Tracking Training Phases in Compositional Learning with Task-Agnostic Measures

Niclas Dern [*1] **Selma Mazioud** [*1] **Jakob Heiss** [1] **Avrajit Ghosh** [2] **Curtis James McDonald** [2] **Gabriel Clara** [2] **Bin Yu** [1 2]

Abstract

Deep neural networks often acquire their final capabilities through qualitatively distinct training phases. Characterizing these phases sheds light on how models learn and could enable steering away from unwanted outcomes. The most reliable existing methods for detecting training phases rely on a prior mechanistic understanding of how the model performs an underlying task. Task-agnostic scalar measures, i.e., quantities computed from a model's parameters, representations, or outputs, offer a more general alternative, but have largely been studied in isolation. In this paper, we systematically compare 53 such measures by fitting Gaussian hidden Markov models (HMMs) to their trajectories across two compositional settings: modular addition and a new multilingual variant we introduce, in which per-language data fractions control how much consecutive phase transitions overlap. This overlapping regime arises naturally when models acquire capabilities in close succession, yet lacks controlled benchmarks. We find that once transitions overlap, recovery quality drops across all 53 measures, even when we fit the HMM directly to validation accuracy, hinting at limitations of our HMM-based framework. Still, some measures perform relatively better, with prediction entropy, PCA effective dimension, and the Local Learning Coefficient most consistently among the top performers. [1]

[*]Equal contribution [1]Department of Statistics, University of California, Berkeley [2]Department of Electrical Engineering and Computer Science, University of California, Berkeley. Correspondence to: Niclas Dern <niclas.dern@berkeley.edu>.

Accepted to the 2nd Workshop on Compositional Learning at ICML 2026, Seoul, South Korea. Copyright 2026 by the author(s).

[1]Our code is available here.

1. Introduction

Modern deep learning systems undergo qualitatively distinct stages during training, as internal structures form, reorganize, and give rise to new capabilities and behaviors both within and beyond their training distribution (e.g., Olsson et al., 2022; McGrath et al., 2022; Chen et al., 2023a; Elhage et al., 2022). Transitions between these stages can be abrupt and have serious downstream consequences, such as a sudden shift toward broadly misaligned behavior (MacDiarmid et al., 2025). Identifying and interpreting the stages through which models pass during training enables detection of unwanted transitions, steering towards safer outcomes, and provides a deeper understanding of the final model (Nanda et al., 2023; Hoogland et al., 2024).

Compositionality, meaning the ability to systematically combine learned concepts and rules to handle novel inputs, is fundamental to language and cognition and underlies many of the desirable capabilities of intelligent systems (Lake et al., 2017; Hupkes et al., 2020; Lake & Baroni, 2018). Compositional tasks yield a particularly interesting testbed for studying training phases, because the modular structure of compositional data invites the hypothesis that concepts and rules are acquired in distinct stages (Lubana et al., 2024). Prior work supports this hypothesis, documenting phase-like development across compositional settings such as grokking on modular arithmetic (Power et al., 2022; Nanda et al., 2023; Chughtai et al., 2023) and the acquisition of digit-level subskills in multi-digit arithmetic (Kangaslahti et al., 2025; Zhao et al., 2026).

Training and validation loss trajectories often evolve smoothly across major model internal reorganizations and are consequently unreliable in predicting phase transitions (Barak et al., 2022; Michaud et al., 2023). One way of addressing this gap lies in defining a target capability or internal mechanism in advance and tracking its development during training. Two recent examples are the Fourier-based analysis of grokking in modular addition by Nanda et al. (2023) and the emergence of in-context-learning via induction heads in Olsson et al. (2022). However, this approach requires prior knowledge of the internal mechanism corresponding to an underlying task, so it cannot predict

unanticipated developmental phases.

A more general approach is to track *task-agnostic scalar measures* (quantities derived from the parameters, loss landscape, internal representations, or outputs of a model) and read phase structure off their trajectories without committing to a specific target in advance. Several such measures have been applied to track structural emergence during training, including the local learning coefficient (LLC) for stagewise development in small transformers (Lau et al., 2023; Chen et al., 2023b; Hoogland et al., 2024), the intrinsic dimension of internal representations (Lee et al., 2025), and neural collapse statistics for representational reorganization (Papyan et al., 2020). Further, Gaussian hidden Markov models (HMMs) fit to weight statistics have been shown to recover discrete training phases (Hu et al., 2023). Prior work has compared small sets of progress measures on grokking tasks (Golechha, 2024; Prakash & Martin, 2025), but to the best of our knowledge no prior work has systematically compared a broad array of task-agnostic measures on a common testbed.

Contributions. To address this gap we evaluate a large set of task-agnostic measures as tools for tracking training phases in two compositional settings, focusing on which measures can recover known phase structure and under what conditions. We make three main contributions:

First, we compile 53 task-agnostic measures from the developmental-interpretability (Hoogland et al., 2023; Chen et al., 2023b; Lau et al., 2023) and classical generalization literature (Jiang et al., 2019; McAllester, 1999; Dziugaite & Roy, 2017; Foret et al., 2020; Keskar et al., 2016; Hochreiter & Schmidhuber, 1994). We fit Gaussian HMMs to each measure's trajectory together with its first-order differences to estimate training phases (Section 3.3, Figure 1 *left*).

Second, we introduce *multilingual modular addition*, a new compositional setting in which a single transformer learns the modular addition task across multiple disjoint vocabularies, and generalizes on each one at a different point in training (Section 3.1). By varying the number of languages (we use both two- and three-language variants) and per-language data fractions, we can control the number and temporal separation of phase transitions, allowing us to stress-test phase estimation. We evaluate it alongside *modular addition* (Power et al., 2022), and assess the resulting phase estimations along three axes (Section 4, Figure 1 *right*): qualitative alignment with known transitions, predictive power for transition timing, and consistency across random seeds.

Third, we find that recovery quality depends on both the structure of the underlying transitions and the choice of measure (Section 5). On modular addition, the best measures reliably recover the known phase structure. In the two multilingual variants we evaluate, performance remains high in the two-language variant (with well-separated transitions), but degrades in the three-language variant (with overlapping transitions). Even the per-language validation-accuracy reference shows the same pattern, hinting at limitations of our HMM-based framework itself. Simple combinations of measures into multivariate HMMs do not rescue performance. Even so, a small group of measures (prediction entropy, LLC, PCA effective dimension) stays near the top across all our settings, and the persistence of this ordering raises new questions about what properties make a measure robust to changes in phase structure.

2. Related Work

Training dynamics and phase transitions. There is growing evidence that neural network training proceeds through qualitatively distinct phases, each with different model behaviors and internal structure (Olsson et al., 2022; Elhage et al., 2022; McGrath et al., 2022; Saxe et al., 2013). Taking inspiration from developmental biology, the effort to understand when and how internal algorithms and representational geometries emerge over training has been termed *developmental interpretability* (Hoogland et al., 2023). Phases manifest both behaviorally and internally, and the two are likely connected. At the behavioral level, stagewise development often appears as extended plateaus broken by rapid capability jumps, observed across settings, from deep linear networks and sparse parity learning (Saxe et al., 2013; Barak et al., 2022) to grokking in modular addition (Power et al., 2022; Nanda et al., 2023; DeMoss et al., 2025), multi-task variants with staggered transitions (Xu, 2026), formal language acquisition (Lubana et al., 2024; Chen et al., 2023a), and misaligned behavior (MacDiarmid et al., 2025). Internal reorganizations have similarly been documented, though not always one-to-one with the behavioral shifts. Examples include the formation of induction circuits for in-context learning (Olsson et al., 2022), stepwise acquisition of vocabulary and syntax (Chen et al., 2023a), staged emergence of factual recall (Zucchet et al., 2025), and phased acquisition of human-like concepts in game-playing agents (McGrath et al., 2022). While some apparent jumps may be artifacts of discontinuous evaluation metrics (Schaeffer et al., 2023), the broader picture motivates quantitative tools that can track developmental structure continuously, without needing to know in advance what mechanism or capability to look for. Our work builds on Hu et al. (2023), who fit Gaussian HMMs to weight statistics to recover latent training-state diagrams; we extend their methodology to a broader measure suite and richer phase structure.

Task-agnostic measures. A task-agnostic scalar measure is a quantity computed from a model's parameters, representations, or outputs that does not depend on the specific

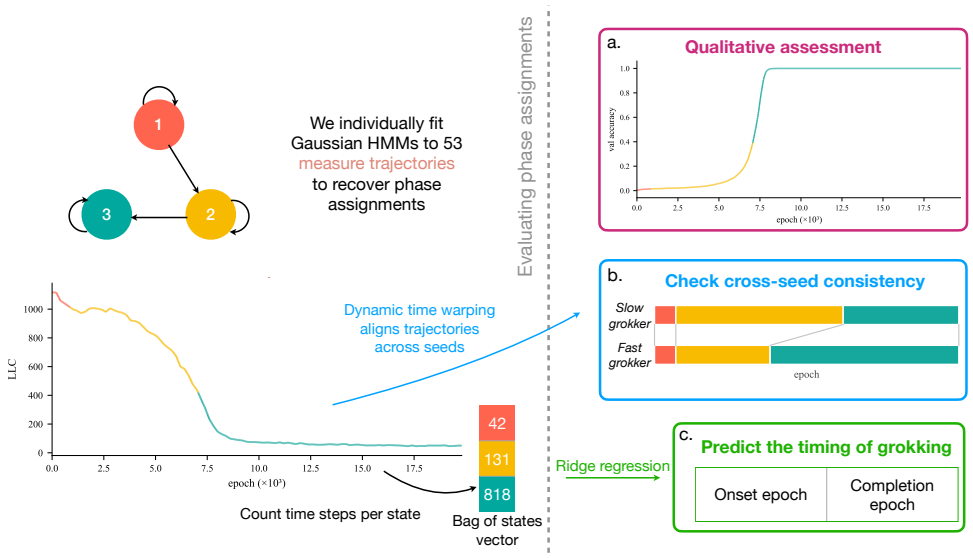

Figure 1. **Overview of our approach.** *(Left)* For each measure, we fit a Gaussian HMM to its trajectory across training runs to recover a phase estimation for each run. *(Right)* We evaluate each set of estimations along three axes: *a.* qualitative alignment with known transitions, *b.* cross-seed consistency, i.e. agreement between phase sequences across seeds, computed as the Adjusted Rand Index (ARI) on sequences aligned with Dynamic Time Warping (DTW), and *c.* retrospective predictability, i.e. how well the time spent in each phase predicts grokking timing on held-out seeds.

task the model is trained on. Such measures have been proposed both as predictors of generalization and as tools for studying training dynamics. Jiang et al. (2019) benchmarked over 40 of the former by their relationship to the generalization gap across thousands of trained convolutional neural networks, and we draw several of the top-ranked measures from their benchmark. We group these into four classes. Parameter-space measures track weight norms and distance from initialization (Neyshabur et al., 2015); loss-landscape measures characterize the geometry of the loss surface around the current parameters, e.g. Hessian-based sharpness (Keskar et al., 2016) and the local learning coefficient (LLC), which estimates the effective dimensionality of the loss basin via Singular Learning Theory (Lau et al., 2023; Watanabe, 2009); representation-space measures track activation geometry, such as intrinsic dimensionality (Ansuini et al., 2019; Facco et al., 2017; Cheng et al., 2023; Lee et al., 2025) and neural collapse (Papyan et al., 2020); output-space measures track properties of model predictions, such as margin-normalized norm bounds, prediction entropy, and prediction churn (Bartlett et al., 2017; Jiang et al., 2019; Milani Fard et al., 2016).

Several of these measures have been applied to track structural emergence during training. The LLC has been used to detect stagewise development in small transformers, identifying interpretable stages such as bigram learning and induction head formation (Hoogland et al., 2024). Other measures have surfaced phase structure tied to specific developmental phenomena, including a complexity transition that precedes the grokking generalization jump (DeMoss et al., 2025), geometric phase transitions correlated with the acquisition of compositional structure (Lee et al., 2025), and latent training-state diagrams recovered from weight-statistics HMMs (Hu et al., 2023).

Compositionality. Compositionality is the property by which a finite set of concepts and rules can combine to produce a vast range of outputs. It underlies both natural language and systematic generalization in learned systems (Fodor & Pylyshyn, 1988; Lake et al., 2017; Lake & Baroni, 2018). Compositional tasks are a particularly useful testbed for studying training phases, because their modular structure invites the hypothesis that concepts and rules are acquired in distinct stages (Lubana et al., 2024), and compositional structure has been shown to drive sudden capability acquisition (Okawa et al., 2023; Arora & Goyal, 2023; Hoffmann et al., 2023). Modular arithmetic is a tractable instance of this, where the task requires learning discrete numerical representations and a specific algebraic combination rule, making the transition from memorization to generalization sharp and interpretable (Power et al., 2022; Nanda et al., 2023; Chughtai et al., 2023).

3. Setup

3.1. Settings

We test our approach on two compositional settings, both centered on modular arithmetic and chosen for their clear phase transitions. The first is a canonical grokking task, and the second is a multilingual variant we introduce in this

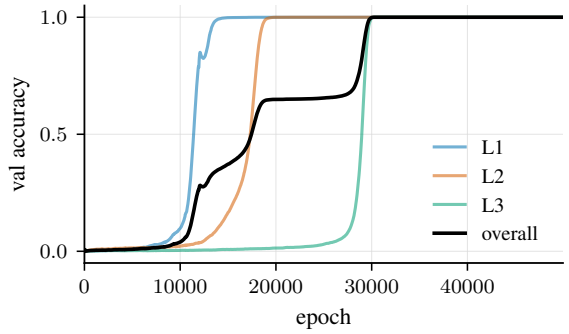

Figure 2. **Staggered grokking in multilingual modular addition with three languages.** Per-language validation accuracy (colored) and overall (black). The languages grok sequentially in order of their training-data fractions (30%, 27%, 21% for L1, L2, L3). The two-language variant drops L2, leaving the better-separated L1 and L3.

work. We train transformer models on each, with further training details in Appendix B.

Modular addition. Our first setting is modular addition, in which the model receives a three-token input $(a, b, =)$ with $a, b \in \{0, \ldots, P-1\}$ and P prime and predicts the token for $a + b \bmod P$. Power et al. first observed *grokking* in this setting, where the model generalizes abruptly long after having achieved perfect training accuracy. Following Nanda et al. (2023), we use $P = 113$, 30% of the P^2 input pairs for training, and full-batch AdamW. We train a one-layer transformer with embedding dimension 128, 4 attention heads, ReLU activations, and no layer normalization, across 40 seeds.

Multilingual modular addition. To stress-test phase estimation under richer phase structure than the single generalization transition of modular addition, we introduce *multilingual modular addition*. Inspired by multilingual language models, which process shared concepts across different and only partially overlapping vocabularies (Wendler et al., 2024; Schäfer et al., 2024; Lindsey et al., 2025; Dumas et al., 2025; Brinkmann et al., 2025) (see Appendix E.1), we train a single transformer on the same modular addition task expressed in multiple disjoint token sets, each referred to as a *language*. By varying the number of languages (we use both two- and three-language variants) and per-language data fractions, we control the number as well as the temporal separation of phase transitions. More precisely, each language samples its training set independently from the full P^2 grid, at staggered fractions (30%, 27%, and 21% in the three-language variant), which causes the languages to grok sequentially, in order of their data fractions (Figure 2). The two-language variant uses only the first and third languages, giving better-separated per-language transitions. We use the same architecture and optimizer as in modular addition, with 40 seeds.

3.2. Measure suite

A *task-agnostic scalar measure* is a scalar quantity computed from the model's parameters, representations, or outputs that does not depend on the tasks the model is trained or evaluated on. Importantly, no measure in our suite uses validation labels, and remain usable in settings where labeled validation sets for the target capability are not readily available. We focus on measures that have either been used to study training dynamics or have been shown to capture properties of trained models that relate to generalization, and draw from two bodies of literature accordingly.

Our suite draws from two bodies of literature. From the *generalization*-measures literature, we retain most top-ranked measures from the benchmark of Jiang et al. (2019), spanning PAC-Bayes flatness, margin-normalized norm bounds, weight-matrix norms, path norm, and Hessian-based sharpness. From the *training dynamics* and *developmental interpretability* literature, we include the Local Learning Coefficient (LLC, an estimate of loss-basin dimensionality) (Lau et al., 2023), TwoNN and PCA-based intrinsic dimensionality of last-layer activations (Lee et al., 2025; Facco et al., 2017), neural collapse statistics (Papyan et al., 2020), prediction churn (Milani Fard et al., 2016), prediction entropy (the entropy of softmax outputs), and spectral entropy and lossy compression complexity (DeMoss et al., 2025). We additionally include the 14 weight statistics of Hu et al. (2023). Prediction entropy, prediction churn, and the intrinsic-dimensionality estimators take validation inputs, though not labels. To enable analysis of whether different measure classes capture distinct aspects of phase structure, we organize the suite into four classes (Table 1). *Parameter-space* measures are derived from weights alone, *loss-landscape* measures from the geometry of the loss surface (requiring both weights and data), *representation-space* measures from internal activations, and *output-space* measures from model predictions. The full list of measures is given in Appendix A.

For each training run s (which differ only in seed) and measure m, we record a trajectory

$$\boldsymbol{\mu}_m^{(s)} = \left(\mu_m^{(s,1)}, \ldots, \mu_m^{(s,T)}\right) \in \mathbb{R}^T \tag{1}$$

over a common T-step grid of checkpoints, where $\mu_m^{(s,t)}$ is the value of m at checkpoint t in run s. These trajectories serve as input to the phase-assignment procedure described below.

3.3. HMM-based phase estimation

We estimate phases by fitting a Gaussian HMM to per-run measure trajectories, following Hu et al. (2023), and, as

Table 1. **We organize our measure suite into four classes.** Each class is defined by which aspect of the model the measure draws on. Full definitions are in Appendix A.

Class	#	Drawn from	Examples
Parameter-space	24	parameter values	param. norm, weight norms, spectral entropy, path norm
Loss-landscape	11	loss-surface geometry	LLC, Hessian sharpness, PAC-Bayes flatness
Representation-space	4	hidden activations	PCA / TwoNN intrinsic dim., neural-collapse stats
Output-space	14	model outputs	margin-based, prediction churn, prediction entropy

a new addition, include the trajectories' first-order differences as features. Concretely, given a measure subset $\mathcal{M} = \{m_1, \ldots, m_M\}$, we stack the trajectories $\boldsymbol{\mu}_{m_1}^{(s)}, \ldots, \boldsymbol{\mu}_{m_M}^{(s)}$ together with their consecutive differences $\Delta\boldsymbol{\mu}_{m_i}^{(s,t)} = \boldsymbol{\mu}_{m_i}^{(s,t)} - \boldsymbol{\mu}_{m_i}^{(s,t-1)}$ into a feature matrix $\boldsymbol{X}^{(s)} \in \mathbb{R}^{2M \times T}$ for each run s. Including the differences gives the HMM access to rate-of-change information and yields qualitatively cleaner phases across our settings. We z-score each row within the run to normalize location and scale, and pool the standardized sequences across runs. We fit a Gaussian HMM with full covariance to the pooled sequences via the Baum–Welch algorithm with 10 random restarts, holding out 20% of runs (matching the setup of Hu et al. (2023)). For each subset we sweep the number of hidden states $K \in \{2, \ldots, 7\}$ and select K^* by an elbow rule on held-out log-likelihood (full criterion in Appendix C.2). In most analyses we instead fix K at a stated value to keep results comparable across subsets, specifying the choice in each experiment. We decode each run's most likely latent state sequence from the fitted model via the Viterbi algorithm, producing the phase estimations used in the analyses that follow (Section 4 and Section 5).

4. Evaluating phase estimations

To compare task-agnostic measures in our compositional settings, we need a way to evaluate the quality of their phase estimations. We propose three complementary methods, described below, with further details in Appendix C.

References. We compare each measure's phase estimation against four references fit with the same protocol of Section 3.3: (i) an equal-K partition that splits training into K contiguous segments of equal length, as a trivial structural floor; (ii) a HMM fit on validation accuracy, a strong behavioral reference since validation accuracy directly reflects the grokking transitions our measures should track (a reference that is only available because our setting is simple enough to admit clean behavioral labels); (iii) an HMM fit on training and validation accuracy together; and (iv) in the multilingual setting, an HMM fit on the three per-language validation accuracies. References (iii) and (iv) test whether richer behavioral signals add information over (ii). All HMM references additionally include the trajectories' differences as features, matching our setup.

Qualitative assessment. We begin by visually inspecting the trajectories of individual measures over training and overlay each measure's inferred Viterbi sequence on the corresponding behavioral curves (train and validation accuracy, shown per language in multilingual modular addition), with each timestep colored by its inferred state. This allows us to qualitatively assess whether the inferred phases correspond to identifiable transitions in model behavior.

Information encoded in the phase structure. To quantify how well the phase structure tracks compositional development, we ask how much behavioral information is carried by the bag-of-states feature $\boldsymbol{c}^{(s)} = \left(c_1^{(s)}, \ldots, c_K^{(s)}\right)$ with $c_k^{(s)} = \left|\{t : z_t^{(s)} = k\}\right|$ (also used by Hu et al. (2023)), where $z_t^{(s)}$ is the Viterbi state of run s at timestep t and k is the state. This summary statistic captures how much time each run spends in each phase, without reference to when. We evaluate how much behavioral information $\boldsymbol{c}^{(s)}$ carries by linearly predicting behavioral targets on held-out runs with 5-fold cross-validated Ridge R^2. The targets are the onset (first epoch where validation accuracy exceeds 0.1) and completion (first epoch where validation accuracy exceeds 0.99) of grokking. In multilingual modular addition these targets have per-language variants. We verify the targets are non-degenerate across seeds and perform robustness checks across the accuracy threshold and the number of HMM states K, following the PCS framework (Yu, 2020).

Consistency across seeds. We furthermore test whether the inferred phases carry a *consistent behavioral meaning* across seeds. Concretely, we ask whether runs at the same level of compositional capability, as proxied by validation accuracy, land in the same inferred phase. Note that consistency in this sense is neither necessary nor sufficient for a good phase estimation. A measure can recover meaningful phases that vary in identity across runs, and conversely a single long phase that always covers the same training window would be highly consistent without being informative. We treat the test as a useful quality signal that nevertheless has to be interpreted with the above caveats in mind.

For each pair of seeds, we align behavioral trajectories (train and validation accuracy jointly in modular addition; the

per-language validation-accuracy trajectories jointly in the multilingual settings) with dynamic time warping (DTW), giving a warp path that maps each timestep of one run to a timestep of the other. We then compute the adjusted Rand index (ARI) between the two Viterbi sequences along the warp path. Because DTW often maps several timesteps of one run to one of the other, the full warp path includes near-duplicate pairs that inflate ARI. We therefore compute ARI on N points spaced evenly along the warp path, so each comparison contributes roughly independent information. We report the mean ARI across all seed pairs and benchmark against the references.

5. Results

We ran 40 seeds per setting across our three settings and recorded the 53 measures introduced in Section 3.2. We exclude runs that fail to generalize by the end of training (in multilingual, runs where any language does not generalize), leaving 39 runs in modular addition, 26 runs in the two-language setting, and 29 runs in the three-language setting. For each setting we fit one Gaussian HMM per measure (Section 3.3) and compare against the four references described in Section 4. To keep comparisons across measures and combinations fair, we fix K per setting, using $K = 4$ in modular addition and $K = 6$ in both multilingual settings. We pick these values because they qualitatively gave the best phase estimations, are broadly consistent with the structure each setting suggests, and align with the modal K^* from an elbow rule on held-out log-likelihood (Appendix C.2).

5.1. Phase recovery across compositional settings

Qualitative recovery. In modular addition, the validation-accuracy curve naturally suggests four phases (memorization, post-memorization plateau, grokking, post-grokking plateau), which align with the mechanistic phases of Nanda et al. (2023). Our choice of $K = 4$ is not imposed to match this count: the elbow rule on held-out log-likelihood (Figure 9) selects a comparable K independently of any mechanistic prior, so the four-phase structure is corroborated by the data rather than assumed. The validation-accuracy reference recovers these phases reliably, and the train-and-validation reference stays well-aligned throughout. Several of our measures produce HMM phases that align well with this four-phase structure, including the LLC, prediction entropy, the TwoNN intrinsic dimension, several PAC-Bayes flatness measures, and the within-class variability collapse statistic (NC1). Other measures, such as the parameter norm, capture the same structure less sharply. Figure 3 (top) shows the validation-accuracy reference and the LLC as representatives.

In the three-language multilingual setting, the picture is noisier across all measures (Figure 3, bottom). The validation-accuracy references sometimes merge two languages into a single transition, collapse all three, or oscillate between states during the staggered transitions, and the per-language reference shows similar difficulties. Since the per-language reference contains the relevant transition information by construction, we take these difficulties to be at least partly a property of our HMM setup, not only of any specific measure. The same difficulty shows up for our measures, where it is also less clear which align well with the staggered transitions. The LLC, which worked well on modular addition, only loosely tracks the staggered transitions here. The two-language setting sits in between the other settings (Figure 15): phase estimations are noisier than in modular addition but cleaner than in three-language multilingual, and the measures that recover the phases are similar to those in modular addition, with the LLC, the TwoNN intrinsic dimension, and several PAC-Bayes flatness measures all aligning reasonably well with the per-language transitions.

Quantitative recovery. To compare our results quantitatively, we ask how well the bag-of-states feature predicts the timing of grokking on held-out seeds (as outlined in Section 4). The targets are onset and completion in modular addition, and per-language variants in the multilingual setting, always scored by 5-fold cross-validated Ridge R^2. Results are shown in Figure 4.

In modular addition, the completion of grokking is well predicted across many measures, while onset is substantially harder. The validation-accuracy reference scores high on onset and completion, and the train-and-validation reference is slightly weaker. The equal-K reference scores at or below zero on both targets, indicating that the HMM-inferred phases carry non-trivial signal about training progress. Multiple output-space measures (prediction churn, prediction entropy) rank relatively highly, with the LLC ranking highest among loss-landscape measures. The relative ordering is broadly stable across K choices and across variations in the onset threshold (see Appendix D.2).

In the three-language multilingual setting, the analysis is again relatively stable across K but R^2 values are generally much lower (Figure 4, right). Among the per-language targets, the onset of L2 is the hardest to predict, while the completion of L3 is predicted relatively well. The per-language validation-accuracy reference is the strongest subset at $K = 6$, though this position is unstable across K, followed by a small group (e.g., parameter norm, prediction entropy, PCA effective dimension), with the aggregate validation-accuracy reference behind them. Unlike modular addition, output-space measures do not dominate the top group. Measure rankings are weakly correlated across languages, meaning a measure that does well on L1 is not necessarily good on L2 or L3. Rankings on the L2 and L3 completion targets are a partial exception, with a Spearman

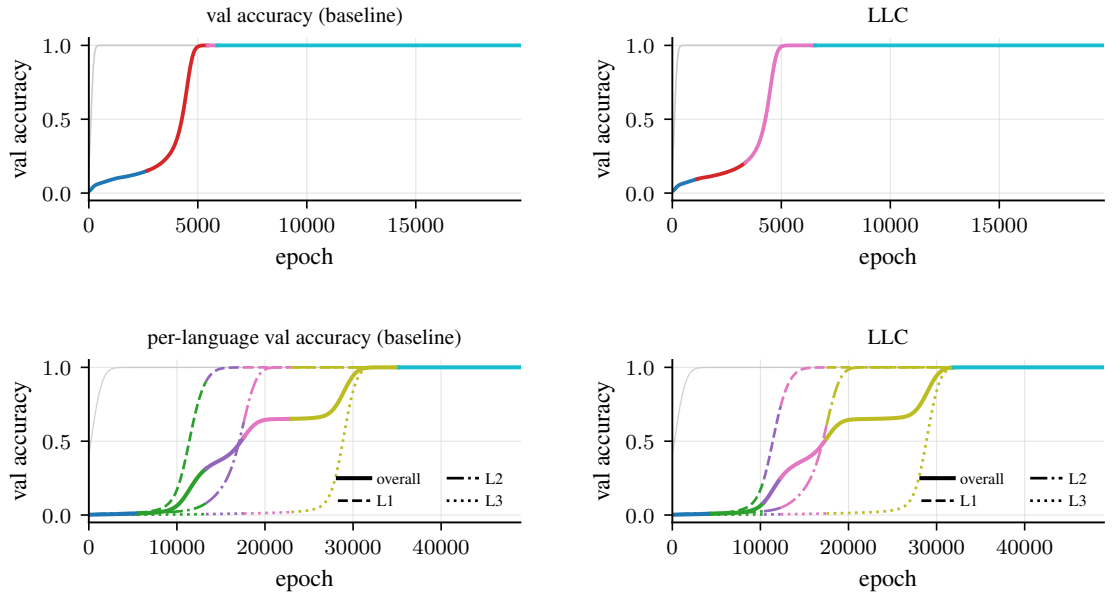

Figure 3. **Inferred phases overlaid on validation accuracy.** Both the validation-accuracy reference and the LLC recover the four phases of modular-addition training (top); in three-language multilingual (bottom), they produce an additional phase to capture moments when two languages grok at the same time. Each panel shows train (silver) and validation (color) accuracy on a representative seed, segmented by the HMM's Viterbi state; in the bottom row, validation accuracy is plotted separately for the three languages. The two-language analogue is in Figure 15.

correlation of around 0.6. In the two-language setting, top retrospective R^2 values reach modular-addition levels on average (Figure 16), substantially above three-language multilingual. The per-language validation-accuracy reference leads, with prediction entropy, PCA effective dimension, and the LLC near the top.

Cross-seed consistency. The bag-of-states analysis shows that phase structure carries timing information, but it does not test whether the same phase means the same thing across runs. Next, we ask whether two runs at the same level of compositional development land in the same inferred phase, following the consistency check described in Section 4. Our results are shown in Figure 17. The full per-subset ARI distributions are in Appendix D.3.

In the modular addition setting, the validation-accuracy reference leads the ARI ranking, with the PCA effective dimension and the train-and-validation reference also among the strongest measures. ARI values are high for top measures, suggesting that phases often carry consistent behavioral meaning across seeds. Output-space measures such as prediction churn and prediction entropy also perform well in terms of consistency. The pattern is different in the three-language multilingual setting. Neither the per-language nor the aggregate validation-accuracy reference obtains a high ARI, indicating that phase sequences differ across seeds even when using the most direct behavioral signal. Unlike in the predictive case, the two-language setting looks closer to the three-language case than to modular addition, suggesting that this difference appears once per-language overlap is introduced. One explanation is that temporal overlap of the grokking events of each language, which varies in degree across seeds, increases the variety of phases assigned during runs. Introducing more languages also expands the variety of paths, leading to different phase sequences.

The role of phase overlap. Looking across the phase-recovery results in our three settings, our explanation for the worse qualitative and quantitative performance of task-agnostic measures on three-language multilingual is overlap between per-language transitions in this setting. Besides being slightly more complex than the two-language multilingual setting, in the three-language multilingual setting the second language has 90% as much training data as the first (compared to 70% in the two-language setting), which causes the first two languages' transitions to often overlap. We conjecture this drives the worse predictive performance on these languages in the three-language setting, and the generally worse predictive performance of phases in the three-language setting compared to the two-language setting. Note that the same degradation appears for the per-language validation accuracy reference, which contains the disentangling information by construction. This indicates that our HMM-based approach itself contributes to the difficulty in the overlapping regime, alongside any limits of

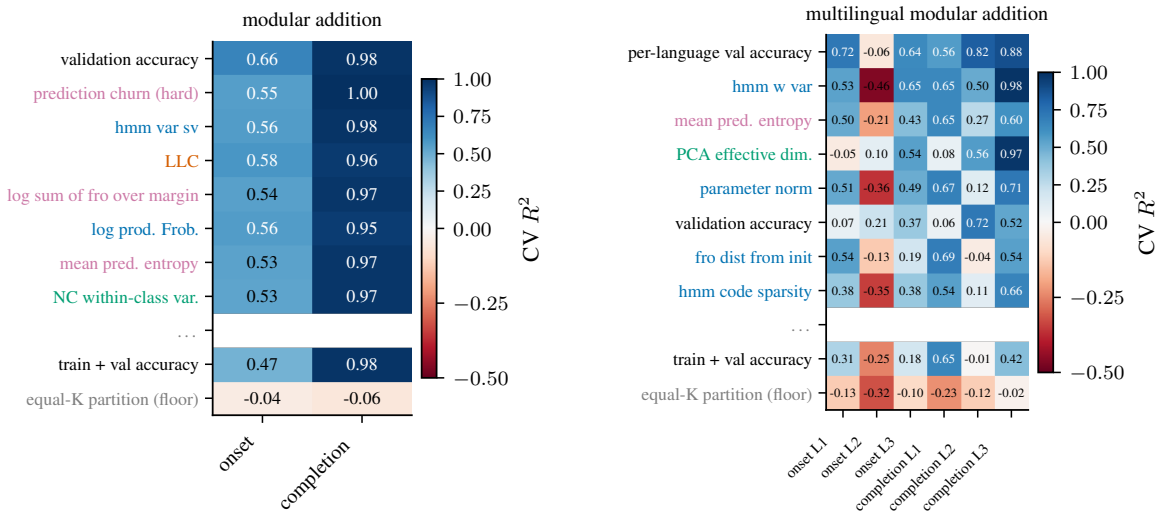

Figure 4. **Retrospective bag-of-states R^2 at fixed K.** Top-8 measures and references per setting, ranked by mean R^2 across the retrospective targets (onset and completion of grokking, per-language in multilingual). References that did not make the top-8 are shown below the '. . .' separator row for context. Cells are 5-fold CV Ridge R^2. *(Left)* On modular addition, the validation-accuracy reference tops both targets, with many measures close behind. *(Right)* On three-language multilingual, R^2 values are systematically lower, with the strongest signal on L3 completion (with L3 onset close behind) and L2 onset as the clear outlier. The two-language analogue is in Figure 16.

the measures themselves. A caveat is that our design cannot vary per-language overlap and sample count independently, because the data fractions (100%, 90%, and 70% for L1–L3; Appendix B.2) control both when a language groks and how visible its transition is in our aggregate measures, so worse recovery could in principle reflect a weaker signal rather than overlap. Two observations still point toward overlap. Within the three-language setting, difficulty does not track sample count, as L3 has the fewest training pairs (2,681) yet is the best-predicted language, while L2, whose transition overlaps L1's, is the clear outlier despite more data (3,447 pairs). Across settings, the 70% language (L2 in the two-language variant, L3 in the three-language variant) keeps the same 2,681 pairs (though a smaller share of the total) but recovers better in the two-language variant, where its transition is well separated. This points to overlap rather than data scarcity, though for L2 we cannot fully rule out a sample-count effect, since no variant contains a 90% language with a separated transition.

Despite this degradation, a small group of measures recurs near the top, though no single measure is uniformly best. Prediction entropy is the only measure that stays in the top six of every setting. The LLC ranks near the top of modular addition and the two-language variant but drops to rank 14 of 53 in the three-language case, and PCA effective dimension ranks near the top of both multilingual variants but drops to rank 20 of 53 on modular addition.

We next examine whether combining measures into a multivariate HMM can disentangle these transitions in §5.2.

5.2. Relations across measures and combinations

To check whether our measure classes correspond to genuinely different aspects of phase structure, we compute pairwise ARI between all measures' Viterbi paths and compare within-class to across-class agreement. We find that the two distributions broadly overlap in all three settings (Figure 18): our class definitions do not carve out genuinely distinct phase structure. Overall ARI is lower in the multilingual settings, consistent with the noisier phase structure found in §5.1. We next ask whether combining measures into multivariate HMMs improves on individual ones, evaluating three strategies (details in Appendix D.5). None reliably improves on the best individual measure in either setting, and combinations score no better than typical individuals from the same source pool.

6. Conclusion

We presented a systematic comparison of 53 task-agnostic scalar measures for tracking training phases in two compositional settings, going substantially beyond prior comparisons that consider only a handful of measures. Fitting Gaussian HMMs to each measure's trajectory together with its consecutive differences, we evaluated 53 measures along three axes (qualitative alignment, retrospective predictability, and cross-seed consistency) on modular addition and on two variants of the multilingual modular addition setting we

introduce. We find that recovery quality depends on both the structure of the underlying phase transitions and the choice of measure. On modular addition, the best measures reliably recover the known phase structure. In the two-language multilingual variant, designed to give well-separated per-language transitions, retrospective R^2 maintains the level of modular addition, but cross-seed consistency drops. In the three-language variant, where transitions overlap, recovery drops sharply across all tested measures. Even the per-language validation-accuracy reference shows the same drop, implicating our HMM-based framework itself. Simple combinations of measures into multivariate HMMs do not rescue performance. Some measures hold up better than others though, and a few (prediction entropy, LLC, PCA effective dimension) stay near the top across all our settings.

Limitations and future work. Our evaluation is limited to small transformers on modular arithmetic, and how phase estimation scales to larger models or more complex tasks remains an open question. We also rely on a single phase-estimation framework throughout, fitting Gaussian HMMs to low-dimensional trajectories with a relatively low number of latent states. Our three-language results suggest this framework contributes to the difficulty in the overlapping regime alongside the measures themselves, motivating alternative approaches to detect phases, e.g., more sophisticated ways to combine measures than the multivariate HMMs we explore. Beyond task-agnostic measures, we hope to see further work using multilingual modular addition as a testbed for studying training phases and specifically overlapping phase transitions.

Acknowledgements

We thank Edmund Lau, Daniel Murfet, and Jesse Hoogland for inspiring discussions. This work was primarily supported by the UK AI Security Institute (AISI) under grant AP-S2-100023, "The Geometry of Risk: Using Model Complexity to Improve Predictions of Catastrophic AI Failures." This work used Bridges-2 GPU at the Pittsburgh Supercomputing Center through allocation CIS260001 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, which is supported by U.S. National Science Foundation grants #2138259, #2138286, #2138307, #2137603, and #2138296 (Boerner et al., 2023). Jakob Heiss gratefully acknowledges support from the Swiss National Science Foundation (SNSF) Postdoc.Mobility fellowship [grant number P500PT_225356]. We gratefully acknowledge support from the NSF through FODSI (grant DMS-2023505), from the NSF and the Simons Foundation through the Collaboration on the Theoretical Foundations of Deep Learning (awards DMS-2031883 and 814639), as well as support from UC Berkeley.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Ansuini, A., Laio, A., Macke, J. H., and Zoccolan, D. Intrinsic dimension of data representations in deep neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Arora, S. and Goyal, A. A theory for emergence of complex skills in language models. *arXiv preprint arXiv:2307.15936*, 2023.

Bahdanau, D., Murty, S., Noukhovitch, M., Nguyen, T. H., de Vries, H., and Courville, A. Systematic generalization: What is required and can it be learned? In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HkezXnA9YX.

Bandarkar, L., Muller, B., Yuvraj, P., Hou, R., Singhal, N., Lv, H., and Liu, B. Layer swapping for zero-shot cross-lingual transfer in large language models, 2025. URL https://arxiv.org/abs/2410.01335.

Barak, B., Edelman, B., Goel, S., Kakade, S., Malach, E., and Zhang, C. Hidden progress in deep learning: Sgd learns parities near the computational limit. *Advances in Neural Information Processing Systems*, 35:21750–21764, 2022.

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

Boerner, T. J., Deems, S., Furlani, T. R., Knuth, S. L., and Towns, J. Access: Advancing innovation: Nsf's advanced cyberinfrastructure coordination ecosystem: Services & support. In *Practice and Experience in Advanced Research Computing 2023: Computing for the Common Good*, PEARC '23, pp. 173–176, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450399852. doi: 10.1145/3569951.3597559. URL https://doi.org/10.1145/3569951.3597559.

Brinkmann, J., Wendler, C., Bartelt, C., and Mueller, A. Large language models share representations of latent grammatical concepts across typologically diverse languages, 2025. URL https://arxiv.org/abs/2501.06346.

Caruana, R. Multitask learning. *Machine learning*, 28(1):41–75, 1997. URL https://doi.org/10.1023/A:1007379606734.

Chen, A., Shwartz-Ziv, R., Cho, K., Leavitt, M. L., and Saphra, N. Sudden drops in the loss: Syntax acquisition, phase transitions, and simplicity bias in mlms. *arXiv preprint arXiv:2309.07311*, 2023a.

Chen, Z., Lau, E., Mendel, J., Wei, S., and Murfet, D. Dynamical versus bayesian phase transitions in a toy model of superposition. *arXiv preprint arXiv:2310.06301*, 2023b.

Cheng, E., Kervadec, C., and Baroni, M. Bridging information-theoretic and geometric compression in language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12397–12420, 2023.

Chizat, L. and Bach, F. On the global convergence of gradient descent for over-parameterized models using optimal transport. *Advances in neural information processing systems*, 31, 2018. URL https://arxiv.org/abs/1805.09545.

Chughtai, B., Chan, L., and Nanda, N. A toy model of universality: Reverse engineering how networks learn group operations. In *International Conference on Machine Learning*, pp. 6243–6267. PMLR, 2023.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. Unsupervised cross-lingual representation learning at scale. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J. (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8440–8451, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.747. URL https://aclanthology.org/2020.acl-main.747/.

DeMoss, B., Sapora, S., Foerster, J., Hawes, N., and Posner, I. The complexity dynamics of grokking. *Physica D: Nonlinear Phenomena*, pp. 134859, 2025.

Dufter, P. and Schütze, H. Identifying elements essential for BERT's multilinguality. In Webber, B., Cohn, T., He, Y., and Liu, Y. (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4423–4437, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.358. URL https://aclanthology.org/2020.emnlp-main.358/.

Dumas, C., Wendler, C., Veselovsky, V., Monea, G., and West, R. Separating tongue from thought: Activation patching reveals language-agnostic concept representations in transformers. In Che, W., Nabende, J., Shutova, E., and Pilehvar, M. T. (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 31822–31841, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1536. URL https://aclanthology.org/2025.acl-long.1536/.

Dziri, N., Lu, X., Sclar, M., Li, X. L., Jiang, L., Lin, B. Y., Welleck, S., West, P., Bhagavatula, C., Le Bras, R., et al. Faith and fate: Limits of transformers on compositionality. *Advances in neural information processing systems*, 36: 70293–70332, 2023.

Dziugaite, G. K. and Roy, D. M. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008*, 2017.

Elhage, N., Hume, T., Olsson, C., Schiefer, N., Henighan, T., Kravec, S., Hatfield-Dodds, Z., Lasenby, R., Drain, D., Chen, C., et al. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, 2022.

Elmoznino, E., Jiralerspong, T., Bengio, Y., and Lajoie, G. A complexity-based theory of compositionality. *arXiv preprint arXiv:2410.14817*, 2024.

Facco, E., d'Errico, M., Rodriguez, A., and Laio, A. Estimating the intrinsic dimension of datasets by a minimal neighborhood information. *Scientific reports*, 7(1):12140, 2017.

Fodor, J. A. and Pylyshyn, Z. W. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2): 3–71, 1988.

Foret, P., Kleiner, A., Mobahi, H., and Neyshabur, B. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020.

Frankle, J. and Carbin, M. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018. URL https://arxiv.org/abs/1803.03635.

Golechha, S. Progress measures for grokking on real-world tasks. *arXiv preprint arXiv:2405.12755*, 2024.

Heiss, J. *Inductive Bias of Neural Networks and Selected Applications*. Doctoral thesis, ETH Zurich, Zurich, 2024. URL https://www.research-collection.ethz.ch/handle/20.500.11850/699241.

Heiss, J., Stockinger, A., and Teichmann, J. Reducing the number of neurons of deep ReLU networks based on the current theory of regularization, 2020. URL https://openreview.net/pdf?id=9GUTgHZgKCH.

Heiss, J., Teichmann, J., and Wutte, H. How infinitely wide neural networks can benefit from multi-task learning– an exact macroscopic characterization. *arXiv preprint arXiv:2112.15577*, 2021. URL https://arxiv.org/abs/2112.15577.

Hochreiter, S. and Schmidhuber, J. Simplifying neural nets by discovering flat minima. *Advances in neural information processing systems*, 7, 1994.

Hoffmann, D. T., Schrodi, S., Bratulić, J., Behrmann, N., Fischer, V., and Brox, T. Eureka-moments in transformers: Multi-step tasks reveal softmax induced optimization problems. *arXiv preprint arXiv:2310.12956*, 2023.

Hoogland, J., Gietelink Oldenziel, A., Murfet, D., and van Wingerden, S. Towards developmental interpretability. https://www.lesswrong.com/posts/TjaeCWvLZtEDAS5Ex/towards-developmental-interpretability, 2023.

Hoogland, J., Wang, G., Farrugia-Roberts, M., Carroll, L., Wei, S., and Murfet, D. Loss landscape degeneracy and stagewise development in transformers. *arXiv preprint arXiv:2402.02364*, 2024.

Hu, M. Y., Chen, A., Saphra, N., and Cho, K. Latent state models of training dynamics. *arXiv preprint arXiv:2308.09543*, 2023.

Hupkes, D., Dankers, V., Mul, M., and Bruni, E. Compositionality decomposed: How do neural networks generalise? *Journal of Artificial Intelligence Research*, 67: 757–795, 2020.

Hutchinson, M. F. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics-Simulation and Computation*, 18(3):1059–1076, 1989.

Jiang, Y., Neyshabur, B., Mobahi, H., Krishnan, D., and Bengio, S. Fantastic generalization measures and where to find them. *arXiv preprint arXiv:1912.02178*, 2019.

K, K., Wang, Z., Mayhew, S., and Roth, D. Cross-lingual ability of multilingual bert: An empirical study. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HJeT3yrtDr.

Kangaslahti, S., Rosenfeld, E., and Saphra, N. Hidden breakthroughs in language model training. *arXiv preprint arXiv:2506.15872*, 2025.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

Kunin, D., Marchetti, G. L., Chen, F., Karkada, D., Simon, J. B., DeWeese, M. R., Ganguli, S., and Miolane, N. Alternating gradient flows: A theory of feature learning in two-layer neural networks. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=t7LKc0MMW6.

Lake, B. and Baroni, M. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International conference on machine learning*, pp. 2873–2882. PMLR, 2018.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40:e253, 2017. doi: 10.1017/S0140525X16001837.

Lau, E., Furman, Z., Wang, G., Murfet, D., and Wei, S. The local learning coefficient: A singularity-aware complexity measure. *arXiv preprint arXiv:2308.12108*, 2023.

Lee, J. H., Jiralerspong, T., Yu, L., Bengio, Y., and Cheng, E. Geometric signatures of compositionality across a language model's lifetime. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5292–5320, 2025.

Lindsey, J., Gurnee, W., Ameisen, E., Chen, B., Pearce, A., Turner, N. L., Citro, C., Abrahams, D., Carter, S., Hosmer, B., Marcus, J., Sklar, M., Templeton, A., Bricken, T., McDougall, C., Cunningham, H., Henighan, T., Jermyn, A., Jones, A., Persic, A., Qi, Z., Thompson, T. B., Zimmerman, S., Rivoire, K., Conerly, T., Olah, C., and Batson, J. On the biology of a large language model, 2025. URL https://transformer-circuits.pub/2025/attribution-graphs/biology.html#dives-multilingual.

Liu, Z., Michaud, E. J., and Tegmark, M. Omnigrok: Grokking beyond algorithmic data. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=zDiHoIWa0q1.

Lubana, E. S., Kawaguchi, K., Dick, R. P., and Tanaka, H. A percolation model of emergence: Analyzing transformers trained on a formal language. *arXiv preprint arXiv:2408.12578*, 2024.

MacDiarmid, M., Wright, B., Uesato, J., Benton, J., Kutasov, J., Price, S., Bouscal, N., Bowman, S., Bricken, T., Cloud, A., et al. Natural emergent misalignment from reward hacking in production rl. *arXiv preprint arXiv:2511.18397*, 2025.

Marchetti, G. L., Kunin, D., Myers, A., Acosta, F., and Miolane, N. Sequential group composition: A window into the mechanics of deep learning, 2026. URL https://arxiv.org/abs/2602.03655.

McAllester, D. A. Pac-bayesian model averaging. In *Proceedings of the twelfth annual conference on Computational learning theory*, pp. 164–170, 1999.

McGrath, T., Kapishnikov, A., Tomašev, N., Pearce, A., Wattenberg, M., Hassabis, D., Kim, B., Paquet, U., and Kramnik, V. Acquisition of chess knowledge in alphazero. *Proceedings of the National Academy of Sciences*, 119 (47):e2206625119, 2022.

Michaud, E., Liu, Z., Girit, U., and Tegmark, M. The quantization model of neural scaling. *Advances in Neural Information Processing Systems*, 36:28699–28722, 2023.

Milani Fard, M., Cormier, Q., Canini, K., and Gupta, M. Launch and iterate: Reducing prediction churn. *Advances in Neural Information Processing Systems*, 29, 2016.

Mittal, S., Raparthy, S. C., Rish, I., Bengio, Y., and Lajoie, G. Compositional attention: Disentangling search and retrieval. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=IwJPj2MBcIa.

Morwani, D., Edelman, B. L., Oncescu, C.-A., Zhao, R., and Kakade, S. M. Feature emergence via margin maximization: case studies in algebraic tasks. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=i9wDX850jR.

Muennighoff, N., Wang, T., Sutawika, L., Roberts, A., Biderman, S., Le Scao, T., Bari, M. S., Shen, S., Yong, Z. X., Schoelkopf, H., Tang, X., Radev, D., Aji, A. F., Almubarak, K., Albanie, S., Alyafeai, Z., Webson, A., Raff, E., and Raffel, C. Crosslingual generalization through multitask finetuning. In Rogers, A., Boyd-Graber, J., and Okazaki, N. (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15991–16111, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.891. URL https://aclanthology.org/2023.acl-long.891/.

Nanda, N., Chan, L., Lieberum, T., Smith, J., and Steinhardt, J. Progress measures for grokking via mechanistic interpretability. *arXiv preprint arXiv:2301.05217*, 2023.

Neyshabur, B., Salakhutdinov, R. R., and Srebro, N. Path-sgd: Path-normalized optimization in deep neural networks. *Advances in neural information processing systems*, 28, 2015.

Neyshabur, B., Bhojanapalli, S., McAllester, D., and Srebro, N. Exploring generalization in deep learning. *Advances in neural information processing systems*, 30, 2017.

Okawa, M., Lubana, E. S., Dick, R., and Tanaka, H. Compositional abilities emerge multiplicatively: Exploring diffusion models on a synthetic task. *Advances in Neural Information Processing Systems*, 36:50173–50195, 2023.

Olah, C. Distributed representations: Composition & superposition, 2023. URL https://transformer-circuits.pub/2023/superposition-composition/index.html.

Olsson, C., Elhage, N., Nanda, N., Joseph, N., DasSarma, N., Henighan, T., Mann, B., Askell, A., Bai, Y., Chen, A., et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

Papyan, V., Han, X., and Donoho, D. L. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663, 2020.

Pires, T., Schlinger, E., and Garrette, D. How multilingual is multilingual BERT? In Korhonen, A., Traum, D., and Màrquez, L. (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4996–5001, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1493. URL https://aclanthology.org/P19-1493/.

Power, A., Burda, Y., Edwards, H., Babuschkin, I., and Misra, V. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*, 2022.

Prakash, H. K. and Martin, C. H. Grokking and generalization collapse: Insights from HTSR theory, 2025. URL https://arxiv.org/abs/2506.04434.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, October 1986. ISSN 1476-4687. doi: 10.1038/323533a0.

Sakamoto, K. and Sato, I. Explaining grokking and information bottleneck through neural collapse emergence, 2026. URL https://arxiv.org/abs/2509.20829.

Saxe, A. M., McClelland, J. L., and Ganguli, S. Exact solutions to the nonlinear dynamics of learning in deep

linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

Schaeffer, R., Miranda, B., and Koyejo, S. Are emergent abilities of large language models a mirage? *Advances in neural information processing systems*, 36:55565–55581, 2023.

Schug, S., Kobayashi, S., Akram, Y., Wolczyk, M., Proca, A. M., von Oswald, J., Pascanu, R., Sacramento, J., and Steger, A. Discovering modular solutions that generalize compositionally. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=H98CVcX1eh.

Schäfer, A., Ravfogel, S., Hofmann, T., Pimentel, T., and Schlag, I. The role of language imbalance in cross-lingual generalisation: Insights from cloned language experiments, 2024. URL https://arxiv.org/abs/2404.07982.

Shai, A., Amdahl-Culleton, L., Christensen, C. L., Bigelow, H. R., Rosas, F. E., Boyd, A. B., Alt, E. A., Ray, K. J., and Riechers, P. M. Transformers learn factored representations, 2026. URL https://arxiv.org/abs/2602.02385.

Singh, J., Misra, D., and Orvieto, A. Explaining grokking in transformers through the lens of inductive bias. *arXiv preprint arXiv:2602.06702*, 2026.

Snell, W., Wind, J. S., Snikkers, B., Fraser, S., Newgas, A., Hoogland, J., Wang, G., Gordon, A., Zhou, W., and van Wingerden, S. Devinterp. https://github.com/timaeus-research/devinterp, 2026.

Veselovsky, V., Argin, B., Stroebl, B., Wendler, C., West, R., Evans, J., Griffiths, T. L., and Narayanan, A. Localized cultural knowledge is conserved and controllable in large language models, 2025. URL https://arxiv.org/abs/2504.10191.

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

Welling, M. and Teh, Y. W. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 681–688, 2011.

Wendler, C., Veselovsky, V., Monea, G., and West, R. Do llamas work in English? on the latent language of multilingual transformers. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15366–15394, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.820. URL https://aclanthology.org/2024.acl-long.820/.

Wiedemer, T., Mayilvahanan, P., Bethge, M., and Brendel, W. Compositional generalization from first principles. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=LqOQ1uJmSx.

Wiedemer, T., Brady, J., Panfilov, A., Juhos, A., Bethge, M., and Brendel, W. Provable compositional generalization for object-centric learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=7VPTUWkiDQ.

Xiao, X., Ping, H., Zhou, C., Cao, D., Li, Y., Zhou, Y.-Z., Li, S., Kanakaris, N., and Bogdan, P. Neuron-based multifractal analysis of neuron interaction dynamics in large models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=nt8gBX58Kh.

Xu, Y. The geometry of multi-task grokking: Transverse instability, superposition, and weight decay phase structure, 2026. URL https://arxiv.org/abs/2602.18523.

Yu, B. Veridical data science. In *Proceedings of the 13th international conference on web search and data mining*, pp. 4–5, 2020.

Zhao, X., Sharma, D., Uppaal, R., and Zhong, Y. Shattered compositionality: Counterintuitive learning dynamics of transformers for arithmetic. *arXiv preprint arXiv:2601.22510*, 2026.

Zucchet, N., Bornschein, J., Chan, S., Lampinen, A., Pascanu, R., and De, S. How do language models learn facts? dynamics, curricula and hallucinations. *arXiv preprint arXiv:2503.21676*, 2025.

A. Measure suite details

We give formal definitions of the measures included in our analysis, and the computational methods used for estimating them. Beyond the definitions below, the 53-measure suite includes the 14 weight statistics of Hu et al. (2023), computed as defined in their work, and the neural-collapse within- and between-class variance traces $\mathrm{Tr}(\Sigma_W)$ and $\mathrm{Tr}(\Sigma_B)$ (see the neural-collapse paragraph below).

Notation. Let $\boldsymbol{\theta} \in \mathbb{R}^d$ denote the full parameter vector, $\boldsymbol{\theta}_0$ its value at initialization, and $W_1, \ldots, W_L$ the 2D weight matrices (biases and embeddings excluded). Let $\widehat{\mathcal{L}}(\boldsymbol{\theta})$ denote the empirical loss on a mini-batch, m the training set size, and C the number of classes. For a matrix A we write $\|A\|_F$ for the Frobenius norm, $\sigma_{\max}(A)$ for the largest singular value, $\sigma_1 \geq \sigma_2 \geq \cdots$ for the full spectrum, and $A^\dagger$ for the Moore–Penrose pseudoinverse.

A.1. Parameter-space measures

These measures depend only on the current parameter values and can be computed without a data pass.

Parameter norm. The ℓ_2-norm of all trainable parameters concatenated into a single vector:

$$\|\boldsymbol{\theta}\|_2 = \sqrt{\sum_{i=1}^{d} \theta_i^2}. \tag{2}$$

An increasing parameter norm during training reflects growing weight magnitudes, which can indicate memorization or over-fitting. This is a common baseline measure but is known to be a weak predictor of generalization on its own (Jiang et al., 2019).

Weight matrix norms (Neyshabur et al., 2017; Bartlett et al., 2017). Six metrics derived from the per-layer Frobenius and spectral norms of the 2D weight matrices. Distance-from-init variants require the initial parameters $\boldsymbol{\theta}_0$.

$$\texttt{log_prod_of_fro} = \sum_{l=1}^{L} \log \|W_l\|_F^2 \tag{3}$$

$$\texttt{log_prod_of_spec} = \sum_{l=1}^{L} \log \sigma_{\max}(W_l)^2 \tag{4}$$

$$\texttt{fro_over_spec} = \sum_{l=1}^{L} \frac{\|W_l\|_F^2}{\sigma_{\max}(W_l)^2} \tag{5}$$

$$\texttt{l2_dist_from_init} = \|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|_2 \tag{6}$$

$$\texttt{fro_dist_from_init} = \sum_{l=1}^{L} \|W_l - W_{0,l}\|_F^2 \tag{7}$$

$$\texttt{spec_dist_from_init} = \sum_{l=1}^{L} \sigma_{\max}(W_l - W_{0,l})^2 \tag{8}$$

The log-product measures capture how much each layer "scales" the signal in log-space. The `fro_over_spec` ratio is related to the stable rank of each weight layer. The distance-from-init measures track how far optimization has moved from the starting point. All of these measures appear in Jiang et al. (2019).

Note on the implementation. Singular values are computed with `torch.linalg.svdvals` (with a CPU fallback on MPS devices). Only parameters with `ndim == 2` are treated as weight matrices; 1D parameters (biases, LayerNorm gains) are excluded.

Spectral entropy (DeMoss et al., 2025). For each 2D weight matrix W_l with singular values $\sigma_1^{(l)} \geq \cdots \geq \sigma_{r_l}^{(l)} > 0$, form the normalized distribution $p_j^{(l)} = \sigma_j^{(l)} / \sum_k \sigma_k^{(l)}$ and compute the entropy

$$H_l \;=\; -\sum_{j=1}^{r_l} p_j^{(l)} \log p_j^{(l)}. \tag{9}$$

We report the mean across all 2D weight matrices:

$$\bar{H} \;=\; \frac{1}{L}\sum_{l=1}^{L} H_l. \tag{10}$$

High spectral entropy indicates a more uniform singular value distribution (higher effective rank), while low entropy indicates that the matrix is dominated by a few directions. Spectral entropy tracks how much of the "capacity" of each layer is being used.

Note on the implementation. Equation (9) uses $p_j \log(p_j + \varepsilon)$ with $\varepsilon = 10^{-10}$ for numerical stability.

A.2. Loss-landscape measures

These measures characterize the geometry of the loss function in the neighborhood of the current parameters, and require a data batch and the loss function to evaluate.

A.2.1. Hessian statistics

Let $H = \nabla^2_{\boldsymbol{\theta}} \widehat{\mathcal{L}}(\boldsymbol{\theta})$ denote the Hessian of the loss with respect to all trainable parameters. We compute four quantities that characterize the curvature of the loss landscape.

We compute four scalar summaries of the Hessian curvature. The maximum eigenvalue

$$\lambda_{\max}(H) \;=\; \max_{\|\mathbf{v}\|=1} \mathbf{v}^\top H \mathbf{v} \tag{11}$$

captures the sharpness of the minimum, which is empirically associated with poorer generalization (Keskar et al., 2016). The trace $\mathrm{Tr}(H) = \sum_i \lambda_i(H)$ summarizes total curvature across all parameter directions, and the squared Frobenius norm $\|H\|_F^2 = \mathrm{Tr}(H^\top H) = \sum_i \lambda_i(H)^2$ summarizes curvature with stronger weight on top eigenvalues. Both are estimated via the Hutchinson identity (Hutchinson, 1989): for Rademacher vectors $\mathbf{v}_k \in \{+1, -1\}^d$,

$$\widehat{\mathrm{Tr}}(H) \;=\; \frac{1}{K}\sum_{k=1}^{K} \mathbf{v}_k^\top H \mathbf{v}_k, \qquad \widehat{\|H\|_F^2} \;=\; \frac{1}{K}\sum_{k=1}^{K} \|H\mathbf{v}_k\|_2^2. \tag{12}$$

Finally, the stable rank

$$\mathrm{sr}(H) \;=\; \frac{\|H\|_F^2}{\lambda_{\max}(H)^2} \tag{13}$$

is the effective number of "active" eigenvalue directions and is always $\leq \mathrm{rank}(H)$.

Note on the implementation. $\lambda_{\max}$ is estimated via power iteration (20 steps by default), each step computing $H\mathbf{v}$ via two backward passes without forming H explicitly. The Hutchinson estimators use $K = 50$ Rademacher samples.

A.2.2. Local Learning Coefficient

The local learning coefficient (LLC), also called the real log canonical threshold (RLCT), estimates the effective dimensionality of the loss basin around the current parameters $\boldsymbol{\theta}$ within the framework of Singular Learning Theory (Watanabe, 2009; Lau et al., 2023).

Formally, the RLCT λ is defined via the volume scaling of the KL-neighborhood:

$$\mathrm{Vol}\big\{\boldsymbol{\theta}' : K(\boldsymbol{\theta}_0, \boldsymbol{\theta}') \leq \epsilon\big\} \;\sim\; \epsilon^{\lambda} \quad (\epsilon \to 0), \tag{14}$$

where $K(\boldsymbol{\theta}_0, \boldsymbol{\theta}')$ is the KL divergence from the true distribution to the model at $\boldsymbol{\theta}'$. A higher RLCT indicates a more complex loss geometry.

In practice $\hat{\lambda}$ is estimated by running N_c independent stochastic gradient Langevin dynamics (SGLD) (Welling & Teh, 2011) chains from $\boldsymbol{\theta}$ and averaging the resulting loss values:

$$\hat{\lambda} \;=\; n\beta \cdot \Big(\langle \widehat{\mathcal{L}} \rangle_{\mathrm{SGLD}} - \widehat{\mathcal{L}}(\boldsymbol{\theta}) \Big), \tag{15}$$

where $n\beta$ is the inverse temperature and the expectation is taken over draws after a burn-in period. A quadratic localization penalty $\frac{\gamma}{2}\|\boldsymbol{\theta}' - \boldsymbol{\theta}\|_2^2$ keeps chains near the current checkpoint.

Note on the implementation. Uses the `devinterp` library (Snell et al., 2026). Default: 5 chains, 300 draws, 300 burn-in steps. Key hyperparameters ($n\beta$, localization γ, learning rate) are calibrated per setting.

A.2.3. PAC-Bayes measures

PAC-Bayes flatness measures how much additive Gaussian noise the parameters can absorb before classification accuracy degrades (Jiang et al., 2019; Dziugaite & Roy, 2017; McAllester, 1999). A binary search (with a doubling warm-up) finds the largest σ^* such that adding $\mathcal{N}(0, \sigma^2 I)$ noise to $\boldsymbol{\theta}$ drops training accuracy by at most $\Delta = 0.1$.

Under *uniform* noise, three measures are reported:

$$\texttt{pacbayes_flatness} \;=\; \frac{1}{(\sigma^*)^2} \tag{16}$$

$$\texttt{pacbayes_orig} \;=\; \frac{\|\boldsymbol{\theta}\|_2^2}{4(\sigma^*)^2} + \log\frac{m}{\sigma^*} + 10 \tag{17}$$

$$\texttt{pacbayes_init} \;=\; \frac{\|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|_2^2}{4(\sigma^*)^2} + \log\frac{m}{\sigma^*} + 10. \tag{18}$$

Under *magnitude-dependent* noise, a second search finds σ^*_{mag} for noise $\mathcal{N}(0,\ (\sigma_{\mathrm{mag}}\,|\theta_i|)^2)$ per coordinate, and we report the analogous three measures:

$$\texttt{pacbayes_mag_flatness} \;=\; \frac{1}{(\sigma^*_{\mathrm{mag}})^2} \tag{19}$$

$$\texttt{pacbayes_mag_orig} \;=\; \frac{1}{4}\sum_{i=1}^{d} \log\frac{((\sigma^*_{\mathrm{mag}})^2 + 1)\,\|\boldsymbol{\theta}\|_2^2}{d\,(\sigma^*_{\mathrm{mag}})^2\,\theta_i^2} + \log\frac{m}{\sigma^*_{\mathrm{mag}}} + 10 \tag{20}$$

$$\texttt{pacbayes_mag_init} \;=\; \frac{1}{4}\sum_{i=1}^{d} \log\frac{((\sigma^*_{\mathrm{mag}})^2 + 1)\,\|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|_2^2}{d\,(\sigma^*_{\mathrm{mag}})^2\,(\theta_i - \theta_{0,i})^2} + \log\frac{m}{\sigma^*_{\mathrm{mag}}} + 10. \tag{21}$$

Note on the implementation. The sigma search uses up to 20 doubling steps and 20 binary search steps (20 and 50 noise samples per evaluation respectively). Expensive: approximately 40 full training-set forward passes. Returns no value for regression settings.

A.3. Representation-space measures

These measures operate on activations at a designated penultimate ("representation") layer and require a forward pass over the validation set.

PCA effective dimension (Lee et al., 2025). Run PCA on the last-layer activations $Z \in \mathbb{R}^{n \times h}$ and count the number of principal components required to explain at least 99% of the total variance:

$$d_{\mathrm{PCA}} \;=\; \min\left\{ k \;:\; \frac{\sum_{j=1}^{k} \lambda_j}{\sum_{j=1}^{h} \lambda_j} \geq 0.99 \right\}, \tag{22}$$

where $\lambda_1 \geq \lambda_2 \geq \cdots$ are the eigenvalues of the empirical covariance of Z. This is a linear measure of intrinsic dimensionality.

TwoNN intrinsic dimensionality (Facco et al., 2017). For each sample i, let $r_1^{(i)}$ and $r_2^{(i)}$ be the distances to its first and second nearest neighbors in last-layer activation space. Define the ratio $\mu_i = r_2^{(i)}/r_1^{(i)} \geq 1$. The maximum-likelihood estimate of intrinsic dimension is

$$\hat{d} \;=\; \left(\frac{1}{n'} \sum_{i\,:\,\mu_i > 1} \log \mu_i \right)^{-1}, \tag{23}$$

where n' is the number of samples with $\mu_i > 1$. Unlike PCA effective dimension, TwoNN captures nonlinear structure in the representation manifold, assuming a locally Euclidean geometry.

Note on the implementation. Pairwise distances are computed with `torch.cdist` on the full validation batch. Returns zero if fewer than 2 samples satisfy $\mu_i > 1$.

Neural Collapse (Papyan et al., 2020). Neural collapse describes a terminal phase of training in which the class mean representations converge to a simplex equiangular tight frame and within-class variability vanishes. Let $\mu_c = \mathbb{E}[z \mid y = c]$ be the class-conditional mean representation and μ the global mean. Define the within- and between-class covariances:

$$\Sigma_W \;=\; \frac{1}{C} \sum_{c=1}^{C} \mathbb{E}\big[(z-\mu_c)(z-\mu_c)^\top \mid y = c\big], \tag{24}$$

$$\Sigma_B \;=\; \frac{1}{C} \sum_{c=1}^{C} (\mu_c - \mu)(\mu_c - \mu)^\top. \tag{25}$$

NC1, Within-class variability collapse:

$$\texttt{nc1} \;=\; \frac{1}{C} \mathrm{Tr}\Big(\Sigma_W \Sigma_B^\dagger\Big). \tag{26}$$

$\texttt{nc1} \to 0$ as within-class variability collapses relative to between-class spread.

NC2, ETF alignment: For centred class means $\tilde{\mu}_c = \mu_c - \mu$, compute pairwise cosine similarities $s_{cc'} = \tilde{\mu}_c^\top \tilde{\mu}_{c'} / (\|\tilde{\mu}_c\| \, \|\tilde{\mu}_{c'}\|)$ for $c \neq c'$. For a perfect simplex ETF all off-diagonal cosines equal $-1/(C-1)$. We report the mean and standard deviation of $\{s_{cc'}\}_{c \neq c'}$; the standard deviation approaches 0 at collapse.

A.4. Output-space measures

Margin measures (Jiang et al., 2019; Bartlett et al., 2017). For a classification sample $(\mathbf{x}, y)$, the margin is

$$\gamma(\mathbf{x}, y) \;=\; f_{\boldsymbol{\theta}}(\mathbf{x})_y - \max_{y' \neq y} f_{\boldsymbol{\theta}}(\mathbf{x})_{y'}, \tag{27}$$

the gap between the correct-class logit and the best competing logit. We use the 10th-percentile margin over the training set, $\gamma = \mathrm{Percentile}_{10}\{\gamma(\mathbf{x}_i, y_i)\}$, and combine it with weight norms to produce eight metrics:

$$\texttt{margin_10th_percentile} = \gamma \tag{28}$$

$$\texttt{inverse_margin} = \gamma^{-2} \tag{29}$$

$$\texttt{log_prod_of_spec_over_margin} = \sum_l \log \sigma_{\max}(W_l)^2 - \log \gamma^2 \tag{30}$$

$$\texttt{log_sum_of_spec_over_margin} = \log L + \tfrac{1}{L} \sum_l \log \sigma_{\max}(W_l)^2 - \log \gamma^2 \tag{31}$$

$$\texttt{log_prod_of_fro_over_margin} = \sum_l \log \|W_l\|_F^2 - \log \gamma^2 \tag{32}$$

$$\texttt{log_sum_of_fro_over_margin} = \log L + \tfrac{1}{L} \sum_l \log \|W_l\|_F^2 - \log \gamma^2 \tag{33}$$

$$\texttt{log_spec_bound_orig} = \texttt{log_prod_of_spec_over_margin} + \log\Big(\textstyle\sum_l \frac{\|W_l\|_F^2}{\sigma_{\max}(W_l)^2}\Big) \tag{34}$$

$$\texttt{log_spec_bound_init} = \texttt{log_prod_of_spec_over_margin} + \log\Big(\textstyle\sum_l \frac{\|W_l - W_{0,l}\|_F^2}{\sigma_{\max}(W_l)^2}\Big) \tag{35}$$

The "sum" variants replace the product of per-layer norms with L times the geometric mean, normalizing for depth. Returns no value for regression settings or when $\gamma \leq 0$.

Prediction entropy. Computed from the softmax distribution $\mathbf{p}(\mathbf{x}) = \mathrm{softmax}(f_{\boldsymbol{\theta}}(\mathbf{x}))$ over the full validation set. Three scalars are reported:

$$\texttt{mean_entropy} = \frac{1}{n} \sum_{i=1}^{n} \Big(-\sum_{c=1}^{C} p_c(\mathbf{x}_i) \log p_c(\mathbf{x}_i) \Big), \tag{36}$$

$$\texttt{mean_max_prob} = \frac{1}{n} \sum_{i=1}^{n} \max_c \, p_c(\mathbf{x}_i), \tag{37}$$

$$\texttt{high_confidence_frac} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}\Big[\max_c p_c(\mathbf{x}_i) > 0.9\Big]. \tag{38}$$

These track the transition from uncertain to confident predictions during training. For sequence models with a loss mask, values are averaged over unmasked output positions.

Prediction churn. Measures how much the model's predictions change between consecutive checkpoints. Let $\mathbf{p}^{(t)}$ and $\mathbf{p}^{(t-1)}$ be the output distributions at the current and previous checkpoint.

Hard churn, fraction of validation examples on which the argmax prediction flips:

$$\texttt{hard} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}\Big[\arg\max_c p_c^{(t)}(\mathbf{x}_i) \neq \arg\max_c p_c^{(t-1)}(\mathbf{x}_i)\Big]. \tag{39}$$

Soft churn (JSD), mean Jensen–Shannon divergence between output distributions:

$$\texttt{soft_jsd} = \frac{1}{n} \sum_{i=1}^{n} \mathrm{JSD}\Big(\mathbf{p}^{(t)}(\mathbf{x}_i) \,\|\, \mathbf{p}^{(t-1)}(\mathbf{x}_i)\Big), \tag{40}$$

where $\mathrm{JSD}(P\|Q) = \frac{1}{2}\mathrm{KL}(P\|M) + \frac{1}{2}\mathrm{KL}(Q\|M)$ with $M = \frac{1}{2}(P+Q)$. Returns an empty result on the first call (no previous checkpoint to compare).

A.5. HMM-baseline weight statistics (Hu et al., 2023)

This is a set of fourteen cheap weight and singular-value summary statistics that reproduce the feature set used by Hu et al. (2023), who fit a hidden Markov model (HMM) over training trajectories to discover discrete latent "states" of training dynamics. We include their feature set as a baseline. All statistics depend only on the current parameters and require no data pass.

Per-matrix statistics. The following eight quantities are computed on each 2D weight matrix and then *averaged* over the L matrices. For a matrix W with entries W_{ij} and singular values $\sigma_1 \geq \cdots \geq \sigma_r > 0$:

$$\texttt{hmm_l1} = \frac{1}{L}\sum_{l=1}^{L}\sum_{ij}\left|(W_l)_{ij}\right| \tag{41}$$

$$\texttt{hmm_fro} = \frac{1}{L}\sum_{l=1}^{L}\|W_l\|_F \tag{42}$$

$$\texttt{hmm_trace} = \frac{1}{L}\sum_{l=1}^{L}\mathrm{Tr}(W_l) \tag{43}$$

$$\texttt{hmm_spectral} = \frac{1}{L}\sum_{l=1}^{L}\sigma_{\max}(W_l) \tag{44}$$

$$\texttt{hmm_code_sparsity} = \frac{1}{L}\sum_{l=1}^{L}\frac{\sum_{ij}|(W_l)_{ij}|}{\|W_l\|_F} \tag{45}$$

$$\texttt{hmm_comp_sparsity} = \frac{1}{L}\sum_{l=1}^{L}\frac{\mathrm{Tr}(W_l)}{\sigma_{\max}(W_l)} \tag{46}$$

$$\texttt{hmm_mean_sv} = \frac{1}{L}\sum_{l=1}^{L}\frac{1}{r_l}\sum_{j=1}^{r_l}\sigma_j^{(l)} \tag{47}$$

$$\texttt{hmm_var_sv} = \frac{1}{L}\sum_{l=1}^{L}\mathrm{Var}_j\big(\sigma_j^{(l)}\big) \tag{48}$$

The two "sparsity" ratios follow the paper's terminology: `hmm_code_sparsity` is the ℓ_1/ℓ_2 ratio of the flattened matrix (smaller for sparser weights), and `hmm_comp_sparsity` is $\mathrm{Tr}(W)/\sigma_{\max}(W)$.

Aggregate weight and bias statistics. Let $\mathbf{w}$ be the concatenation of all entries of the 2D weight matrices and $\mathbf{b}$ the concatenation of all 1D parameters (biases). We report the mean, median, and variance of each:

$$\texttt{hmm_w_mean} = \mathrm{mean}(\mathbf{w}), \qquad \texttt{hmm_w_median} = \mathrm{median}(\mathbf{w}), \qquad \texttt{hmm_w_var} = \mathrm{Var}(\mathbf{w}), \tag{49}$$

$$\texttt{hmm_b_mean} = \mathrm{mean}(\mathbf{b}), \qquad \texttt{hmm_b_median} = \mathrm{median}(\mathbf{b}), \qquad \texttt{hmm_b_var} = \mathrm{Var}(\mathbf{b}). \tag{50}$$

Note on the implementation. Singular values below 10^{-5} are zeroed before computing `hmm_mean_sv` and `hmm_var_sv`.

B. Experimental setup details

B.1. Modular addition

Task and data. The task is to predict $(a + b) \bmod p$ for $p = 113$. The full dataset consists of all $p^2 = 12{,}769$ ordered pairs (a, b) with $a, b \in \{0, \ldots, p-1\}$. Each example is encoded as a 3-token sequence $[a,\ b,\ =]$, where the equals token is the special token $p = 113$; the vocabulary therefore has $p + 1 = 114$ tokens. The target is a class label in $\{0, \ldots, p-1\}$.

A random 30/70 train/val split is applied to the full grid (seed fixed per run), giving 3,830 training pairs and 8,939 validation pairs. Training uses a single full-batch gradient step per epoch; the validation set is never used for gradient updates.

This setup matches Nanda et al. (2023) with the following known differences: (i) the linear-warmup scheduler starts at 10% rather than 0%; (ii) attention is implemented via `nn.MultiheadAttention` rather than a custom einsum, and as a consequence the packed Q/K/V projection (`in_proj_weight`) keeps PyTorch's default `xavier_uniform_` initialization (std $1/\sqrt{2\,d_{\text{model}}}$, uniform) rather than the reference's $\mathcal{N}(0,\ 1/d_{\text{model}})$. Cross-entropy is computed in `float64` (via `log_softmax`).

Architecture and optimization. The model is a 1-layer decoder-only transformer with:

- $d_{\text{model}} = 128$, 4 attention heads, $d_{\text{ff}} = 4 \times 128 = 512$
- no LayerNorm, ReLU activations, no attention biases
- scaled initialization: $\mathcal{N}(0,\ 1/d_{\text{model}})$ for the embeddings, the attention output projection, and the MLP weights; $\mathcal{N}(0,\ 1/\text{vocab_size})$ for the output (unembedding) head; all biases initialized to zero (the packed Q/K/V projection is the one exception, see known differences above)
- causal attention mask; classification from the last token position (=)

Training uses AdamW with learning rate 10^{-3}, weight decay 1.0, $(\beta_1, \beta_2) = (0.9, 0.98)$, and a 10-step linear warmup. The model is trained for 20,000 epochs with cross-entropy loss.

Measurement schedule. Complexity measures are evaluated every 50 epochs by default. The LLC, the most expensive measure, is computed every 200 epochs (5 chains, 300 draws, 300 burn-in steps, $n\beta = 200$, localization $\gamma = 5.0$, lr $= 10^{-5}$). PAC-Bayes measures are evaluated every 100 epochs, and margin measures, excluded loss, and prediction entropy/churn every 20 epochs.

B.2. Multilingual modular addition

Task and data. The multilingual setting trains a single transformer on the same modular addition task expressed in N "cloned languages". Each language $l \in \{1, \ldots, N\}$ is an independent copy of the task with a *disjoint* token vocabulary: input tokens for language l are offset by $(l-1)(p+1)$, so the full input vocabulary has $N(p+1)$ tokens, and target labels for language l are offset by $(l-1)p$, giving Np output classes (see Figure 5). This ensures the model must learn to route each example through language-specific circuitry while sharing parameters across languages.

Each language independently samples its training data from the full p^2 grid using a per-language multiplier applied to the base training fraction of 30%, so that the languages grok sequentially, in order of their training-set sizes. We evaluate two variants:

- **Three-language.** $N = 3$ at multipliers $(1.0,\ 0.9,\ 0.7)$, i.e. effective fractions of 30%, 27%, and 21% of the p^2 grid. The input vocabulary has $N(p+1) = 3 \times 114 = 342$ tokens and the output head has $Np = 3 \times 113 = 339$ classes.
- **Two-language.** $N = 2$ using only the first and third languages of the three-language variant, i.e. multipliers $(1.0,\ 0.7)$ / effective fractions 30% and 21%. This wider separation gives better-separated per-language transitions. The input vocabulary has $2 \times 114 = 228$ tokens and the output head has $2 \times 113 = 226$ classes.

The resulting per-language training-set sizes are:

Language	Multiplier	Effective fraction	Training pairs
L1	1.0	30%	3,830
L2	0.9	27%	3,447
L3	0.7	21%	2,681

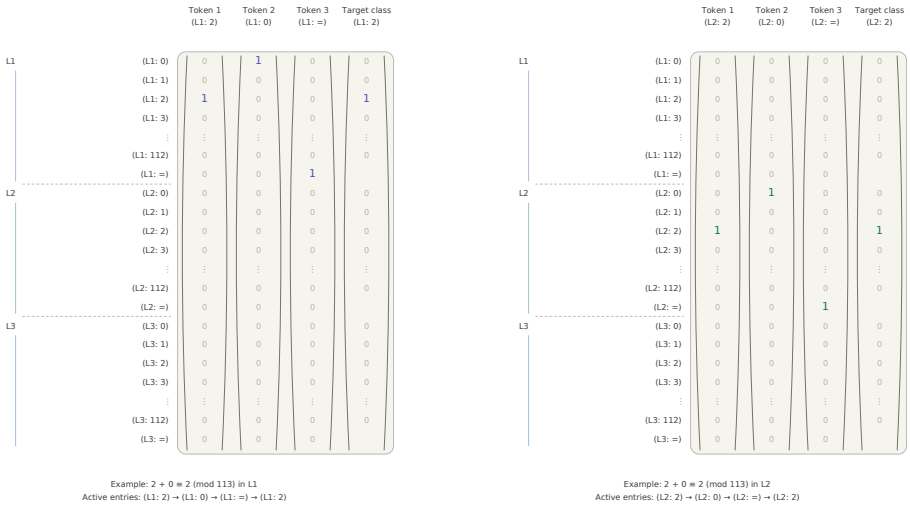

Figure 5. This figure shows two training samples for $p = 113$ and $N = 3$, displayed as a sequence of one-hot input tokens and the target class. The same equation "$2 + 0 \equiv 2 \pmod{113}$" is expressed in two of the $N = 3$ languages, which share no tokens.

The three-language pool thus contains $3{,}830 + 3{,}447 + 2{,}681 = 9{,}958$ training pairs and the two-language pool (L1, L3) contains $3{,}830 + 2{,}681 = 6{,}511$. In both variants the combined training pool is used as a single full batch. Validation uses the per-language held-out complements and is evaluated both in aggregate and per-language. Per-language metrics track exact accuracy (`Li/acc`), cross-entropy loss (`Li/loss`), and numerical accuracy (`Li/num_acc`), where numerical accuracy counts a prediction as correct if the predicted number matches $(a + b) \bmod p$ regardless of language.

Architecture and optimization. The architecture is identical to the modular addition setting, with the exception of the enlarged vocabulary and output head (342/339 for the three-language variant, 228/226 for the two-language variant). All other architectural choices (1 layer, 4 heads, $d_{\text{model}} = 128$, no LayerNorm, ReLU, scaled init) are unchanged. The optimizer is also identical: AdamW, lr $= 10^{-3}$, weight decay 1.0, $(\beta_1, \beta_2) = (0.9, 0.98)$, 10-step linear warmup. The three-language variant is trained for 50,000 epochs and the two-language variant for 60,000 epochs.

Measurement schedule. Both variants use the same schedule. Most measures are evaluated every 50 epochs; the Hessian-based measures every 100. The LLC is evaluated every 1,000 epochs (same chain/draw/burn-in settings as above). PAC-Bayes measures are evaluated every 1,000 epochs and margin measures every 200.

C. Evaluation details

C.1. Preprocessing

The pipeline shape is shared across all three settings, with only constants differing.

Run filter and truncation. We exclude runs where the model never reached generalization, using an accuracy threshold of 0.99. In modular addition we filter on validation accuracy and drop 1 of 40 seeds. In the multilingual settings we filter on the validation accuracy of the language with the smallest data fraction. This drops 11 of 40 seeds in the three-language setting and 13 of 39 seeds in the two-language setting (one of its 40 runs is missing its artifacts), leaving the 29 and 26 runs used throughout. Excluding non-grokking runs gives the HMM a cleaner input and avoids right-censoring the retrospective targets on a meaningful fraction of seeds. Furthermore, we truncate each run at the last firing epoch of the slowest measure (the LLC in all settings) so that no measure has to be extrapolated beyond its computed range. The resulting window is epoch $19{,}800$ for modular addition (LLC fires every 200 epochs), and epochs $49{,}000$ and $59{,}000$ for the three- and two-language multilingual settings respectively (LLC fires every $1{,}000$ epochs in both).

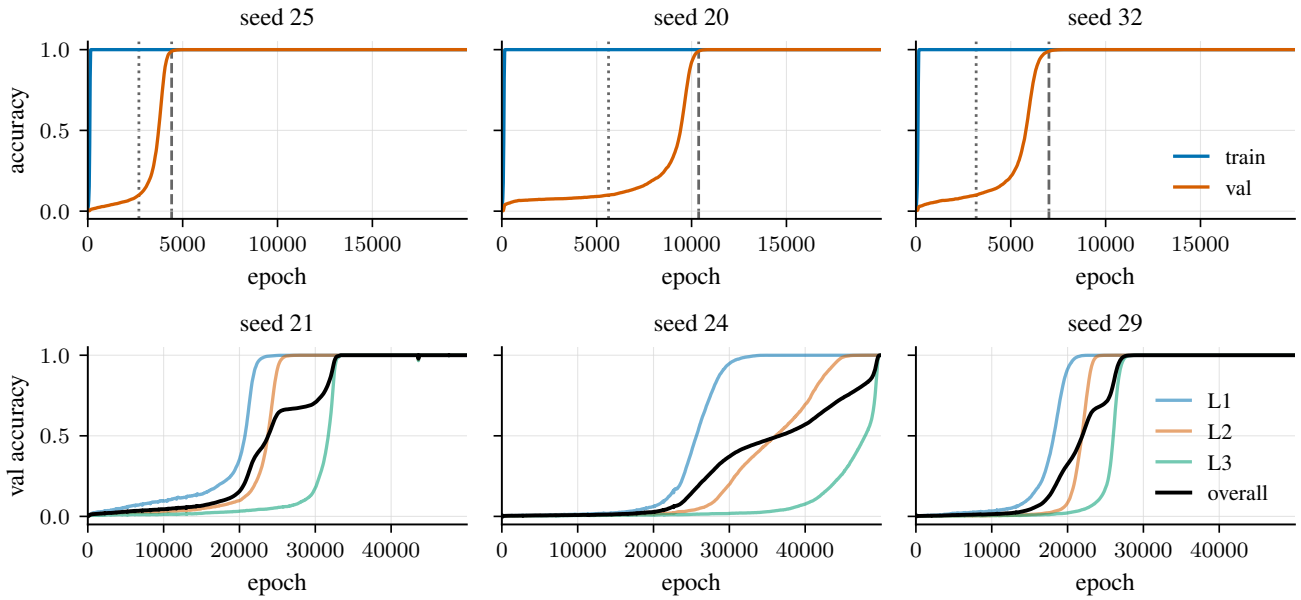

Figure 6. **Behavioral curves for three random seeds per setting.** *(Top)* Modular addition. *(Bottom)* Multilingual modular addition; colored curves are per-language validation accuracy, the black curve is overall.

Imputation and gridding. A small number of cells are NaN where we expected a measure value, mostly from measures that are mathematically undefined for parts of training (e.g., prediction churn is undefined at epoch 0 because it is a first difference and there is no earlier checkpoint, and the margin-normalized norm bounds are undefined whenever the training margin in the denominator is non-positive). A few NaNs also appear sporadically elsewhere, but rarely. We impute these by either backfilling at the start of training or linearly interpolating between adjacent valid values.

In the next step, we restrict each run to a uniform-step grid (50 epochs in modular addition, 100 in multilingual; the larger step in the longer setting keeps the pooled HMM input at a comparable length) and linearly interpolate the slower measures onto it. Measure firing intervals are listed in Appendix B.

Smoothing and z-scoring. We apply a per-run, per-column Gaussian smoothing to the value trajectories with $\sigma = 2$ grid-steps in modular addition and $\sigma = 10$ in multilingual, then per-run z-score every scalar column. The sigmas were chosen to balance capturing the behavioral inflection points against absorbing small-scale measure noise, and were set per setting from the overall noise level of the trajectories (multilingual trajectories, especially the first differences, are noisier than modular addition, which is why the multilingual sigmas are higher). They are fixed across measures within a setting, so the smoothing does not advantage any individual measure's ranking. We then build a parallel diff frame by taking first differences of the smoothed values, smoothing again with $\sigma = 8$ grid-steps in modular addition and $\sigma = 20$ in multilingual, and per-run z-scoring once more. Per-run rather than across-run z-scoring keeps trajectories comparable across seeds without leaking absolute-scale information into the HMM. Figures 7 and 8 show four canonical measures (one per class) for one seed of each setting, both as values and as the smoothed first differences that the bivariate HMM consumes alongside them. The validation accuracy curve sits in light gray behind each panel for behavioral context.

C.2. HMM model selection

The HMM input is the bivariate $[\text{value}_t, \text{diff}_t]$ trajectory described in Section 3. Here we cover the model-selection choices on top of that setup.

Sweep range and split. We sweep K from 2 to 7. The upper bound gets picked for a non-trivial fraction of subsets in both settings, and we did not extend further because beyond 7 the overfitting risk on these sequence lengths starts to outweigh the marginal log-likelihood gain. The 20% held-out fraction is split at the seed level, so the held-out log-likelihood measures generalization to unseen runs rather than to unseen timesteps within a known run.

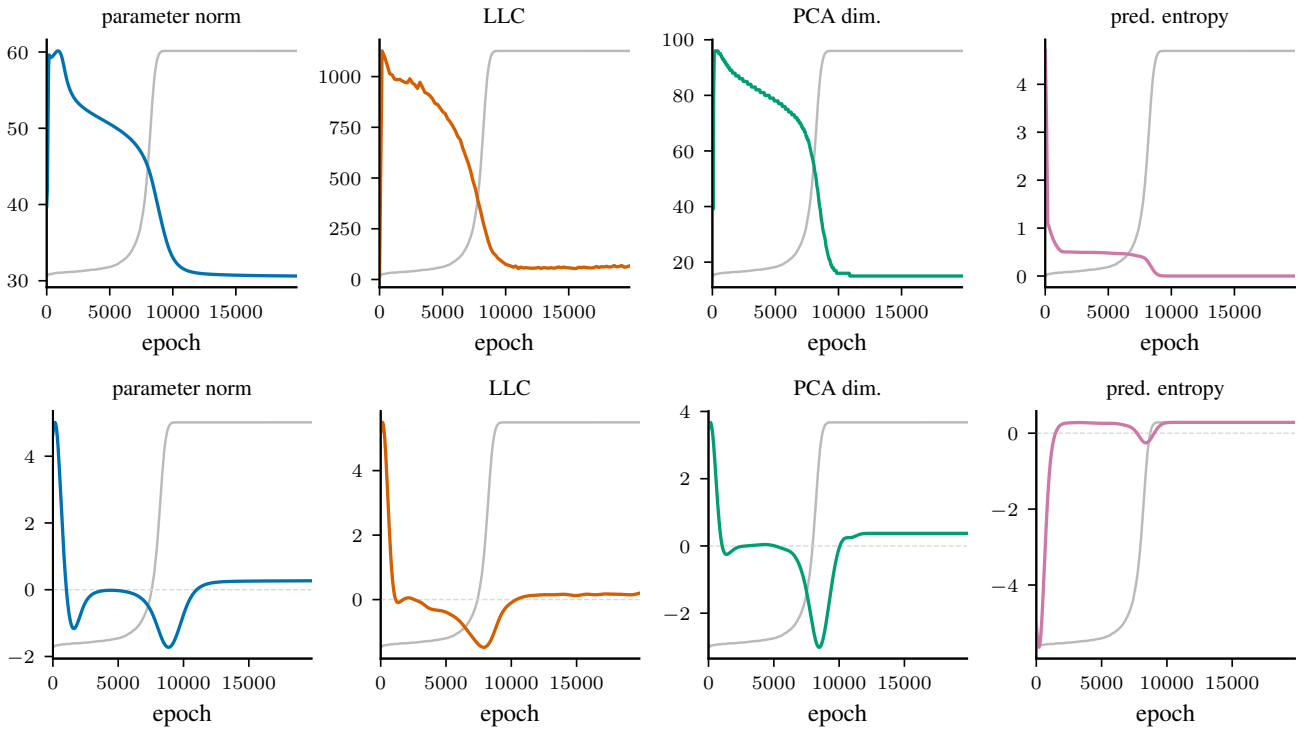

Figure 7. **Selected measures for modular addition, one per class, single seed.** Colored by class (parameter, loss-landscape, representation, output). Validation accuracy in light gray behind each panel. *(Top)* Values, post-interpolation. *(Bottom)* Smoothed first differences (per-run z-scored); dashed gray horizontal at zero is the no-change baseline.

Fitting. For each (K, subset) pair we run 10 Baum–Welch restarts from random initial parameters with full-covariance Gaussian emissions, and keep the restart with the highest held-out log-likelihood. Full covariance lets each state-conditional distribution capture correlation between the value and its rate of change, at negligible parameter cost for the bivariate measure subsets.

Selecting $K^\star$. We pick $K^\star$ by an elbow rule on the held-out log-likelihood. Starting from $K = 2$, we accept the increment to $K + 1$ as long as the gain stays at least 20% of the first ($K = 2 \to 3$) gain, and we stop at the first K where the gain drops below this threshold. If the threshold is never crossed, we fall back to the upper bound. We also computed BIC as a cross-check, since this is the choice of Hu et al. (2023), whose setup ours is built on top of. For many subsets BIC decreases close to monotonically in K (bottom rows of Figures 9 and 10), so picking the argmin would push the selected K to the upper bound by default. The elbow rule on held-out LL avoids this and gives a stopping point that varies meaningfully across subsets. We read this near-monotone BIC profile as a mild signature of model misspecification: a Gaussian-emission HMM is at best a smooth approximation to the true, likely non-Gaussian, generating process, so additional states keep buying likelihood without necessarily indicating genuinely new phases. This is consistent with the framework-level limitations discussed in Section 6.

C.3. Information encoded in the phase structure

The HMM-inferred phases summarise each run by how much time the model spent in each state. We use this summary as a feature for predicting behavioral landmarks on held-out seeds, which gives a quantitative read on how much behavioral information the phase structure carries.

Procedure. For each subset and each K-label we decode the per-seed Viterbi paths and build the bag-of-states feature, the count of grid steps each seed spent in each state (the same vector we use elsewhere in the paper). For the equal-K-partition floor we substitute the deterministic uniform-length partition for the Viterbi sequence. We then define retrospective behavioral targets per seed as the first epoch where validation accuracy crosses the onset threshold (0.1) or the completion threshold (0.99). In multilingual we use the per-language validation accuracy. We score each (subset, target) pair by 5-fold

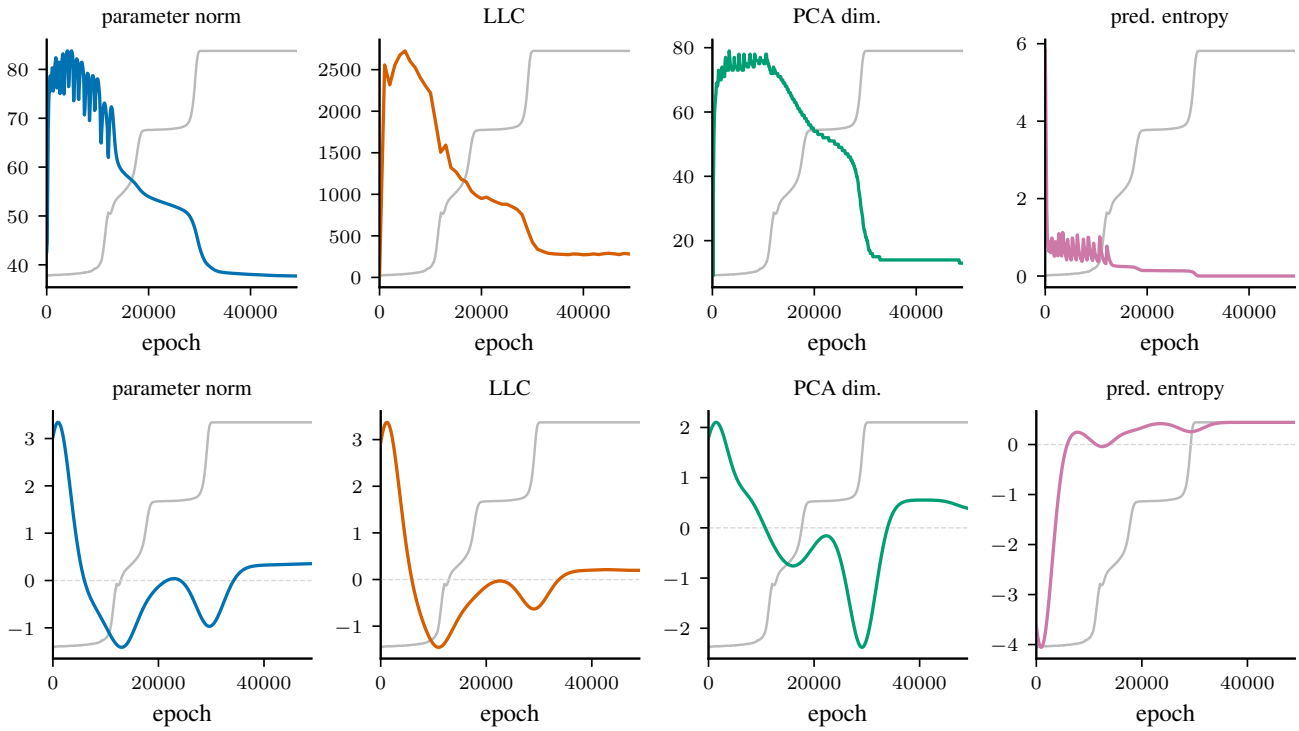

Figure 8. **Selected measures for multilingual modular addition, same four measures as in Figure 7.** Validation accuracy in light gray behind each panel. *(Top)* Values, post-interpolation. *(Bottom)* Smoothed first differences (per-run z-scored); dashed gray horizontal at zero is the no-change baseline.

cross-validated Ridge R^2, with the regularisation strength picked per fold by RidgeCV. The K-labels we sweep are the per-subset elbow $K^\star$ plus two fixed K values per setting ($\{3, 4\}$ in modular addition, $\{4, 6\}$ in three-language multilingual), so we can compare the headline ranking against neighbouring K choices.

Targets we considered and dropped. We initially also computed end-of-memorization (first epoch where train_acc $>$ 0.99) as a target, but it crossed at almost the same epoch on every seed, so the variance was too small for ridge to fit anything meaningful and we dropped it. We also tried duration (completion $-$ onset) and got relatively poor scores.

This makes sense given the correlation structure of the targets. Onset and completion are strongly correlated across seeds ($r = 0.80$ in modular addition, 0.69–0.90 per language in multilingual), so duration has much less variance than either endpoint, while predicting it inherits the errors of both. Both effects depress R^2 even when the endpoints themselves are predicted well. A small number of multilingual seeds are right-censored on the L3 targets because the preprocessing grok filter checks the original training trajectory while we truncate the analysis window at the LLC's last firing; the affected seeds get their target pinned to the last epoch and contribute as a small amount of extra noise to the regression.

Robustness across K and onset threshold. The per-subset ranking is relatively stable with respect to the K choice. Pairwise scatters of R^2 at different K-labels (Figure 11) cluster along the diagonal across the three K-pairs we compare. To check robustness with respect to the onset threshold, we also score each subset against onset targets defined at thresholds in $\{0.05, 0.10, 0.20, 0.30, 0.50\}$ and look at the Spearman ρ between the resulting per-subset rankings (Figure 12). On modular addition the rankings stay close for thresholds at or above 0.10 and drift somewhat at 0.05. On multilingual L1 even the 0.05 threshold stays closely related to the others.

C.4. Consistency across seeds

This analysis quantifies whether runs at the same validation accuracy, our behavioral proxy for compositional capability, are assigned to the same inferred phase. We align validation-accuracy trajectories between pairs of seeds with Dynamic Time Warping.

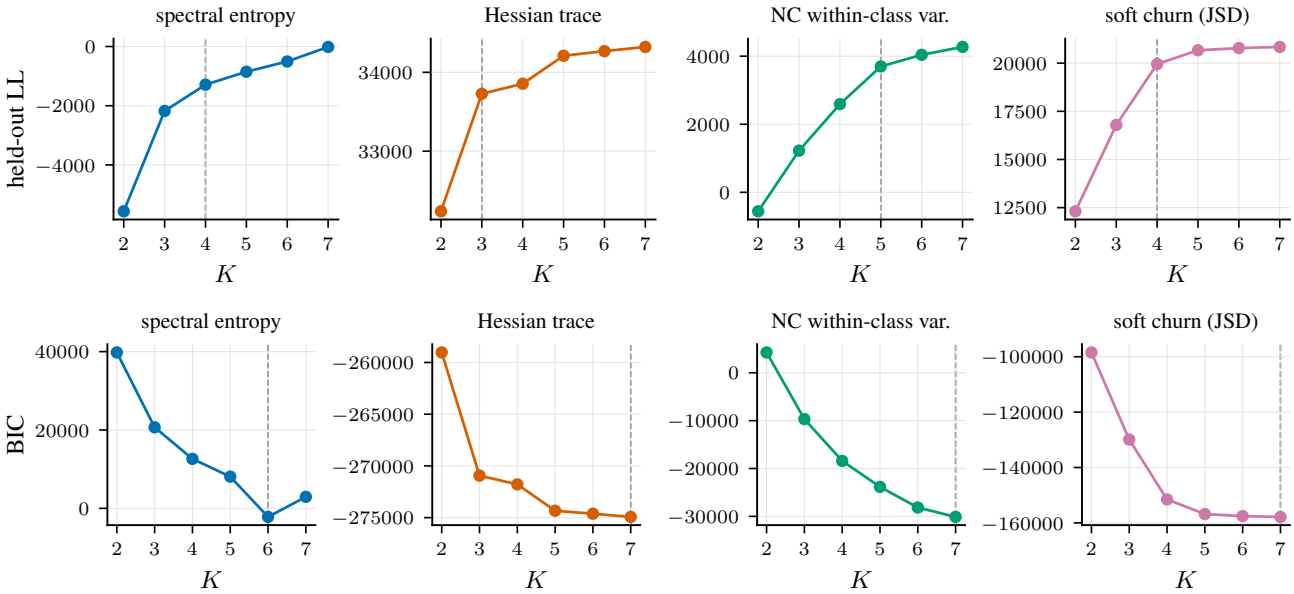

Figure 9. **Held-out log-likelihood (top) and BIC (bottom) vs K for one representative measure per class, modular addition.** Colored by class (parameter, loss-landscape, representation, output). The dashed vertical line marks the elbow $K^\star$ (top) and $\arg\min_K$ BIC (bottom).

Capability proxy and DTW alignment. In modular addition the proxy is the bivariate [train_acc, val_acc] trajectory, so the alignment sees both memorization and generalization. In multilingual we use the per-language validation-accuracy trajectory, so the warp captures the staggered transitions across languages rather than just aggregate progress. We align each pair of trajectories with Dynamic Time Warping under a Sakoe-Chiba band of width 60% of the run length, which keeps alignments local and prevents pathological warps. The qualitative effect is shown in Figures 13 and 14.

Anchor coarsening. The DTW warp path is a sequence of (i_A, i_B) index pairs of length at least T, where T is the number of timesteps per run. DTW often pins many timesteps of one run to a single timestep of the other, so naively scoring the agreement at every position along the warp path inflates apparent agreement through repeated labels. We control for this by sampling N positions evenly spaced along the warp path. Each sampled position contributes one (i_A, i_B) pair, and we compute the ARI between the two seeds' Viterbi states at those positions. We use $N = 20$ for the main analysis. A sweep over $\{10, 20, 40\}$ confirmed that the per-pair ARI distribution stays close across this range.

Circular-shift null. Even Viterbi sequences with no shared temporal structure can produce positive ARI under DTW. The warp can match states with the same identity wherever they occur in either run, so two unrelated runs that visit the same set of states produce non-trivial agreement. The equal-K partition floor row in Figure 17 sits above zero for this reason, and gives a direct read of what a structure-free phase sequence scores. As a per-pair version of the same control, we also compute a circular-shift null. We shift seed B's full Viterbi sequence by 20 random offsets, recompute the anchor-ARI at the same warp-path positions, and average across the 20 shifts. The shift preserves run-length structure, marginal state proportions, and transition counts, but destroys the alignment with seed A. We do not report the centered mean ARI (observed $-$ null mean) as a separate figure; the floor row in Figure 17 carries the same message at the figure level, and the per-subset ranking is broadly preserved whether we read raw mean ARI or center each subset against its own null (Spearman ρ between the two rankings is 0.79–0.94 across settings and K-labels), though individual subsets can move by many ranks; in modular addition the Hessian-based subsets rise into the top ranks under centering while the equal-K floor drops toward the bottom.

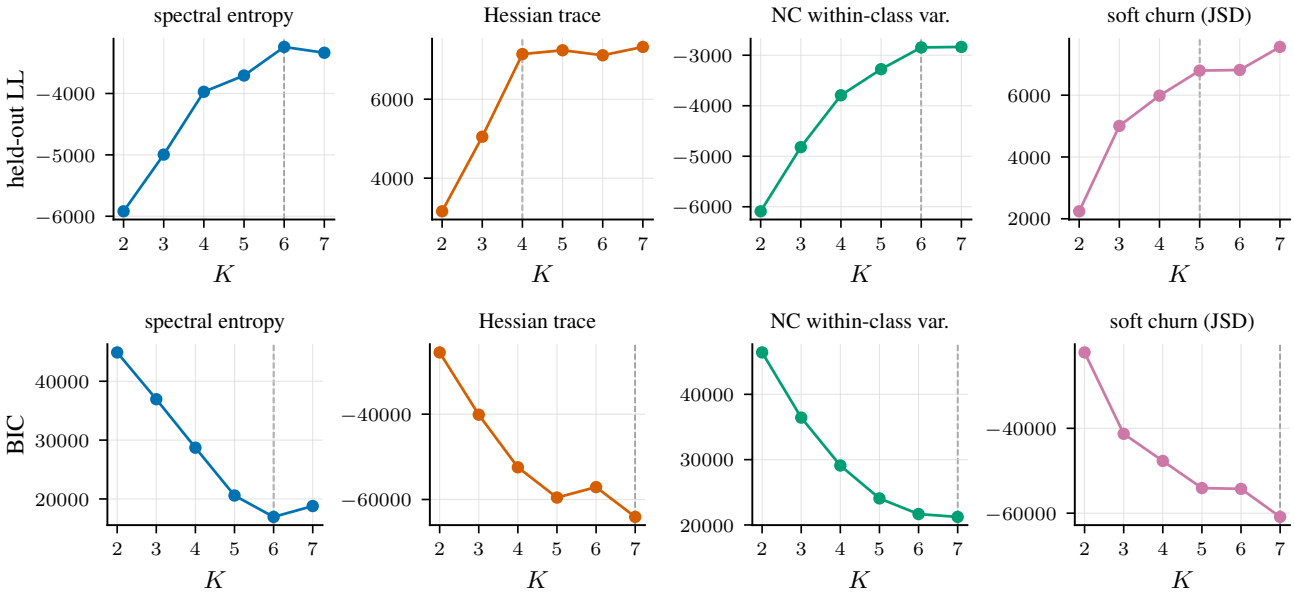

Figure 10. **Held-out log-likelihood (top) and BIC (bottom) vs K, multilingual modular addition.** Same four measures as Figure 9. The dashed vertical line marks the elbow $K^\star$ (top) and $\arg\min_K$ BIC (bottom).

D. Additional results

D.1. Qualitative phase overlays

Figure 15 is the two-language analogue of Figure 3. At $K = 6$, both the validation-accuracy reference and the LLC partition the trajectories better than in the three-language case, with state boundaries near the per-language onsets and completions. State assignments still show some oscillation around the transitions, putting the two-language setting between modular addition and three-language multilingual.

D.2. Retrospective predictability

Figure 16 (left) is the two-language analogue of Figure 4. The relative ordering of subsets and the gap between the validation-accuracy references and the rest of the measures sit between the modular addition and three-language multilingual cases. Appendix C.3 reports robustness of the per-subset ranking across K choices (Figure 11) and across the onset threshold (Figure 12).

D.3. Cross-seed consistency

Figure 17 shows the full per-subset ARI distributions for modular addition and three-language multilingual.

D.4. Relations across measures

Pooled Viterbi and pairwise ARI. For each measure subset we concatenate the per-seed Viterbi paths into one length-NT label vector, where N is the number of seeds and T is the analysis-window length per seed. We then compute the pairwise ARI matrix between subsets. ARI is chance-corrected and label-permutation invariant, so subsets with different state cardinalities are directly comparable, which matters because per-subset $K^\star$ varies. The whole analysis runs at a single fixed K ($K = 4$ in modular addition, $K = 6$ in both multilingual settings), matching the per-setting K used in the rest of the paper.

Hierarchical clustering and block-median view. We run average-linkage hierarchical clustering on $1 - \mathrm{ARI}$ as the dissimilarity and reorder the rows and columns of the full subset-by-subset ARI matrix by the resulting dendrogram leaf order. The block-median view in Figures 18 and 19 collapses this matrix to one cell per pair of measure classes, taking the median ARI across all member-pair combinations and excluding the diagonal in the within-class blocks. Behavioral

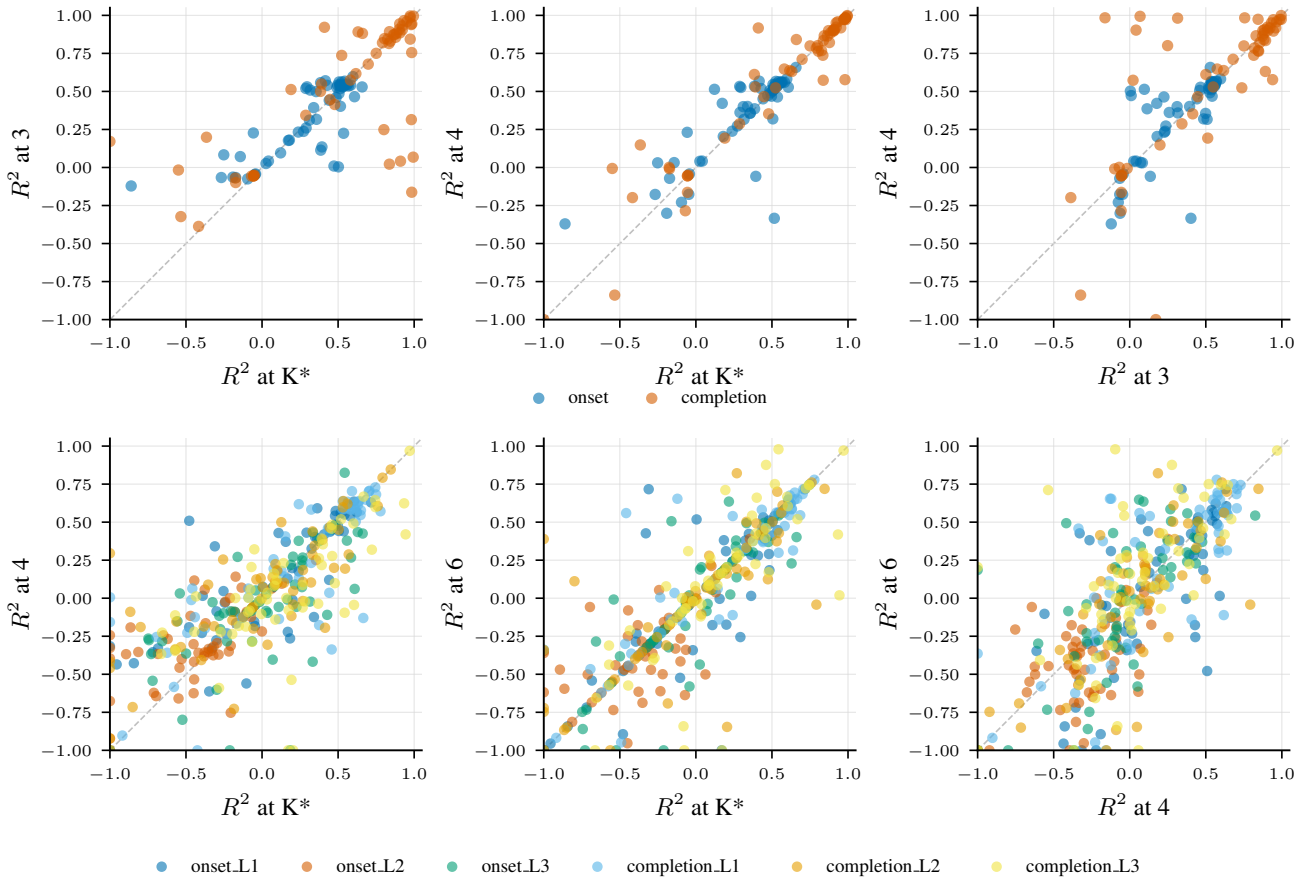

Figure 11. **Stability of per-subset R^2 across K choices.** Each point is one (subset, target) pair, clipped to $[-1.0, 1.05]$. Tight diagonals indicate that the ranking is insensitive to K. *(Top)* Modular addition: per-subset R^2 at three pairs of K-labels, by target. *(Bottom)* Multilingual modular addition: per-subset R^2 at three pairs of K-labels, with the six per-language onset and completion targets distinguished by colour.

anchors are kept as their own singleton rows and columns rather than being aggregated into the measure classes, since each anchor's relationship to each class is independently informative.

Class-level structure. The reordered full heatmap shows a few visible clusters, but none of them line up with the measure-class boundaries from Section 3.2. The block-median view confirms this at the class level, and the per-pair within-versus-across-class ARI distributions show the same overlap. Most classes contain one or two small sub-clusters of measures that agree closely with each other, but the agreement does not extend to the class as a whole.

D.5. Combinations of measures

Motivated by the possibility that complementary measures could help detect overlapping transitions, we evaluate three combination strategies at fixed size $K_{\text{combo}} = 3$, each fit with the protocol of Appendix C.2 on stacked values and differences. *Top* picks the three measures with the highest bag-of-states R^2 on the onset target; *Same-cluster* draws random triples from within each cluster of the measure-only ARI matrix; *Diff-cluster* draws triples across clusters. We focus on onset since it was the harder target in §5.1.

Clustering measures. We recompute Ward-linkage hierarchical clustering on the measures-only $1 - \text{ARI}$ submatrix from Section D.4. The behavioral anchors stay out of the combinatorial pool, so the linkage is over the 53 individual measures. We sweep the cluster count N from 2 to 10, score each N with the silhouette on the precomputed distance, and pick N by a knee rule that takes the largest N whose silhouette stays within 20% of the sweep maximum, capped at MAX_N $= 8$. The knee rule matters because the strict argmax sometimes lands at $N = 2$ in three-language multilingual, which leaves only

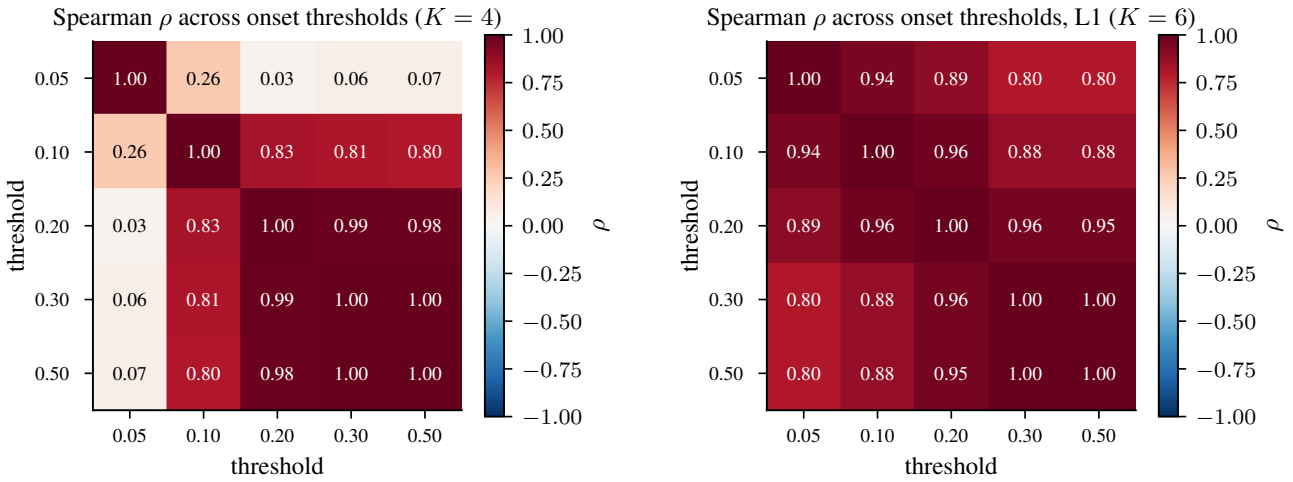

Figure 12. **Robustness of the per-subset ranking to the onset threshold.** Spearman ρ between per-subset R^2 rankings produced by each onset threshold in $\{0.05, 0.10, 0.20, 0.30, 0.50\}$. Rankings stay close for thresholds at or above 0.10; the 0.05 threshold drifts on modular addition but stays close to the others on multilingual L1. *(Left)* Modular addition ($K = 4$). *(Right)* Multilingual modular addition, L1 onset ($K = 6$).

one cluster-triple to draw diff-cluster combinations from. Even with the knee rule both settings land at $N = 3$.

Combination strategies. We form three-measure combinations under three strategies. *Same-cluster* draws five random triples from each cluster of size at least three, testing whether redundant measures combine usefully. *Diff-cluster* draws five random representative-triples from each cluster-triple, with one member per cluster, testing whether complementary measures combine usefully. *Top* picks the strict three-measure subset with the highest individual bag-of-states R^2 on the focal target, plus five random triples from the top-10 pool to give the same distributional treatment as the cluster-based strategies. Onset is the focal target in both settings, with the per-language onsets used in three-language multilingual and one set of top combinations picked per language. Each combination is fit as a multivariate Gaussian HMM with the same protocol as Appendix C.2.

Evaluation and findings. We score each combination with the bag-of-states ridge R^2 on the retrospective onset target, using the same protocol as Appendix C.3. We benchmark against two references, the best single individual on the same target and the distribution of individual R^2s in the same source pool the combination was drawn from. The source-pool comparison is the more relevant of the two since the question is whether the joint HMM extracts something its members do not already carry on their own. Figure 20 shows the source-pool comparison per strategy. In both settings no combination reaches the best single individual. The combination distributions overlap with the source-pool distributions rather than separating cleanly above them, although in three-language multilingual we do see a small but consistent lift over the pool median in most panels. The lift, where it appears, is well below the gap to the best single in absolute terms. The setup is small in scope and we view it as an initial investigation rather than a confident statement on whether combinations help.

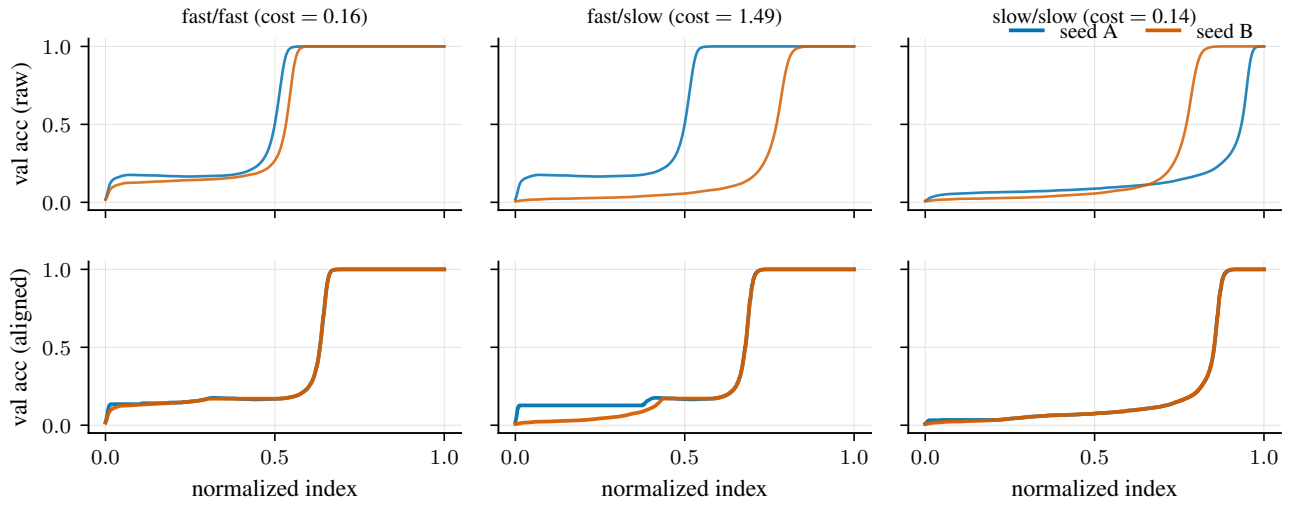

Figure 13. **DTW alignment for modular addition.** Three diagnostic seed pairs: two earliest grokkers, earliest vs latest, and two latest. Top row: raw `val_acc` curves on each seed's own normalized epoch index. Bottom row: same curves re-indexed along the DTW warp path. The DTW input was the 2-D `[train_acc, val_acc]` signal; we visualize `val_acc` only here for readability. Lower cost indicates a closer alignment.

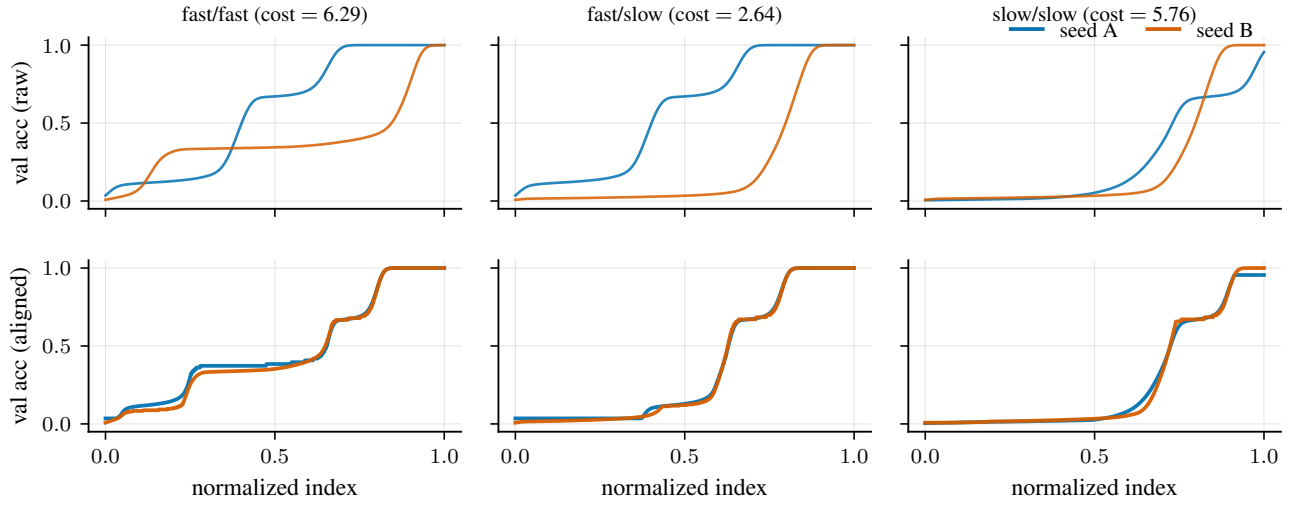

Figure 14. **DTW alignment for three-language multilingual.** Same layout as Figure 13, with DTW computed on the 3-D `[val_L1, val_L2, val_L3]` per-language signal. We visualize the aggregate `val_acc` for readability.

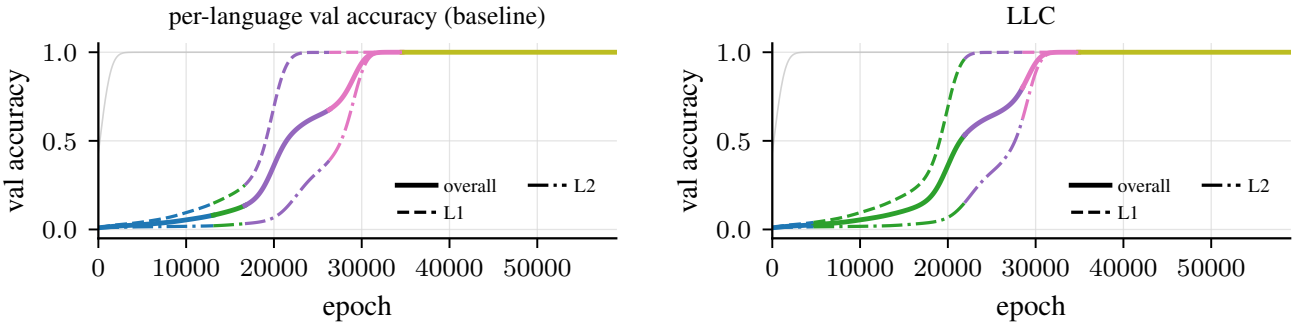

Figure 15. **Inferred phases overlaid on validation accuracy, two-language multilingual** ($K = 6$)**.** Same layout as Figure 3. *(Left)* Per-language validation-accuracy reference. *(Right)* LLC.

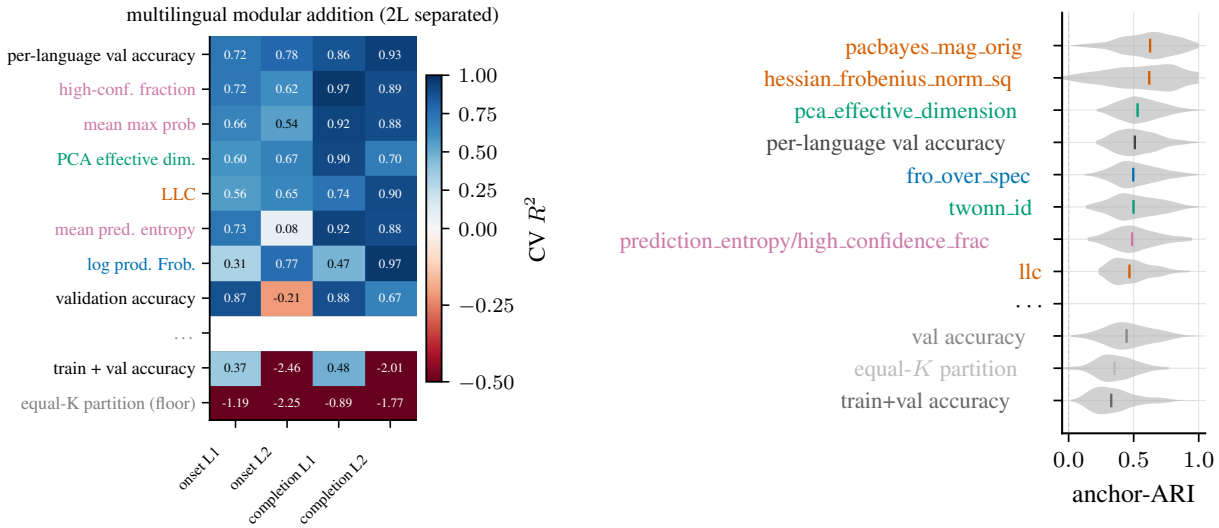

Figure 16. **Two-language multilingual at $K = 6$.** *(Left)* Retrospective bag-of-states R^2 analogue of Figure 4: top-8 subsets ranked by mean R^2 across the per-language onset and completion targets, with non-top references below the '...' separator. Cells are 5-fold CV Ridge R^2, clipped at $[-0.2, 1.0]$ with a diverging colormap centered at zero. *(Right)* Seed-consistency analogue of Figure 17: per-seed-pair DTW-aligned ARI for the top 8 measures, with the behavioral references and equal-K floor as their own block.

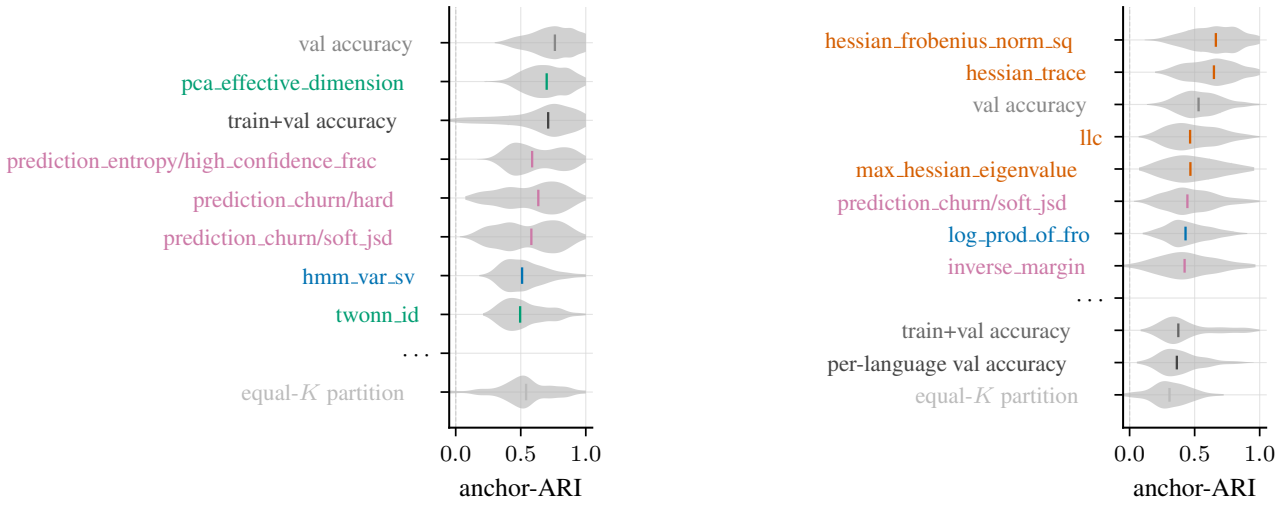

Figure 17. **Seed consistency at fixed K.** Each violin shows the per-seed-pair DTW-aligned ARI distribution for one subset, with the median tick highlighted. We show the top 8 subsets by mean ARI, ranking measures and behavioral references together, with references outside the top 8 and the equal-K partition floor as their own block below the separator; the floor's positive ARI comes from DTW partially aligning even an equal-length partition, and floors the displayed top-8 rather than the full measure pool. *(Left)* Modular addition ($K = 4$): the validation-accuracy reference leads the ranking, with only the PCA effective dimension between it and the train-and-validation reference, all well separated from the equal-K floor. *(Right)* Three-language multilingual ($K = 6$): the validation-accuracy references no longer separate from the top measures, indicating that phase identity is unstable across seeds in this setting. The two-language analogue is in Figure 16.

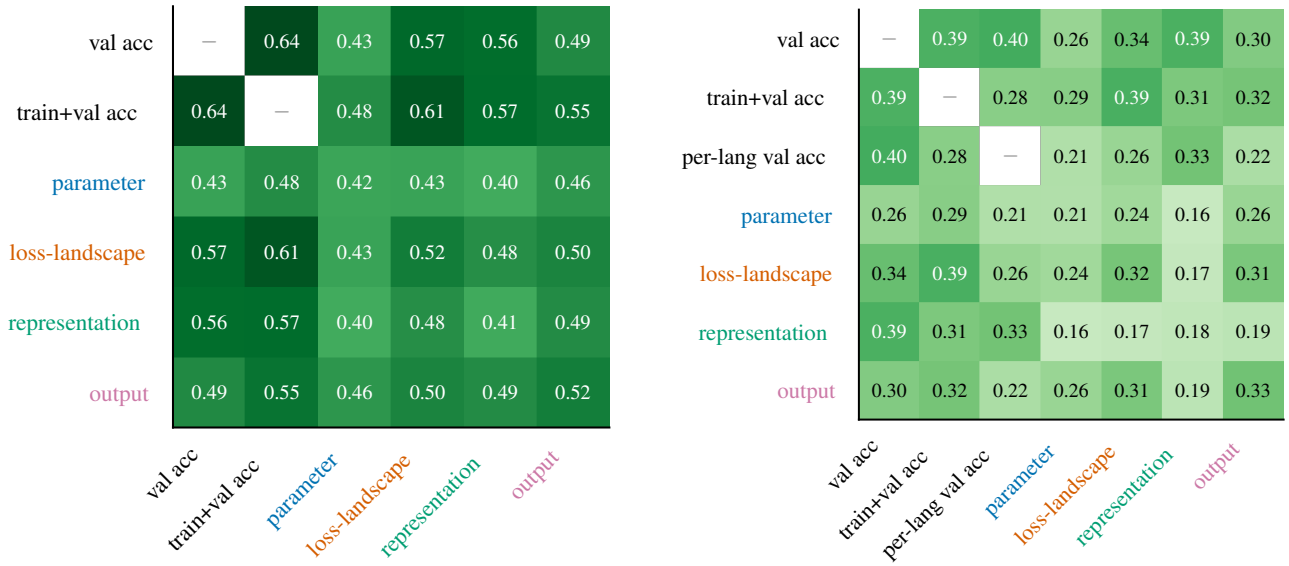

Figure 18. **Block-median ARI between measure classes and behavioral references.** Within-class agreement is not higher than across-class agreement, indicating that our measure-class definitions do not capture genuinely distinct phase structure. Each cell is the median pairwise ARI between members of two classes, or between a class and a reference. The pattern holds in both *(left)* modular addition and *(right)* three-language multilingual, at lower overall ARI in the latter. The two-language analogue is in Figure 19.

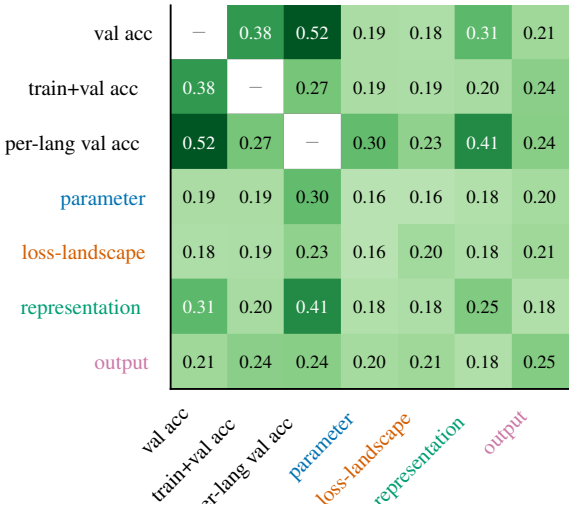

Figure 19. **Block-median ARI between measure classes and behavioral anchors, two-language multilingual** ($K = 6$)**.** Two-language analogue of Figure 18. Within-class and across-class blocks sit at similar levels, with the same flat pattern as the modular addition and three-language panels.

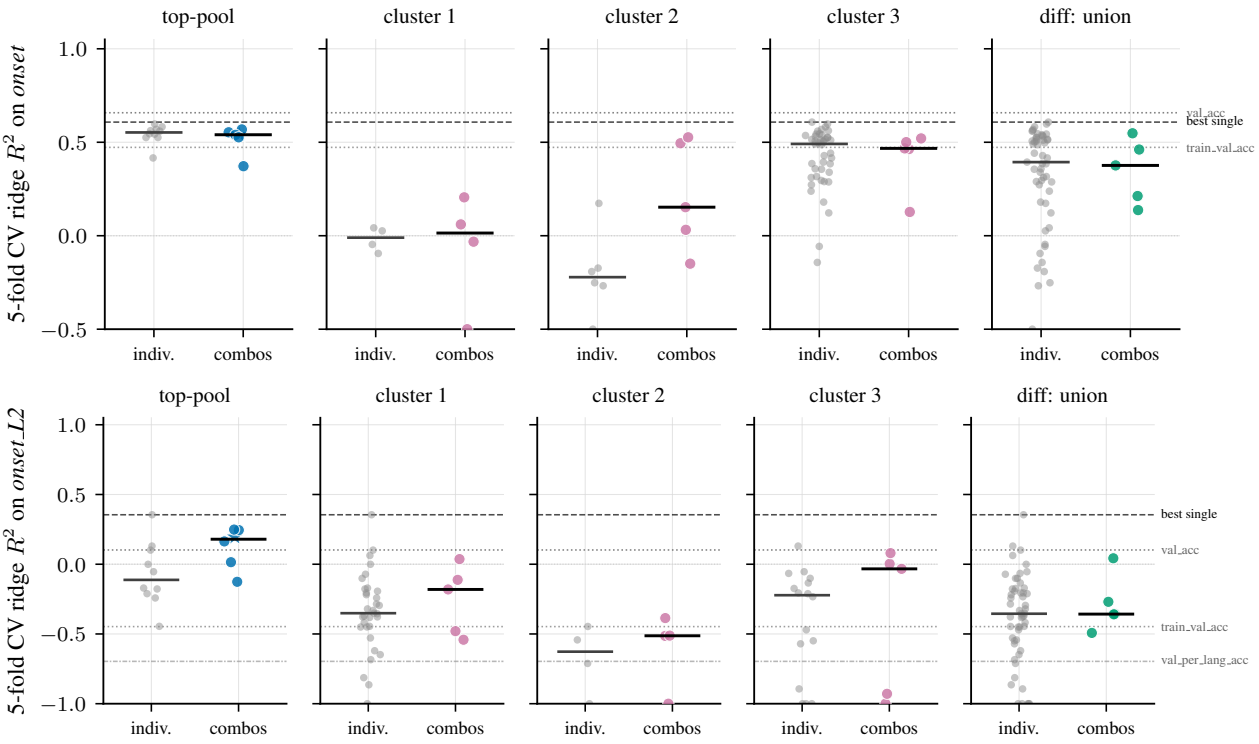

Figure 20. **Combinations vs the individuals they were drawn from.** Combination R^2 in color and source-pool individual R^2 in gray, panel by panel. Wide ticks mark the medians. The dashed line marks the best single individual on the same target, dotted lines mark the behavioral anchors. No combination reaches the best-single line. Combinations overlap with the source-pool distributions, with a small but consistent lift over the pool median in most multilingual panels and a more mixed pattern in modular addition. *(Top)* Modular addition, onset target. *(Bottom)* Three-language multilingual modular addition, `onset_L2` target.

E. Detailed relations to the literature

E.1. Multilingual learning

Modern LLMs are trained on texts from multiple languages. Mechanistic interpretability methods provide evidence that real-world LLMs share certain representations of concepts across multiple languages; even two texts that do not share a single token can yield very similar representations if they express identical semantic content (Wendler et al., 2024; Lindsey et al., 2025; Dumas et al., 2025; Brinkmann et al., 2025; Veselovsky et al., 2025). Heiss (2024, Section 1.6) argues theoretically that weight decay induces an inductive bias towards learning such shared representations, based on mathematical theory from (Heiss et al., 2021). Furthermore, it is commonly observed that adding more training data in one language improves downstream performance in other languages (Schäfer et al., 2024; Conneau et al., 2020; Pires et al., 2019; Muennighoff et al., 2023).

To obtain more controlled experiments, following (K et al., 2020; Dufter & Schütze, 2020), Schäfer et al. (2024) conducted semi-synthetic experiments using cloned languages. Because real-world multilingual datasets are inherently messy and naturally contain overlapping vocabularies, they explicitly duplicated the original token vocabulary. This allowed them to express each text either in the original language (using the original tokens) or in the cloned language (using the cloned tokens), guaranteeing that absolutely no tokens are shared across the languages. While anchor points (e.g., shared tokens across languages, such as "computer" in English and German) are known to facilitate cross-lingual generalization, Schäfer et al. (2024) systematically demonstrate that cross-lingual generalization can occur even in this extreme case of completely disjoint token vocabularies, as theoretically suggested by Heiss (2024, Section 1.6). We introduce the new synthetic dataset of multilingual modular addition by adopting this approach, replacing real-world language data with simple modular addition data. Our goal is to create a minimally complex dataset that preserves these qualitatively similar phenomena. This simplicity aids mechanistic interpretability and significantly reduces the computational costs of training and inference, enabling extensive experimentation across hyperparameter configurations and making the pipeline accessible to a wider range of researchers.

Multilingual modular addition allows for compositional representations in the mechanistic sense described by Olah (2023). Under this perspective, a representation is considered compositional if it decomposes a concept like "blue circle" into independent properties ("blue" and "circle"). This enables non-local generalization; for example, seeing "green circles" and "blue squares" allows a model to generalize to "green squares". For multilingual datasets, a sequence of tokens can be decomposed into its abstract content and its language. If the model learns to decouple these, extracting the content and language in early layers, computing the arithmetic ($a + b \bmod p$) independent of the language, and ultimately composing the result with the target language in later layers (see Heiss, 2024, Example 1.6.2), it achieves non-local generalization. Such a circuit structure implies that any capability learned in one language automatically transfers to another, even if they share no tokens. Recent empirical work provides further evidence for this decoupled structure in real models. For instance, Bandarkar et al. (2025) achieve zero-shot cross-lingual transfer for mathematical reasoning by taking an LLM trained on English math and swapping its top and bottom layers with those of an LLM trained on the target language.

However, models do not inherently have to learn this specific circuit structure. Heiss (2024, Section 1.6) notes that weight decay induces an inductive bias towards shared representations only for sufficiently deep models, and only if the complexity of translation is sufficiently low compared to the complexity of the core task. Even when these conditions are met, gradient descent may still get stuck in suboptimal local minima. Intuitively, wider layers should reduce the likelihood of converging to bad local minima; for two-layer MLPs, it is proven that the probability of getting stuck in a bad local minimum vanishes asymptotically as the number of neurons per layer tends to infinity (Chizat & Bach, 2018). For deeper architectures, larger models are more likely to contain highly performant sub-networks (winning lottery tickets) at initialization (Frankle & Carbin, 2018). Empirically, Lindsey et al. (2025) observe that Anthropic's smaller models share fewer representations across languages, whereas extensive sharing emerges in their larger models. Schäfer et al. (2024) similarly observe that increasing model scale improves cross-lingual generalization. While one could argue that a computational bottleneck (fewer neurons per layer) might force a model to share concepts, this often makes it harder for gradient descent to locate these solutions in practice. Theoretical results (Heiss et al., 2021; Heiss, 2024) confirm that even models with infinite width maintain an inductive bias to share representations when weight decay is applied. Future work should systematically explore which hyperparameter choices strengthen or weaken these inductive biases.

Closest to our synthetic multilingual modular addition setting in the broader phase-detection literature, Xu (2026) studies staggered grokking in shared-trunk Transformers trained on multiple distinct modular operations (addition, multiplication, squaring). Our setting differs in keeping the underlying operation identical across languages, which matches per-language

difficulty, and in providing a tunable overlap knob via per-language data fractions. On a conceptual level, the synthetic datasets suggested by (Caruana, 1997; Heiss et al., 2020; 2021; Shai et al., 2026) are similar in that they encourage representations to be shared across multiple sub-tasks, though they are not explicitly framed around multilingual data.[2] Ultimately, any synthetic or real dataset can be adapted into a multilingual format to study these representational dynamics.

E.2. Detailed Related Work on Compositionality

Formalizations and Generalization. Compositionality is the property by which a finite set of concepts and rules can combine to produce a vast range of outputs. It underlies both natural language and systematic generalization in learned systems (Fodor & Pylyshyn, 1988; Lake et al., 2017; Lake & Baroni, 2018), and capturing the compositional structure of the data lets models handle novel combinations of previously seen concepts. A key consequence of this property is the ability to generalize to unseen combinations of known concepts. For example, as Elmoznino et al. (2024) note, if a model is compositional with respect to a set of features in its training data, it need not observe all possible combinations of those features in order to generalize to novel ones (Bahdanau et al., 2019; Mittal et al., 2022; Schug et al., 2024; Wiedemer et al., 2024; 2023).

To capture these dynamics, compositionality has been formalized in several complementary ways: as a compression ratio grounded in Kolmogorov complexity (Elmoznino et al., 2024), as a property of the computation graph induced by problem decomposition (Dziri et al., 2023), and mechanistically through the lens of representation geometry (Olah, 2023). As discussed in Appendix E.1, Olah (2023) considers a representation compositional if it disentangles a concept into independent, composable properties (e.g., decomposing a "blue circle" into "blue" and "circle"). In our multilingual modular addition setting, this ideal compositional structure would correspond to the model decoupling a token "(L1:27)" into its mathematical meaning "27" and its language "L1". Furthermore, even for classical modular addition, a token's representation can be decomposed into its Fourier frequencies. While the structure of multilingual modular addition allows a model to nest these two types of decompositions in principle, whether standard transformers actually converge to such perfectly decoupled representations depends heavily on their inductive biases and training dynamics (see Appendix E.1).

Behavioral Properties of Compositionality. From a behavioral perspective, Hupkes et al. (2020) formalize compositionality through five empirical tests: *systematicity*, *productivity*, *substitutivity*, *localism*, and *overgeneralization*. Our proposed task of multilingual modular addition serves as an excellent testbed precisely because its underlying structure isolates and evaluates several of these properties:

- **Systematicity** tests whether a model can recombine constituents that have not been seen together during training. (Multilingual) modular addition directly evaluates this: the transition from memorization to generalization (grokking) occurs when the model systematically applies the learned modulo arithmetic rule to unseen combinations of tokens from the $P \times P$ grid. In our case, we can directly measure this via the validation accuracy, as the validation set consists entirely of unseen combinations of tokens.

- **Substitutivity**, as formulated by Hupkes et al. (2020), tests whether a model's behavior is robust to synonymous substitutions, which mathematically corresponds to *invariance* under synonym replacement. Our multilingual variant tests a related, arguably more complex property: *equivariance* under language translation. In multilingual modular addition, the true target function is equivariant with respect to the language; translating the input tokens to a different language requires the output to shift into that corresponding language, even though the underlying mathematical operation remains identical. The validation accuracy of multilingual modular addition inherently tests for this equivariance, which is naturally satisfied if the model learns to structurally decouple the mathematical content from the language identity as in Heiss (2024, Example 1.6.2). Future work could easily introduce pure synonyms into our setup (e.g., multiple disjoint input tokens within the same language mapping to the exact same output token) to explicitly test for invariance alongside this equivariance.

- **Localism** (requiring operations to depend only on immediate constituents), **Productivity** (the ability to extrapolate to longer sequences), and **Overgeneralization** (favoring systematic rules over memorized exceptions) are not directly evaluated as learning challenges in our current setup. Localism is satisfied trivially by definition, as modular arithmetic

[2] Historically, Rumelhart et al. (1986) proposed a conceptually similar dataset dealing with two isomorphic family trees, which could be viewed as two cloned languages describing the same structure. However, the inputs to their models did not use completely disjoint coordinates, as the family relations were represented by the same input coordinates, independent of the specific family tree.

inherently lacks broader sequence context. Meanwhile, productivity and overgeneralization are not tested because the equations have a fixed length and the underlying target function has no true exceptions. Future work could easily extend our setting to allow for these tests, for instance, by evaluating the model on longer compositional sequences or by introducing artificial exceptions to the training data.

Compositionality and Training Phases. Because of these properties, compositional tasks are a particularly useful testbed for studying training phases. Their modular structure invites the hypothesis that concepts and rules are acquired in distinct stages (Lubana et al., 2024). In fact, compositional structure has been shown to be a driver of sudden capability acquisition (Okawa et al., 2023; Arora & Goyal, 2023; Hoffmann et al., 2023). Modular arithmetic is a tractable instance of this, where the task requires learning discrete numerical representations and a specific algebraic combination rule, making the transition from memorization to generalization sharp and interpretable (Power et al., 2022; Nanda et al., 2023; Chughtai et al., 2023; Kangaslahti et al., 2025; Zhao et al., 2026). By expanding this setup to multilingual modular addition, we induce sequential capability transitions across languages. This allows us to track these overlapping phases with task-agnostic measures, while providing a foundation for future mechanistic studies into how models iteratively construct shared compositional representations.

E.3. Further related work on Grokking and modular addition

Modular Arithmetic as a Testbed. The phenomenon of delayed generalization, or "grokking", was initially popularized in the context of neural networks learning small algorithmic tasks, most notably modular arithmetic (Power et al., 2022). While grokking has since been shown to be a general phenomenon driven by representation complexity across diverse, non-algorithmic datasets involving images or language (Liu et al., 2023), modular addition remains the standard "fruit fly" model for studying capability emergence. This is because its underlying simplicity and exact mathematical ground truth make the transition from memorization to generalization both sharp and fully mechanistically tractable. Mechanistic interpretability work has shown that this transition typically corresponds to the network discarding memorized solutions in favor of structured trigonometric algorithms based on Fourier features (Nanda et al., 2023). By extending this foundational setup into a multilingual variant, we introduce just enough compositional complexity to study how multiple learned mechanisms overlap and interact during training, providing a richer testbed for future mechanistic work.

Recent Structural and Optimization Insights. Recent work has increasingly analyzed the structural drivers and representational geometry of grokking. From a structural perspective, Singh et al. (2026) demonstrate that inductive biases from Transformer architecture (e.g., Layer Normalization position) and optimization settings strongly modulate grokking speed. From a geometric perspective, Sakamoto & Sato (2026) theoretically link grokking to Neural Collapse. They prove that generalization depends on the contraction of the population within-class variance—a geometric process driven by weight decay that occurs much slower than fitting the training data, naturally explaining delayed capability jumps.

Generalization to Broader Group Structures. While canonical grokking studies focus on cyclic groups via modular addition, recent work extends these insights to broader algebraic settings. Empirically, Chughtai et al. (2023) showed that networks learning operations on finite groups develop features corresponding to irreducible group representations. Morwani et al. (2024) explained this analytically: for modular addition, sparse parities, and finite group operations, margin maximization alone determines the features learned—Fourier features in the modular case and irreducible representations in the group-operation case. Building on the optimization-dynamics side, Kunin et al. (2026) characterize the training dynamics of two-layer networks on modular addition, proving that networks learn Fourier features in decreasing order of coefficient magnitude. Most recently, Marchetti et al. (2026) introduced the Sequential Group Composition task for arbitrary finite groups, showing that two-layer networks acquire one irreducible representation at a time, in an order determined by the Fourier statistics of the encoding. Just as we introduce multilingual modular addition to study overlapping phase transitions across shared concepts, future work could naturally adapt these broader group and sequential operations into multilingual formats.

E.4. Further related work on phase transitions and task-agnostic measures

Xiao et al. (2025) introduce Neuron-based Multifractal Analysis (NeuroMFA), a framework that derives task-agnostic structural measures from neuron interaction graphs and links them to the emergence of capabilities in large language models. This shares our goal of detecting capability transitions through task-agnostic measures rather than downstream evaluations, but the measures all come from a single multifractal-analysis framework. In contrast, we compare a broad suite of measures

drawn from different aspects of the model (parameters, loss landscape, representations, and outputs) on small transformers with controlled, ground-truth phase transitions, allowing direct quantitative comparison of which measures recover known phase structure. Future work could add structural measures like NeuroMFA to our suite and apply them to multilingual modular addition to test whether they help disentangle overlapping phase transitions.