
Local Learning Coefficients Predict Developmental Milestones During Group Relative Policy Optimization¹

Jeremias Ferrao Ilija Lichkovski
AI Safety Initiative Groningen

With
PIBBSS, Timaeus, & Apart Research

Abstract

In this work, we investigate the emergence of capabilities in reinforcement learning (RL) by framing them as developmental phase transitions. We propose that the individual components of the reward function can serve as direct observables for these transitions, avoiding the need for complex, derived metrics. To test this, we trained a language model on an arithmetic task using Group Relative Policy Optimization (GRPO) and analyzed its learning trajectory with the Local Learning Coefficient (LLC) from Singular Learning Theory. Our findings show a strong qualitative correlation between spikes in the LLC—indicating a phase transition—and significant shifts in the model’s behavior, as reflected by changes in specific reward components for correctness and conciseness. This demonstrates a more direct and scalable method for monitoring capability acquisition, offering a valuable proof-of-concept for developmental interpretability and AI safety. To facilitate reproducibility, we make our code available at github.com/ilijalichkovski/apart-physics.

Keywords: Physics-informed AI safety, Singular Learning Theory (SLT), Developmental Interpretability, Reinforcement Learning (RL)

1 Introduction

The emergence of complex capabilities in Large Language Models (LLMs) often occurs not as a gradual process, but through abrupt, stage-wise shifts analogous to phase transitions in physical systems (Olsson et al. 2022). The nascent field of developmental interpretability (DI) aims to understand these phenomena, which are often invisible when observing the model’s loss function alone, as this scalar metric typically decreases smoothly and fails to capture the underlying structural reorganization. A key framework in this endeavor is Singular Learning Theory (SLT), which provides metrics like the Local Learning Coefficient (LLC) to characterize the geometry of the loss landscape and detect developmental stages during model training (Hoogland et al. 2025). To date, identifying these stages has required a suite of auxiliary metrics, such as bigram

¹Research conducted at the AI Safety x Physics Grand Challenge, 2025

scores, previous-token matching, and complex measures of circuit composition like the formation of induction heads. While powerful, these metrics are byproducts of the learning process rather than its explicit goals, making their identification a non-trivial reverse-engineering task.

This reliance on imperfect metrics introduces a significant challenge: the metrics needed to observe phase transitions can be difficult to define, and their scalability to more complex tasks remains an open question. This work proposes a novel approach for studying these transitions within the context of Reinforcement Learning (RL)—a training regime of critical importance for modern AI systems, yet largely unexplored through a DI lens. We hypothesize that the multi-component reward function in RL provides a natural set of observables for detecting phase transitions. This reframes the problem from studying phenomena akin to quantum phase transitions (Vojta 2003) —which require specialized, non-obvious metrics for observation, to those resembling classical phase transitions like ice melting or water turning to steam, where changes can be measured using readily available metrics like temperature and pressure.

Unlike the derived metrics used in prior work, the reward function is the explicit objective of the optimization process. In many practical applications, this reward is not monolithic but a composite signal, directly encoding desired behaviors such as output correctness, structure, or output length. We argue that monitoring these fundamental components of the reward signal offers a simpler, more direct, and more scalable method for observing the developmental stages of a model as it acquires new capabilities.

To test this hypothesis, we trained a model using Group Relative Policy Optimization (GRPO) (Shao et al. 2024), on a small, synthetic arithmetic task of varying difficulty created for this project. Our investigation seeks to determine if patterns in the reward function correlate with established indicators of phase changes, such as spikes in the LLC. This research bridges DI with RL, proposing the reward function as a window for understanding capability emergence. Our preliminary findings show a promising correlation between reward dynamics and LLC spikes, laying the groundwork for future causal analyses into how models learn complex, multi-stage tasks. This approach could ultimately lead to more practical methods for monitoring capability gains during safety-critical RL fine-tuning processes.

2 Methods

Theoretical Framework We utilize singular learning theory (SLT) to frame the emergence of capabilities during RL training. If the model parameters act as microstates in accordance to an energy (loss) landscape, phase transitions correspond to rapid reorganizations of the internal structure as a consequence of the particular geometry of the loss landscape. In particular, the idea of degeneracy in parameter space naturally lends itself to describing phase transitions, since the sharp changes in the loss surface it results in lead to step-changes in internal model structure during training. The key theoretical framework we utilize to quantify the characteristics of internal model structure is the localized learning coefficient (LLC) (Lau et al. 2024), defined as the local average over a neighborhood of weights $\hat{w}^*$ of the number of entries n times the objective function L . Termed $\hat{\lambda}$, the expression for computing the LLC is given by

$$\hat{\lambda}(\hat{w}^*) = \frac{E_{w|\hat{w}^*, \gamma}^{\beta^*} [nL_n(w)] - nL_n(\hat{w}^*)}{\log n}, \quad (1)$$

and is typically approximated using an optimizer like Stochastic Gradient Markov Chain Monte Carlo (Nemeth and Fearnhead 2019) or Stochastic Gradient Langevin Dynamics (Wingerden et al. 2024).

The dynamics of the localized learning coefficient, by revealing the structure of the loss geometry, hints at phase transitions during training. Namely, when a model starts a rapid internal reorganization process, this is characterized by the loss associated with a local neighborhood of weights $\hat{w}^*$ exhibiting strong non-uniformities. By quantifying these phase-transitions enabled by the particular loss-landscape geometry as a function of time, the localized learning coefficient provides a principled measure of key developmental milestones during model training.

Experimental Design Our empirical investigation probed a 0.5 billion-parameter instruction-tuned language model (Qwen-3-0.6B) during the process of learning basic arithmetic operations. Our procedurally-generated dataset consists of 456 arithmetic tasks of progressive difficulty. We defined 4 rewards to guide the training process: a format reward incentivizing the model to provide answers within XML tags, a reward for providing a numerical answer (`float` or `int`), a length penalty incentivizing more concise answers, and a correctness reward. The definition of these rewards is provided in the Appendix (Section 5.2)

We employed Group Relative Policy Optimization — a reinforcement learning algorithm building on Proximal Policy Optimization (PPO) (Schulman et al. 2017) — in order to train our model on efficiently performing arithmetic tasks. Unlike PPO, GRPO removes the need for a value model by instead comparing rollouts within a group to calculate the advantage, and replaces the reward model with a deterministic verifier (which has made it attractive for verifiable tasks). The full set of hyperparameters used during the GRPO training can be found in Table 5.1 in the Appendix along with our prompts.

To compute the LLC, we used the Stochastic Gradient Markov Chain Monte Carlo (SGMCMC) sampling functionality in the `devinterp` library (Wingerden et al. 2024) using hyperparameters specified in the Appendix 2. This method explores the local proximity around the parameters of each model checkpoint. We processed all available checkpoints (saved every 10 training steps) to find the mean LLC across all post-burn-in samples and the corresponding standard deviation from the full trace data. This gives a chronological view of the evolution of the localized learning coefficient during key training milestones. The SGMCMC algorithm for sampling the loss landscape requires hyperparameters like the SGMCMC learning rate and inverse temperature, which were found using a grid sweep and a manual choosing of the optimal values based on convergence quality. Finally, to pinpoint crucial milestones during training, we compared LLC minima and inflections in the reward curve. Such a correspondence indicates that the training process has developed novel circuits in the network.

3 Results

Our primary empirical finding is the strong qualitative correlation between periods of high Local Learning Coefficient (LLC) volatility and tangible shifts in the model’s behavior, as measured by the components of the reward function. Figure 1 illustrates this relationship. We identified three suspected phase transitions, highlighted by shaded regions, where spikes in the LLC coincide with critical moments in the learning trajectory.

The first transition, between training steps 44 and 66, is characterized by a sharp decrease in the LLC. This instability corresponds to the model overcoming a significant, deliberately imposed challenge. While the base model can perform the required arithmetic, we constrained the maximum completion length to increase the task’s difficulty and reduce training time. This first phase transition thus marks the point where the model learns to formulate its answer concisely enough to fit within the token limit. This is evidenced by the gradual increase in

the "Numeric Reward" (as answers become parsable) and the "Length Penalty Reward" (as answers are no longer prematurely truncated).

A second transition occurs between steps 88 and 110. Here, another LLC spike precedes a general flattening of the reward curves. This suggests a consolidation of learned skills, where the model shifts from rapid capability acquisition to a more stable phase of refinement. The final observed transition, between steps 187 and 209, is notable for an LLC spike that aligns with a sudden drop in the "Correctness Reward," while other rewards remain high. This may represent a brief period of policy degradation or exploration before the model recovers.

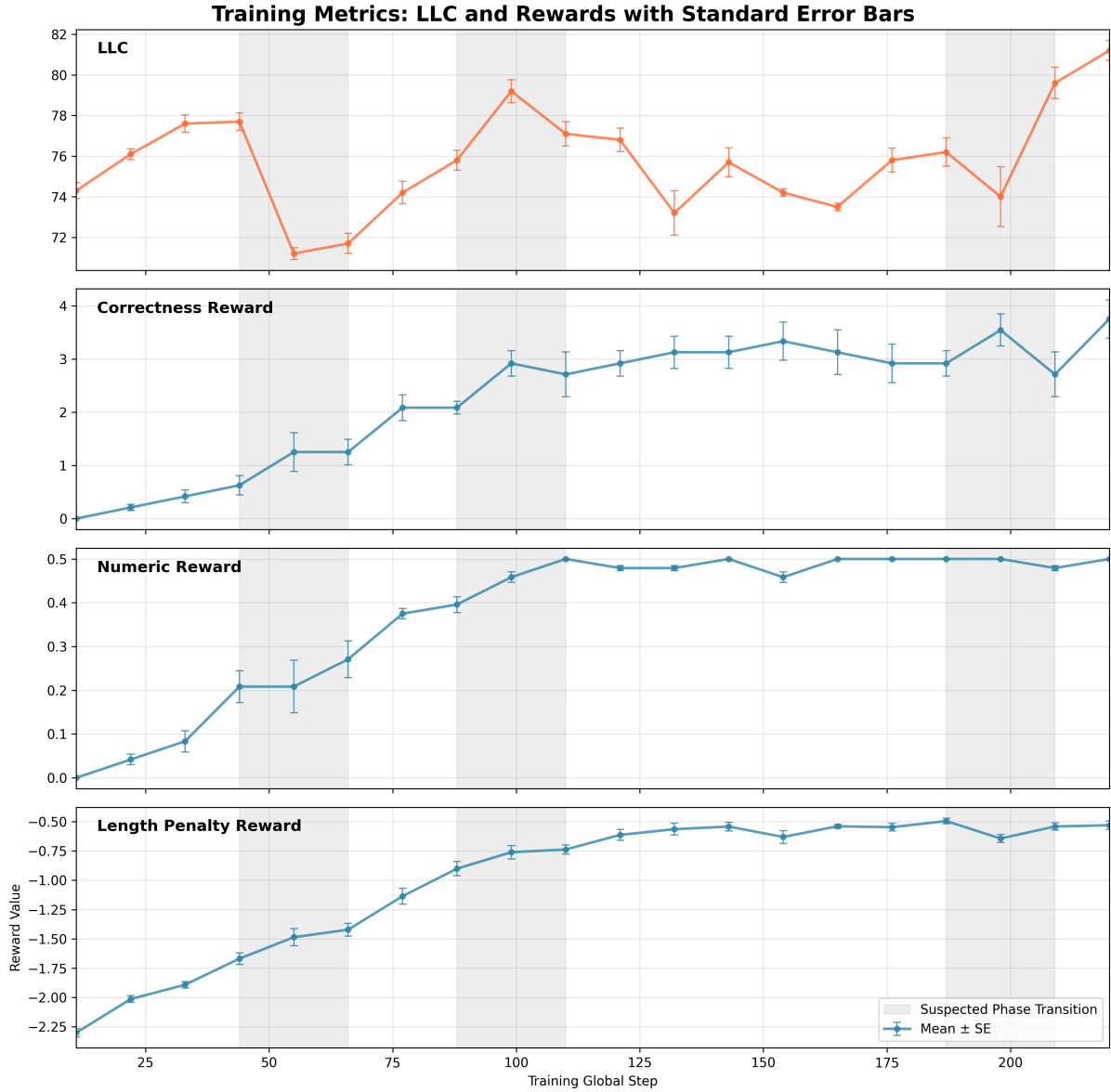


Figure 1: Training metrics over 220 global steps. The top plot shows the Local Learning Coefficient (LLC), with spikes indicating suspected phase transitions (shaded regions). The subsequent plots show three components of the reward function: Correctness Reward, Numeric Reward, and Length Penalty Reward. The alignment of LLC spikes with distinct changes in the reward curves suggests that reward components are effective observables for developmental transitions.

4 Discussion and Conclusion

Our results demonstrate a promising correlation between LLC dynamics and reward-based observables in a reinforcement learning context, suggesting that the explicit goals of the training process can serve as a powerful lens for developmental interpretability. The alignment of LLC spikes with critical learning events—such as overcoming a token limit constraint or consolidating a new skill—supports the hypothesis that these mathematical markers from Singular Learning Theory correspond to meaningful, tangible changes in a model’s capabilities. This provides a valuable, proof-of-concept for applying physics-inspired techniques to better understand and monitor AI development during safety-critical fine-tuning.

Limitations It is crucial to acknowledge the limitations of this study. The findings are based on a single model architecture and a small, synthetic dataset (456 training samples). Therefore, the specific patterns observed may not generalize to other tasks or larger models. Most importantly, our analysis establishes a correlation, not a causal link. The observed LLC spikes and reward shifts are concurrent phenomena, but we have not shown that one directly causes the other.

Future Directions To build upon these findings and establish causality, a mechanistic analysis is essential. Future work should aim to “look inside the box” during these identified phase transitions. Techniques from mechanistic interpretability are particularly well-suited for this. For instance, one could employ sparse autoencoders to discover highly interpretable features within the model and investigate whether specific features activate or change during the LLC spikes (Cunningham et al. 2023). Furthermore, automated circuit discovery could be used to identify the specific neural pathways responsible for the newly acquired capabilities, verifying that the observed phase transitions correspond to the formation of new, functional circuits within the model (Conmy et al. 2023). Such an analysis would bridge the gap from correlation to causation, solidifying the link between the mathematical theory and the model’s internal workings. As new evidence suggests that RL can promote better generalization in models than traditional supervised finetuning (Chu et al. 2025), understanding the mechanics of how RL achieves this is of great interest, both for capabilities and safety research. Given the suitability of metrics like the LLC in characterizing internal model phase transitions that occur during grokking (Power et al. 2022), it is of interest to observe LLC dynamics during longer convergent GRPO training runs. Furthermore, building on our demonstration of LLC’s ability to predict step-changes in skill acquisition rates in language models, we find it particularly fruitful to use SLT techniques to predict the emergence of compositional generalization capabilities in models when faced with novelty in-context.

Impact Assessment: If proven robust at larger model/data/compute scales, this approach could enable a contextualized view of capability emergence during reinforcement learning. Perhaps more importantly, it allows alignment researchers to reliably probe models for misaligned behavior and trace back the reward signals that might have incited it. We hope that reframing reward not solely as a training signal, but also as a diagnostic tool when employed with frameworks from SLT, can help drive the AI field towards more interpretability and alignment with human values.

References

- Chu, Tianzhe et al. (2025). *SFT Memorizes, RL Generalizes: A Comparative Study of Foundation Model Post-training*. arXiv: [2501.17161](https://arxiv.org/abs/2501.17161) [cs.AI]. URL: <https://arxiv.org/abs/2501.17161>.
- Conny, Arthur et al. (2023). *Towards Automated Circuit Discovery for Mechanistic Interpretability*. arXiv: [2304.14997](https://arxiv.org/abs/2304.14997) [cs.LG]. URL: <https://arxiv.org/abs/2304.14997>.
- Cunningham, Hoagy et al. (2023). *Sparse Autoencoders Find Highly Interpretable Features in Language Models*. arXiv: [2309.08600](https://arxiv.org/abs/2309.08600) [cs.LG]. URL: <https://arxiv.org/abs/2309.08600>.
- Hoogland, Jesse et al. (2025). *Loss Landscape Degeneracy Drives Stagewise Development in Transformers*. arXiv: [2402.02364](https://arxiv.org/abs/2402.02364) [cs.LG]. URL: <https://arxiv.org/abs/2402.02364>.
- Lau, Edmund et al. (2024). *The Local Learning Coefficient: A Singularity-Aware Complexity Measure*. arXiv: [2308.12108](https://arxiv.org/abs/2308.12108) [stat.ML]. URL: <https://arxiv.org/abs/2308.12108>.
- Nemeth, Christopher and Paul Fearnhead (2019). *Stochastic gradient Markov chain Monte Carlo*. arXiv: [1907.06986](https://arxiv.org/abs/1907.06986) [stat.CO]. URL: <https://arxiv.org/abs/1907.06986>.
- Olsson, Catherine et al. (2022). *In-context Learning and Induction Heads*. arXiv: [2209.11895](https://arxiv.org/abs/2209.11895) [cs.LG]. URL: <https://arxiv.org/abs/2209.11895>.
- Power, Alethea et al. (2022). *Grokking: Generalization Beyond Overfitting on Small Algorithmic Datasets*. arXiv: [2201.02177](https://arxiv.org/abs/2201.02177) [cs.LG]. URL: <https://arxiv.org/abs/2201.02177>.
- Schulman, John et al. (2017). *Proximal Policy Optimization Algorithms*. arXiv: [1707.06347](https://arxiv.org/abs/1707.06347) [cs.LG]. URL: <https://arxiv.org/abs/1707.06347>.
- Shao, Zhihong et al. (2024). *DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models*. arXiv: [2402.03300](https://arxiv.org/abs/2402.03300) [cs.CL]. URL: <https://arxiv.org/abs/2402.03300>.
- Vojta, Matthias (Nov. 2003). “Quantum phase transitions”. In: *Reports on Progress in Physics* 66.12, pp. 2069–2110. ISSN: 1361-6633. DOI: [10.1088/0034-4885/66/12/r01](https://doi.org/10.1088/0034-4885/66/12/r01). URL: <http://dx.doi.org/10.1088/0034-4885/66/12/R01>.
- Werra, Leandro von et al. (2020). *TRL: Transformer Reinforcement Learning*. <https://github.com/huggingface/trl>.
- Wingerden, Stan van et al. (2024). *DevInterp*. <https://github.com/timaeus-research/devinterp>.

5 Appendix

5.1 Hyper-parameters

5.2 Reward Function Definitions

The model’s behavior was shaped by four distinct reward functions, which were inspired by the demonstration from (**brown2024grpodemo**). These were evaluated on the model’s completions and summed to produce the total reward signal for the GRPO algorithm.

Correctness Reward This is the primary reward for task success. It first extracts the numerical answer from the model’s completion by searching for content between `<answer>` and `</answer>` tags. If the extracted answer exactly matches the ground truth for the given arithmetic problem, the model receives a reward of **+5.0**. Otherwise, the reward is 0.0.

Parameter	Value
Learning rate	8e-6
Weight decay	0.1
Warmup ratio	0.05
LR scheduled	cosine
Optimizer	AdamW, 8-bit
Batch size	2
Gradient accumulation steps	4
# of generations	2
Max. grad norm	10
Max. completion length (tok)	512

Table 1: Hyperparameters using during RL training, performed using the Transformer Reinforcement Library (TRL) framework by Hugging Face (Werra et al. 2020)

Parameter	Value
SGMCMC learning rate	1e-5
Localization, γ	100.0
Inverse temperature, β	Default based on data
Sampling method	SGMCMC
Batch size	4
Number of chains	2
Number of burn-in draws	300
Number of post-burn-in draws	300

Table 2: Hyperparameters used for computing the Local Learning Coefficient (LLC) with the devinterp library.

Numeric Reward This function encourages the model to produce a parsable answer, even if incorrect. It checks if the extracted answer string can be converted to a number. If the extracted answer is numeric, the model receives a reward of **+0.5**. This helps guide the model toward the correct data type early in training.

Format Reward This function incentivizes the correct XML-style formatting. It awards **+0.125** for the presence of the opening tag `\n<answer>\n` and another **+0.125** for the closing tag `\n</answer>`. It also applies a small penalty (-0.001 per character) for any extraneous text generated after the closing tag.

Length Penalty To promote conciseness, this function applies a penalty proportional to the total length of the generated completion. The reward is calculated as $-0.002 \times N_{\text{chars}}$, where N_{chars} is the number of characters in the response. This discourages the model from producing unnecessarily verbose outputs.

5.3 Training Prompts

The model was trained on a synthetic dataset of arithmetic problems generated by a script. Each training example consists of a prompt and the expected answer. The prompt follows a specific chat template structure, comprising a system prompt that sets the task rules and a user prompt that contains the arithmetic problem.

The system prompt is constant for all problems and instructs the model on the required output format:

Respond ONLY with the answer to the question in the following format:

`<answer>`

`numerical_answer`

`</answer>`

Don't overthink it.

The user prompt is a single, procedurally generated arithmetic problem. The problems were created by varying the number of digits (from 5 to 8) and the number of terms (from 2 to 4) for addition operations. An example of a user prompt is:

12345 + 67890 =

For the example user prompt above, the corresponding expected answer string would be '80235', which the model is expected to place within the XML tags as specified by the system prompt.

5.4 Detailed Accuracy Performance

To further investigate model performance, we analyzed accuracy on sub-tasks of varying difficulty, defined by the number of digits in the operands and the number of terms in the expression. Figure 2 shows the model's accuracy on these different task configurations throughout training. While this detailed view reveals complex learning dynamics for specific problem types, no clear, generalizable patterns emerged across the different difficulty levels that could be robustly interpreted from this data alone. The learning trajectories appear highly idiosyncratic to the specific number of digits and terms.

Additional Materials Large language models were used to assist in this research. Specifically, Google Gemini 2.5 Pro and Anthropic's Claude 4 Sonnet were utilized for debugging Python code for data generation and analysis, as well as for assistance in formatting sections of this report.

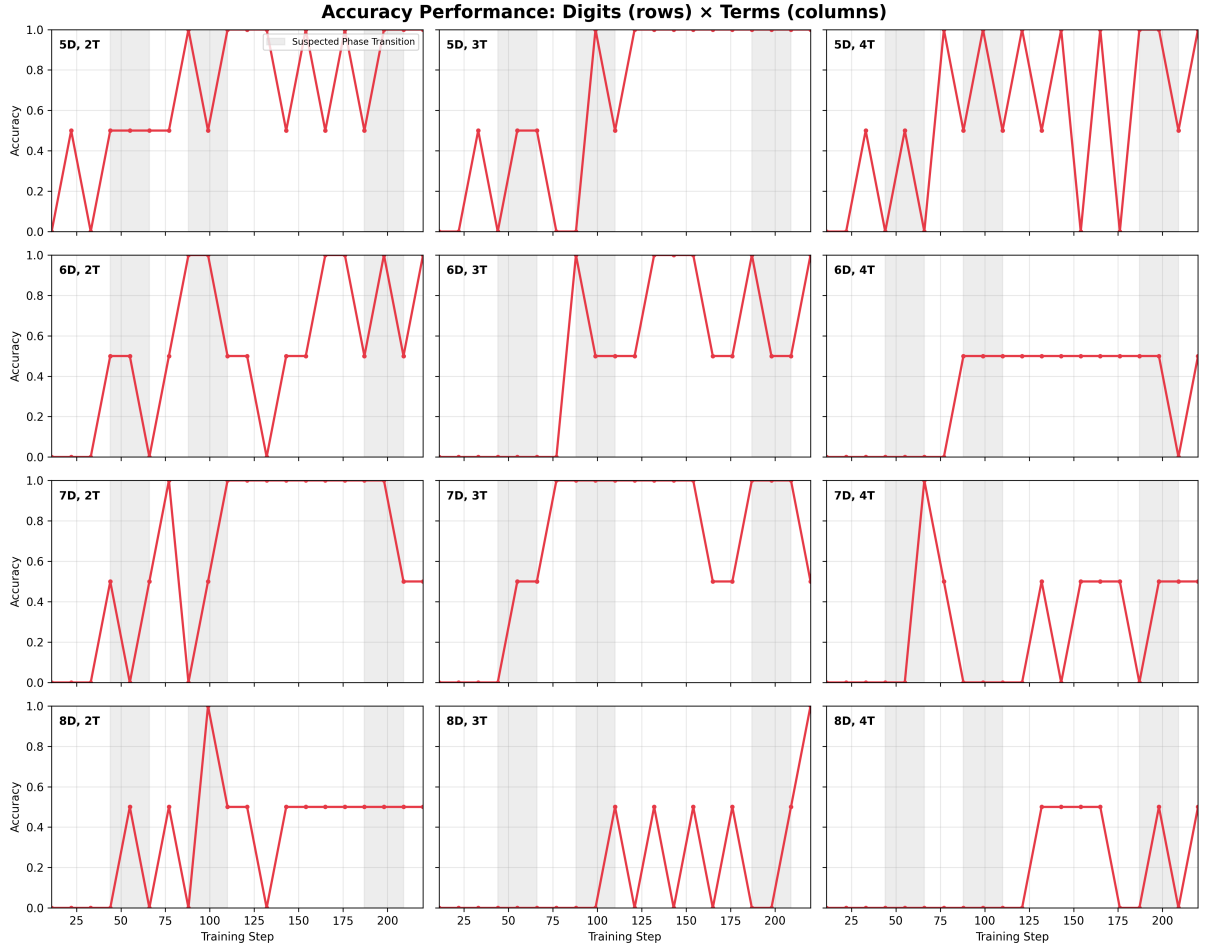


Figure 2: Accuracy performance across different arithmetic task difficulties. Each subplot shows the model’s accuracy over training for a specific combination of operand digits (rows) and expression terms (columns). The shaded regions correspond to the suspected phase transitions identified via the LLC in Figure 1.