Preprints of the 23rd IFAC World Congress
Busan, Republic of Korea, August 23-28, 2026

Abstraction Learning via Decreasing Kolmogorov Complexity

Junjie Zhang [*,**] **Zhen Shen** [**] **Gang Xiong** [**] **Fei-Yue Wang** [***]

[*] *School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China*
[**] *State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China*
[***] *State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China*

(e-mail: zhangjunjie2024@ia.ac.cn, zhen.shen@ia.ac.cn, gang.xiong@ia.ac.cn, feiyue.wang@ia.ac.cn)

Abstract: Abstract learning—the capacity to extract invariant rules from high-dimensional observations—is the cornerstone of out-of-distribution generalization. However, standard optimization paradigms often trap models in high-entropy memorization regimes due to the lack of mechanisms enforcing structural parsimony. While the existence of "Grokking "suggests a latent pathway from such memorization to generalization, the specific regularization forces required to actively drive this transition through the entropy barrier remain underexplored. In this work, we present a comprehensive thermodynamic evaluation of complexity constraints—ranging from statistical norms (L_1/L_2) and spectral entropy to algorithmic compression proxies—within a unified transformer testbed. Our empirical results demonstrate that statistical regularizers often lead to instability or brittle acceleration. In contrast, the Block Decomposition Method (BDM) (Zenil et al., 2018), which approximates Kolmogorov Complexity (KC) (Kolmogorov, 1965)via algorithmic probability, serves as a potent "control variable."BDM imposes effective entropic friction to filter out fragile memorization circuits, guiding the system toward a thermodynamically stable, rule-based local minimum. These findings underscore that robust abstraction necessitates theoretically grounded algorithmic complexity constraints rather than mere capacity control.

Keywords: Abstract learning; Algorithmic Information Theory; Thermodynamic of Learning; Kolmogorov Complexity; Grokking; Optimization

1. INTRODUCTION

The core vision of artificial intelligence lies in realizing Abstract Learning (Premack, 1978)—constructing intelligent systems capable of transcending mere statistical fitting to extract latent invariants from high-dimensional data and achieve genuine generalization. From the perspective of control theory and system identification, this process is equivalent to identifying latent, low-order nonlinear dynamic models from noisy observations (Sjöberg et al., 1995).

Historically, statistical learning theory has relied primarily on VC dimension or Rademacher complexity (Bartlett and Mendelson, 2002; Vapnik and Chervonenkis, 2015) to bound generalization, predicting that overfitting risk should increase monotonically with parameter capacity. However, the phenomenon of "Double Descent "(Belkin et al., 2019) universally observed in modern deep learning has broken this classical assumption: in the over-parameterized regime where model size far exceeds data size, test error decreases further as scale increases. This paradox has forced the community to re-examine the definition of complexity, realizing that simple parameter counting is insufficient to describe the behavior of deep models. Against this background, Algorithmic Information Theory (AIT) provides a new paradigm for understanding abstract learning: the generalization of neural networks is fundamentally a process of finding the latent generative rules behind data. Mathematically, this is equivalent to finding the shortest program capable of generating that data, i.e., minimizing KC, thereby physically realizing Occam's Razor (Kolmogorov, 1965). Therefore, the abstract learning of neural networks is essentially an algorithmic compression process, whose objective is to minimize the KC of the input–output mapping under prescribed error constraints.

Given the fundamental incomputability of KC (Vitányi, 2020), the quantification of this "shortest program "has undergone a theoretical evolution from shallow proxies to

⋆ This work was supported in part by the National Key Research and Development Program of China under Grant 2025YFE0112300, by the National Natural Science Foundation of China under Grants 62461160259 and 92360307, and by the Beijing Natural Science Foundation under Grants L233005 L243009 and L242093.
Zhen Shen is the corresponding author.

©2026 the authors. Accepted by IFAC for publication under a Creative Commons Licence CC-BY-NC-ND

deep geometric interpretations. Early research attempted to use generic compression algorithms like gzip or LZMA as proxies for KC (Cilibrasi and Vitányi, 2005), but these were gradually abandoned due to their inability to capture deep semantic abstractions, as they rely primarily on detecting superficial syntactic redundancies in the serialized byte stream. Subsequently, the proposal of Singular Learning Theory (SLT) and the Local Learning Coefficient (LLC) (Watanabe, 2009) provided a geometric interpretation for complexity, proving that the "flatness"of the loss landscape represents "compressibility."This coincides with robust stability theory in the control domain (Balakrishnan and Kashyap, 1999): a solution located in a wide basin implies insensitivity of its parameters to perturbations, meaning the system possesses higher structural stability. Recent research (Shaw et al., 2025) has further proven via variational inference (VI) that neural networks possess the capability to adaptively learn minimal encodings without external compressors, effectively treating the optimization process as a mechanism for intrinsic algorithmic compression consistent with the Minimum Description Length (MDL) (Rissanen, 1978) principle.

However, identifying a "low complexity"state as a potentially favorable target does not guarantee reachability; models often remain in a "high complexity"regime for extended periods, which constitutes the core empirical observation of Grokking (Power et al., 2022). From a thermodynamic-inspired perspective, this behavior can be interpreted as the optimization landscape containing regions of high algorithmic randomness, which we refer to conceptually as an "Entropy Barrier". By treating KC as a heuristic "Algorithmic Complexity Proxy", Grokking can be viewed not as an anomaly, but as a phase in which the model transitions from a memorization-dominated regime (high Shannon entropy) to a more rule-based, low-KC regime (interpreted as understanding). This analogy mirrors certain behaviors observed in nonlinear systems with multi-stable landscapes (Drazin, 1992), where the optimizer may escape local basins, explore plateaus through stochastic updates, and eventually converge to solutions characterized by simpler algorithmic structure.

Given the current lack of comprehensive empirical evaluation on how algorithmic complexity regularization specifically affects phase transition dynamics, we establish a standardized testbed based on Algorithmic Modular Arithmetic. This task is often referred to as the "Drosophila" of abstract learning research, due to the large complexity gap between the memory-based high-entropy solution (lookup table) and the rule-based low-entropy solution (group axioms). Such a task provides a convenient experimental setting to investigate whether and how models transition from memorization-dominated behavior to rule-based, low-complexity solutions, allowing us to examine the effects of algorithmic complexity regularization on this phase transition in a controlled manner. On this platform, we trained standard Transformer models and comprehensively evaluated the impact of existing regularization methods on the Grokking phenomenon. The evaluation covers:

- Statistical norm constraints ($L_0/L_1/L_2$ regularization);
- Geometric spectral constraints (spectral entropy);
- Stochastic noise injection (Gaussian noise);
- Loss landscape geometric constraints (second-order approximation based on SLT);
- Compressor proxies (gzip/LZMA);
- Variational inference (VI);
- Block decomposition method (BDM).

Through comparative experiments, we explore the internal mechanisms by which different regularization methods act as "control signals"to drive the phase transition.

2. LITERATURE REVIEW

2.1 Intelligence Emerges from Abstraction

The historical roots of abstract learning profoundly reveal the essence of intelligence: it is not merely fitting data, but efficiently reconstructing physical reality (Premack, 1978). From a cognitive science perspective, to survive within limited constraints of brain capacity and energy budget, biological intelligence must strip away redundant information from high-dimensional sensory streams, extracting compact relational invariants (e.g., symmetry, causality) (Barlow et al., 1961). This paradigm of "Understanding by Compression"is considered the core driver for the emergence of intelligence. In the realm of machine learning, while early Statistical Learning Theory relied on capacity metrics like VC dimensionVapnik and Chervonenkis (2015), recent research has gradually returned to this physical essence. The "Consciousness Prior"proposed by (Bengio, 2017) advocates that agents should learn sparse factor graphs, essentially seeking the minimal description of the world. Subsequent neuro-symbolic research (Deletang et al., 2023; Wang et al., 2024) and recent debates on Large Language Models (LLMs) have further reinforced this view: genuine logical reasoning capability stems not from memorizing superficial statistical correlations, but from the formation of modular, low-entropy algorithmic circuits within the network. Therefore, abstraction is no longer an ethereal cognitive concept, but a concrete physical process where neural networks eliminate entropic noise through compression to approximate the generative laws behind the data.

2.2 KC in Deep Learning: Minimum Description Length

Since abstraction can be viewed as compression, quantifying the "degree of compression"becomes central. KC provides the theoretical ideal, while practical neural-network formulations have largely followed the MDL principle. Rissanen (1978) established MDL by favoring models that maximally compress data, and Hinton and Van Camp (1993) introduced this idea into neural networks through the Bits-Back argument, showing that variational weight compression can improve generalization. This direction has recently been extended by Shaw et al. (2025), who proposed asymptotically optimal variational objectives that allow transformers to learn compact weight encodings without external compressors. Beyond variational methods, generic lossless compressors have also been used as KC proxies. For example, Jiang et al. (2023) adopted gzip length as a classification metric, but such methods mainly capture shallow byte-level redundancy rather than deep algorithmic structure. Algorithmic Probability, introduced by Levin (1974), offers a more principled alternative. Based

Table 1. Grokking epochs across four modular arithmetic tasks. × denotes that grokking was not observed in 3000 epochs across three seeds. Bold indicates the earliest grokking epoch for each task.

Regularizer	λ	Addition	Subtraction	Multiplication	Division
Gzip	10^{-4}	×	×	×	×
	10^{-3}	×	×	×	×
L_0	10^{-4}	×	×	×	×
	10^{-3}	×	×	×	×
L_1	10^{-4}	2533 ± 417	×	2783 ± 209	×
	10^{-3}	×	×	×	×
L_2	10^{-4}	1917 ± 85	×	2000 ± 41	×
	10^{-3}	867 ± 62	1417 ± 62	833 ± 125	1350 ± 82
LZMA	10^{-4}	×	×	×	×
	10^{-3}	×	×	×	×
Noise	10^{-4}	×	×	×	×
	10^{-3}	×	×	×	×
SLT	10^{-4}	×	×	×	×
	10^{-3}	×	×	×	×
Spectral	10^{-4}	×	×	×	×
	10^{-3}	×	×	×	×
VI	10^{-4}	2700 ± 108	×	2833 ± 62	×
	10^{-3}	983 ± 24	1617 ± 143	950 ± 41	1717 ± 62
Ours	10^{-4}	**567 ± 62**	**1067 ± 85**	**600 ± 41**	**1050 ± 294**
	10^{-3}	×	×	×	×

on this idea, Zenil et al. (2018) developed CTM and BDM to estimate algorithmic complexity from the output probabilities of small Turing machines. Unlike Shannon-style compressors, BDM can assign low complexity to statistically irregular but rule-generated patterns, shifting regularization from magnitude or sparsity control toward penalizing algorithmic generation cost. Thus, these approaches share a common goal: finding a mathematical force that drives models to extract low-complexity structure from high-complexity observations.

2.3 Grokking: Delayed Generalization in Modular Arithmetic Tasks

If abstraction is the goal and KC is the metric, then the Grokking (Power et al., 2022) phenomenon is the macroscopic projection of this compression process onto training dynamics. Grokking describes the delayed phenomenon where validation accuracy jumps long after the model has memorized all training samples. This counter-intuitive discovery shatters the traditional wisdom that "overfitting is the end,"implying the existence of two distinct solutions within neural networks: a high-entropy memory solution and a low-entropy rule solution. Thermodynamic-inspired interpretations (Liu et al., 2022) view Grokking as a transition from memorization-dominated solutions to rule-based generalization. In this view, models may first fit the training data through lookup-table-like mechanisms with high algorithmic complexity, and later move toward simpler structured solutions under sustained regularization. However, while existing literature offers qualitative explanations for Grokking, there remains a lack of a comprehensive empirical framework to evaluate the extent to which the aforementioned KC approximation methods can serve as effective control variables to actively modulate this phase transition and accelerate the emergence of intelligence.

3. APPROACH

To comprehensively investigate the impact of different complexity constraints on the phase transition of generalization (Grokking), we implemented multiple approximations within a unified optimization framework. The total objective function is defined as $\mathcal{L}_{\text{total}}(\theta) = \mathcal{L}_{\text{task}}(\theta) + \lambda \mathcal{R}(\theta)$, where $\mathcal{R}(\theta)$ represents the specific complexity regularization term derived from the categories discussed above.

3.1 Statistical Norm Constraints

These methods control complexity by constraining parameter magnitude or sparsity, ignoring structural correlations.

L_2 Regularization (Weight Decay) limits the parameter energy, corresponding to a Gaussian prior over weights.

$$\mathcal{R}_{L2}(\theta) = \sum_{l} \|W^{(l)}\|_F^2 = \sum_{l,i,j} (W_{i,j}^{(l)})^2 \tag{1}$$

L_1 Regularization (Lasso) acts as a convex proxy to induce sparsity, corresponding to a Laplacian prior.

$$\mathcal{R}_{L1}(\theta) = \sum_{l} \|W^{(l)}\|_1 = \sum_{l,i,j} |W_{i,j}^{(l)}| \tag{2}$$

Smoothed L_0 Approximation To address the non-differentiability of the discrete L_0 count, we adopt a non-linear smoothed proxy that simulates soft circuit pruning.

$$\mathcal{R}_{L0}(\theta) \approx \sum_{j} \left(1 - e^{-|\theta_j|/\tau}\right) \tag{3}$$

Table 2. Runtime comparison between L_2 regularization and our Fast-BDM regularization. Grokking epoch and total runtime are reported as mean ± standard deviation over three seeds. Speedup is computed as the ratio of the L_2 grokking epoch to the Fast-BDM grokking epoch.

Task	Method	λ	Grokking Epoch	Total Time (s)	Speedup
Addition	L_2	10^{-3}	867 ± 62	8.9 ± 0.8	–
Addition	**Ours**	10^{-4}	$\mathbf{567 \pm 62}$	7.0 ± 0.7	**1.53**×
Subtraction	L_2	10^{-3}	1417 ± 62	13.7 ± 0.5	–
Subtraction	**Ours**	10^{-4}	$\mathbf{1067 \pm 85}$	12.8 ± 1.0	**1.33**×
Multiplication	L_2	10^{-3}	833 ± 125	8.3 ± 1.2	–
Multiplication	**Ours**	10^{-4}	$\mathbf{600 \pm 41}$	7.7 ± 0.4	**1.39**×
Division	L_2	10^{-3}	1350 ± 82	12.9 ± 0.9	–
Division	**Ours**	10^{-4}	$\mathbf{1050 \pm 294}$	12.8 ± 3.2	**1.29**×
Average	L_2	10^{-3}	1117 ± 286	11.0 ± 2.6	–
Average	**Ours**	10^{-4}	$\mathbf{821 \pm 275}$	10.1 ± 3.2	**1.38**×

where τ is a temperature parameter controlling the smoothness of the approximation. When $|\theta_j| \ll \tau$, the penalty vanishes; when $|\theta_j| \gg \tau$, it approaches 1. This term applies strong shrinking gradients to minute non-zero weights, effectively approximating a discrete sparse topology during training.

Stochastic Noise Injection: Instead of an explicit additive penalty, this implements complexity control via multiplicative or additive noise injection into the weights during the forward pass. For a noise level σ, the perturbed weights are $\tilde{\theta} = \theta + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$.

$$\mathcal{L}_{\text{noise}}(\theta) = \mathbb{E}_\epsilon[\mathcal{L}_{\text{task}}(\theta + \epsilon)] \tag{4}$$

This operation implicitly regularizes the trace of the Fisher Information Matrix (approximating the Hessian) (Bishop, 1995), thereby limiting the information capacity of the weights and acting as an Information Bottleneck constraint.

3.2 Geometric and Spectral Constraints

These methods focus on the effective rank of the weight matrix or the curvature features of the loss landscape.

Spectral Entropy Regularization aims to minimize the effective rank of weights. Treating the normalized singular value spectrum $p_k = \sigma_k / \sum \sigma_j$ as a probability distribution, we minimize its Spectral entropy:

$$\mathcal{R}_{\text{SE}}(\theta) = \sum_l \left(- \sum_k p_k^{(l)} \log p_k^{(l)} \right) \tag{5}$$

This encourages the weight matrix to become degenerate (low-rank), reducing the effective degrees of freedom in the network.

SLT-inspired Hessian Trace (Geometric Flatness): We adopt an SLT-inspired regularizer because over-parameterized neural networks are often singular: many parameter settings may represent the same function, making classical parameter-count-based complexity measures less informative. SLT motivates analyzing generalization through the local geometry of the loss landscape. Here, we use the Hessian trace as a tractable proxy for average local curvature, rather than as a direct SLT quantity.

Let

$$H(\theta) = \nabla_\theta^2 \mathcal{L}(\theta) \tag{6}$$

be the Hessian of the training loss. A smaller $\text{Tr}(H)$ indicates lower average local curvature and is commonly used as a proxy for flatter geometry:

$$\mathcal{R}_{\text{geom}}(\theta) = \text{Tr}(H(\theta)). \tag{7}$$

Since explicitly forming the full $N \times N$ Hessian is infeasible for large models, we approximate its trace with the Hutchinson estimator. Let $v \in \mathbb{R}^N$ be a random vector with zero mean and unit covariance, e.g., Rademacher or standard Gaussian, such that $\mathbb{E}[vv^\top] = I$. Then

$$\text{Tr}(H) = \mathbb{E}_v \left[v^\top H v \right]. \tag{8}$$

The term $v^\top H v$ can be computed through a Hessian-vector product without materializing H:

$$H(\theta)v = \nabla_\theta \left(\nabla_\theta \mathcal{L}(\theta)^\top v \right). \tag{9}$$

Thus, our stochastic geometric regularizer is

$$\mathcal{R}_{\text{geom}}(\theta) \approx v^\top \nabla_\theta \left(\nabla_\theta \mathcal{L}(\theta)^\top v \right). \tag{10}$$

In implementation, we compute $g = \nabla_\theta \mathcal{L}(\theta)$ with the computation graph retained, form $g^\top v$, and apply automatic differentiation to obtain the Hessian-vector product. This avoids storing the full Hessian and makes curvature-based regularization feasible for deep models.

3.3 Algorithmic Complexity Constraints

These methods aim to explicitly approximate the KC or MDL of the parameters.

VI (MDL) adopts the "Bayes-by-Backprop"approach where each weight is modeled as a normal distribution $\theta_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$. The complexity cost is the Kullback-Leibler divergence between the learned posterior and a fixed standard normal prior $p(\theta)$:

$$\mathcal{R}_{\text{VI}}(\theta) = \sum_i D_{KL}\big(\mathcal{N}(\mu_i, \sigma_i^2) \| \mathcal{N}(0, 1)\big) \tag{11}$$

This term explicitly quantifies the description length of the weights in bits, penalizing parameters that deviate significantly from the prior or require high precision (low variance).

Compressor Proxy uses generic lossless compression algorithms (e.g., gzip) as an upper-bound approximation for KC. For a quantized parameter stream $s(\theta)$, the regularization term is:

$$\mathcal{R}_{\text{zip}}(\theta) = \text{Length}\big(\texttt{Compress}(s(\theta))\big) \tag{12}$$

Table 3. Effect of CTM block size in Fast-BDM. Patterns: total CTM binary patterns; Unique: distinct CTM values; Range: CTM dynamic range; Reg. (ms): per-step regularization time; Overhead: regularization cost ratio. Results are averaged over four modular arithmetic tasks and three random seeds.

Block	Patterns	Unique	Range	Grok. Epoch	Time (s)	Reg. (ms)	Overhead
2×2	16	4	0.0949	1308 ± 357	15.2 ± 3.8	1.303 ± 0.010	$24.1\% \pm 0.4\%$
3×3	512	51	5.2675	1112 ± 357	13.8 ± 4.2	1.344 ± 0.006	$23.8\% \pm 0.5\%$
4×4	**65536**	**3365**	**14.0161**	**821 ± 275**	**10.1 ± 3.2**	1.339 ± 0.013	$24.5\% \pm 0.5\%$

Since standard compressors are non-differentiable, optimization is typically performed via proxy gradients.

Fast BDM (ours): To obtain an efficient proxy for BDM-based algorithmic complexity regularization, we split each weight matrix W into non-overlapping $B \times B$ blocks $\{W_k\}$. For each block, we keep only the sign pattern of its weights: positive entries are mapped to 1 and non-positive entries to 0,

$$\hat{b}_k = \mathbf{1}(W_k > 0). \tag{13}$$

Thus, $\hat{b}_k$ is a binary pattern describing the sign structure of block W_k. We then flatten it in a fixed order and convert it into an integer index i_k for querying the precomputed CTM dictionary $\mathcal{T}$, which stores one complexity value for each possible $B \times B$ binary pattern.

Since the sign operation and CTM lookup are discrete and non-differentiable, we do not directly optimize the CTM value. Instead, the lookup result is used as a detached block-wise coefficient that reweights the Frobenius norm of the original continuous block:

$$\mathcal{R}_{\text{Fast-BDM}}(W) = \sum_k \text{sg}(\mathcal{T}[i_k]) \, \|W_k\|_F^2, \tag{14}$$

where $\text{sg}(\cdot)$ denotes stop-gradient.

Thus, for an element $w_{ij} \in W_k$, the gradient is

$$\frac{\partial \mathcal{R}_{\text{Fast-BDM}}}{\partial w_{ij}} = 2 \, \mathcal{T}[i_k] \, w_{ij}. \tag{15}$$

In implementation, we extract blocks by reshaping or unfolding W, compute binary codes from their signs, query $\mathcal{T}$ once per block, detach the resulting coefficients, and apply a weighted L_2 penalty to the original weights. This makes the method a CTM-weighted weight decay: blocks with more complex sign patterns receive stronger shrinkage, while simpler patterns are penalized less. Compared with Bernoulli relaxation or Monte Carlo estimation, this deterministic variant avoids repeated sampling and conditional CTM evaluations, and has approximately $O(|W|)$ cost with vectorized block extraction and lookup.

4. EXPERIMENTS

4.1 Experimental Settings

The experiment employs the Modular Addition ($x + y \pmod{113}$) task. The dataset consists of 113×113 sample pairs. We randomly split the data into a 30% training set and a 70% test set. This high ratio of held-out data aims to simulate the sparse observation environment required for the Grokking phenomenon, forcing the model to choose between memorization and generalization.

We utilize a lightweight transformer model containing 2 Encoder layers, 4 Attention Heads, a hidden layer dimension (d_{model}) of 128, and a feed-forward network dimension of 512, utilizing learnable positional encodings. This architecture is compact enough to facilitate high-frequency experimental iteration and analysis. All models are trained using the AdamW optimizer with a learning rate set to 10^{-3} . The Batch size is set to 2048. Each experiment runs for 3000 Epochs to ensure the model has ample time to evolve from the early memorization overfitting stage to the generalization stage.

To verify the robustness of the results, we execute the following evaluation process:

- Parameter Search: For each regularization method, a grid search is performed within the range $\lambda \in \{10^{-4}, 10^{-3}\}$, as preliminary experiments indicate that using $\lambda \geq 10^{-2}$ leads to training instability and causes the model to fail to learn on the training set.
- Randomness Control: Each configuration is repeated under 3 independent random seeds (42, 43, 44).
- Metric Reporting: The full trajectories of training and testing accuracy are recorded, and error bands are plotted in the form of "mean ± standard deviation"to quantify the stability of the phase transition.

4.2 Results and Analysis

Table 1 reveals a clear gap between statistical regularization and algorithmic structural regularization. Most alternative methods fail to consistently induce grokking across the four modular arithmetic tasks. This is because their regularization signals are only indirectly related to the underlying rule structure: L_0 and L_1 encourage sparsity, but sparse weights may still implement memorization; gzip and LZMA capture surface-level byte redundancy rather than algorithmic organization; and SLT or spectral entropy constrain geometry or rank without ensuring that the learned representation encodes the modular operation.

Among conventional baselines, L_2 is the most effective, suggesting that limiting parameter energy can suppress some unstable memorization paths. However, L_2 penalizes all weights uniformly and cannot distinguish a simple rule-like pattern from a complex memorization pattern with similar magnitude. Therefore, as shown in Table 2, it can eventually trigger grokking but reaches the transition later than Fast-BDM.

Fast-BDM is more effective because CTM provides a block-wise estimate of algorithmic complexity. High-complexity binary weight patterns receive stronger penalties, whereas simpler, more structured blocks are penalized less. This creates a targeted pressure against memorization-oriented configurations and biases optimization toward rule-consistent parameter topologies. Although this introduces additional per-step regularization cost, the stronger structural signal shortens the search process toward the rule-based basin, so the total time to grokking remains comparable to or lower than that of L_2.

Table 3 further explains the role of block size. Small CTM blocks provide only coarse complexity resolution, mapping many different structures to a limited set of scores. Larger blocks yield a richer complexity spectrum and better separate irregular memorization-like patterns from simple algorithmic ones. The 4×4 setting therefore offers the best trade-off between structural discrimination and computational cost, and is used as the default Fast-BDM configuration.

5. CONCLUSION

We investigate how complexity regularization affects Grokking in transformer models on modular arithmetic tasks in 3000 epochs. Although L_2 is the strongest conventional baseline, its transition remains consistently later than ours. By deterministically binarizing weight blocks and using CTM lookup to penalize high-complexity patterns, our method based on BDM directly regularizes structural algorithmic complexity and achieves the earliest grokking across all tasks. Runtime and block-size analyses further show that this acceleration does not increase the total time to grokking, and that 4×4 CTM blocks provide the best trade-off between complexity resolution and computational efficiency. These results suggest that robust abstraction learning requires regularizing algorithmic structure rather than only parameter magnitude, sparsity, or generic compression statistics.

REFERENCES

Balakrishnan, V. and Kashyap, R. (1999). Robust stability and performance analysis of uncertain systems using linear matrix inequalities. *Journal of Optimization Theory and Applications*, 100(3), 457–478.

Barlow, H.B. et al. (1961). Possible principles underlying the transformation of sensory messages. *Sensory communication*, 1(01), 217–233.

Bartlett, P.L. and Mendelson, S. (2002). Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of machine learning research*, 3(Nov), 463–482.

Belkin, M., Hsu, D., Ma, S., and Mandal, S. (2019). Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32), 15849–15854.

Bengio, Y. (2017). The consciousness prior. *arXiv preprint arXiv:1709.08568*.

Bishop, C.M. (1995). Training with noise is equivalent to tikhonov regularization. *Neural computation*, 7(1), 108–116.

Cilibrasi, R. and Vitányi, P.M. (2005). Clustering by compression. *IEEE Transactions on Information theory*, 51(4), 1523–1545.

Deletang, A., Catt, E., Lopez, A., Skithakis, N., Vedantam, R., Saxe, A.M., Zaremba, W., Pascanu, R., and Badia, A.P. (2023). Language modeling is compression. In *The Eleventh International Conference on Learning Representations*. URL `https://openreview.net/forum?id=iz9ruOsilU`. ICLR 2023.

Drazin, P.G. (1992). *Nonlinear systems*. 10. Cambridge University Press.

Hinton, G.E. and Van Camp, D. (1993). Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the sixth annual conference on Computational learning theory*, 5–13.

Jiang, Z., Yang, M., Tsirlin, M., Tang, R., Dai, Y., and Lin, J. (2023). "Low-resource" text classification: A parameter-free classification method with compressors. In *Findings of the Association for Computational Linguistics: ACL 2023*, 6810–6828.

Kolmogorov, A.N. (1965). Three approaches to the definition of the concept "quantity of information". *Problemy peredachi informatsii*, 1(1), 3–11.

Levin, L.A. (1974). Laws of information conservation (nongrowth) and aspects of the foundation of probability theory. *Problemy Peredachi Informatsii*, 10(3), 30–35.

Liu, Z., Michaud, E.J., and Tegmark, M. (2022). Omnigrok: Grokking beyond algorithmic data. *arXiv preprint arXiv:2210.01117*.

Power, A., Burda, Y., Edwards, H., Babuschkin, I., and Misra, V. (2022). Grokking: Generalization beyond overfitting on small algorithmic datasets. URL `https://arxiv.org/abs/2201.02177`.

Premack, D. (1978). On the abstractness of human concepts: Why it would be difficult to talk to a pigeon. In S.H. Hulse, H. Fowler, and W.K. Honig (eds.), *Cognitive processes in animal behavior*, 423–451. Erlbaum, Hillsdale, NJ.

Rissanen, J. (1978). Modeling by shortest data description. *Automatica*, 14(5), 465–471.

Shaw, P., Cohan, J., Eisenstein, J., and Toutanova, K. (2025). Bridging kolmogorov complexity and deep learning: Asymptotically optimal description length objectives for transformers. *arXiv preprint arXiv:2509.22445*.

Sjöberg, J., Zhang, Q., Ljung, L., Benveniste, A., Delyon, B., Glorennec, P.Y., Hjalmarsson, H., and Juditsky, A. (1995). Nonlinear black-box modeling in system identification: a unified overview. *Automatica*, 31(12), 1691–1724.

Vapnik, V.N. and Chervonenkis, A.Y. (2015). On the uniform convergence of relative frequencies of events to their probabilities. In *Measures of complexity: festschrift for alexey chervonenkis*, 11–30. Springer.

Vitányi, P.M. (2020). How incomputable is kolmogorov complexity? *Entropy*, 22(4), 408.

Wang, B., Yue, X., Su, Y., and Sun, H. (2024). Grokking of implicit reasoning in transformers: A mechanistic journey to the edge of generalization. *Advances in Neural Information Processing Systems*, 37, 95238–95265.

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

Zenil, H., Hernández-Orozco, S., Kiani, N.A., Soler-Toscano, F., Rueda-Toicen, A., and Tegnér, J. (2018). A decomposition method for global evaluation of shannon entropy and local estimations of algorithmic complexity. *Entropy*, 20(8), 605.