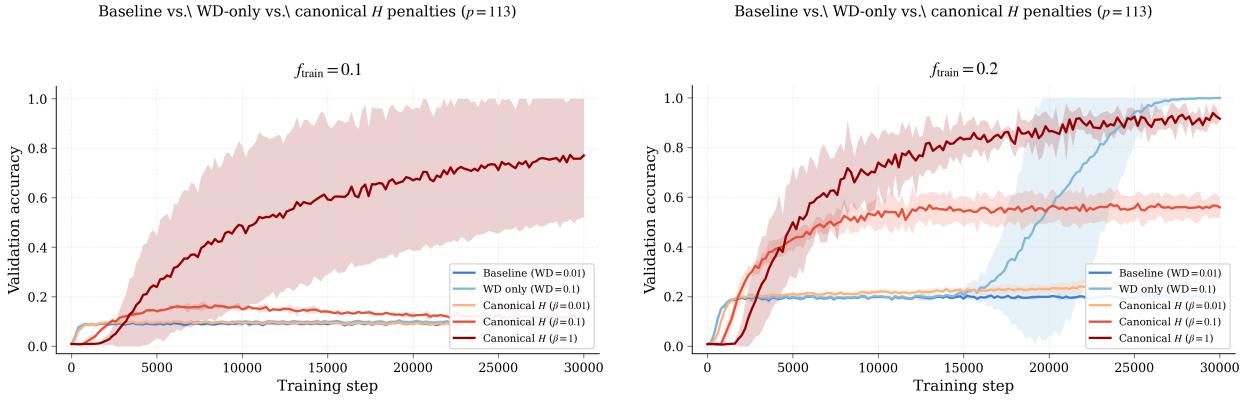
(b) Training fraction 0.15

Theorem 3 decomposition plots for Experiment 3.



Theorem 3 decomposition for Experiment 3 at training fraction 0.25.

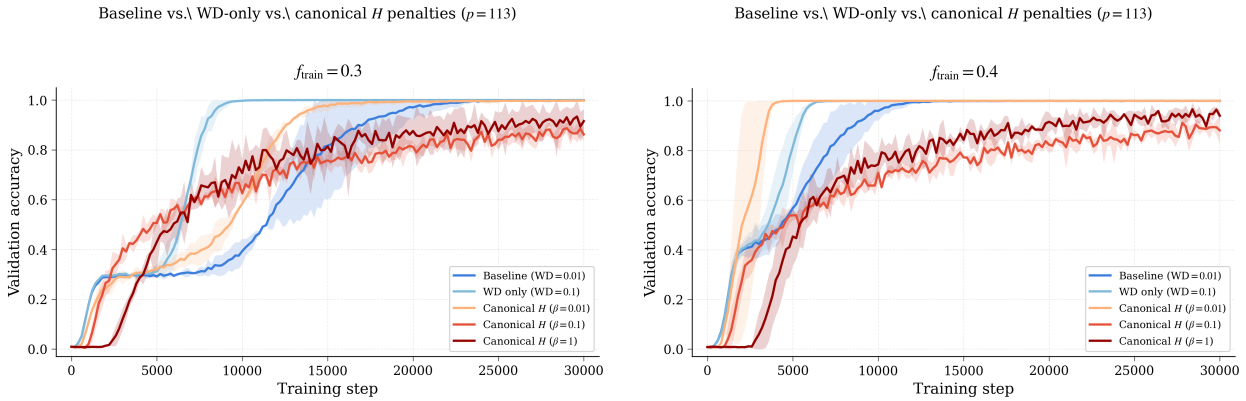
D.4 Experiment 4



(a) Training fraction 0.1

(b) Training fraction 0.2

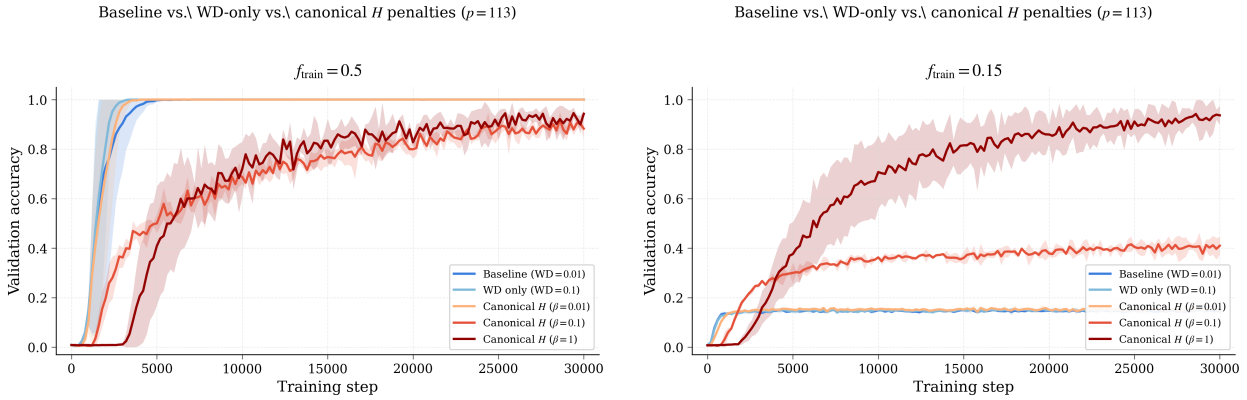
Regularizer comparison plots for Experiment 4.



(a) Training fraction 0.3

(b) Training fraction 0.4

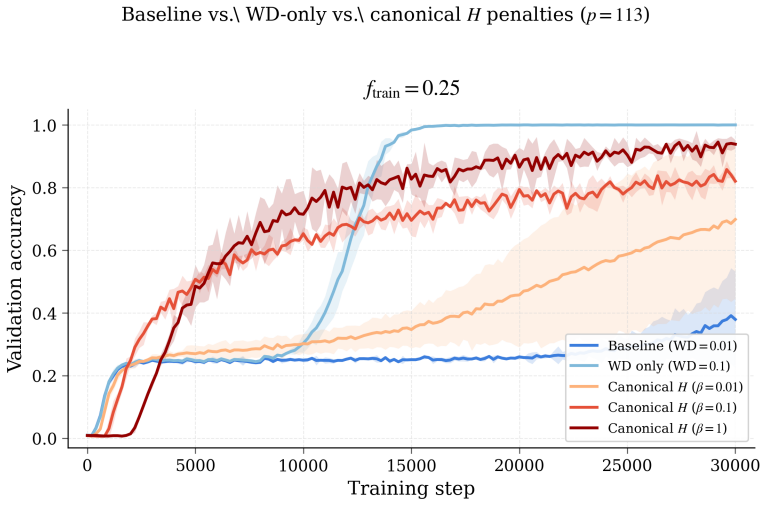
Regularizer comparison plots for Experiment 4.



(a) Training fraction 0.5

(b) Training fraction 0.15

Regularizer comparison plots for Experiment 4.



Regularizer comparison for Experiment 4 at training fraction 0.25.