CVPR JUNE 3-7, 2026 DENVER COLORADO

ReLaX: Reasoning with Latent Exploration for Large Reasoning Models

Shimin Zhang, Xianwei Chen*, Yufan Shen*, Ziyuan Ye, Jibin Wu*[†]

shimin25.zhang@connect.polyu.hk

paper

github

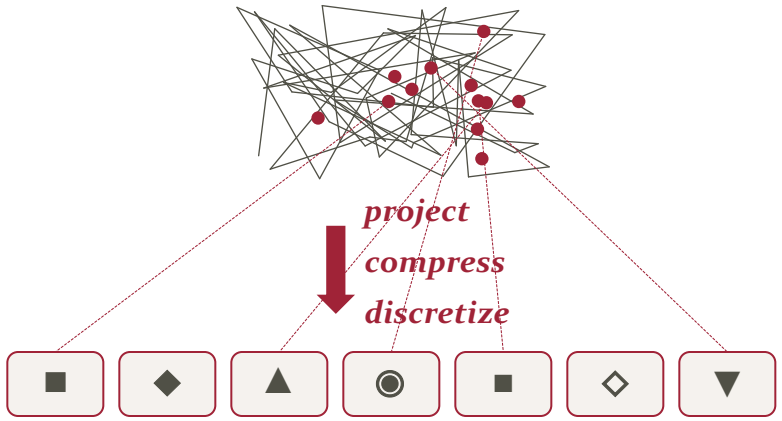
jiqizhixin

Common View from Token Space

From behavioral probes to mechanistic interpretability, ***generated tokens of the model*** *are where we usually look.*

Latent representation: hidden states

continuous · high-dimensional



Tokens — the shadow

discrete · low-dimensional

Token-centric research studies only ***what survives this projection.***

WHY THE FIELD LOOKS HERE?

Tokens are observable, measurable, benchmarkable

- observable
- measurable
- benchmarkable
- scalable

WHY IT FALLS SHORT?

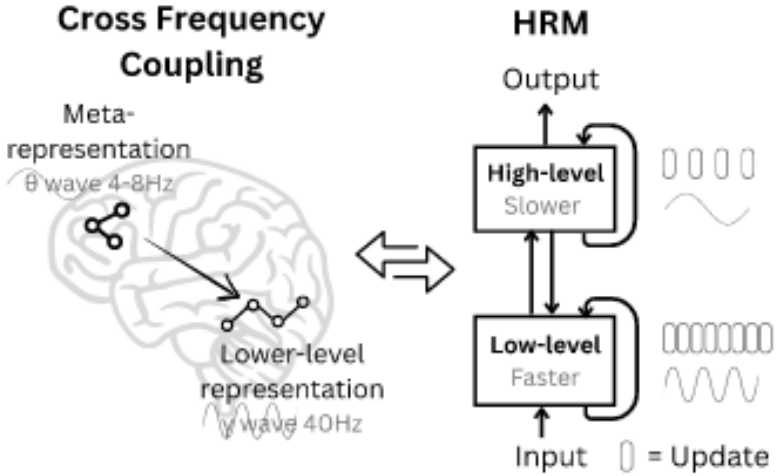
Tokens hide the machinery that produced them

- shallow signal
- hides machinery
- no dynamics
- hard to diagnose

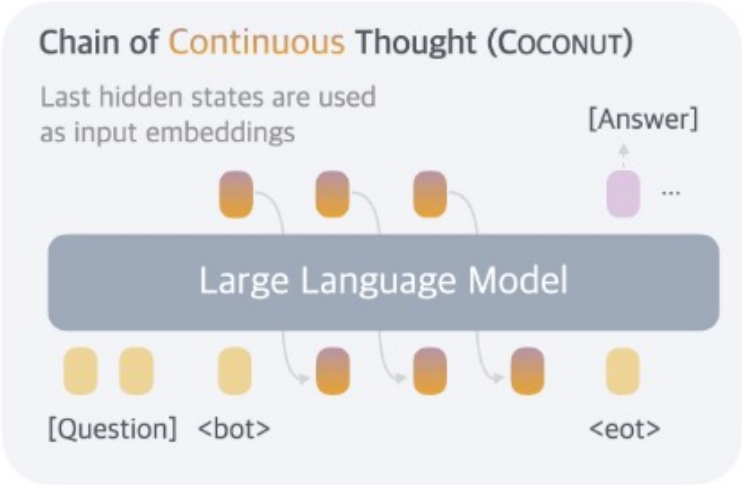
Tokens are the shadow of the computation — not the computation itself.

Scaling Latent Computation ≠ Understanding It

*With growing research interest in latent space, current work **expands latent computation for performance**, but leaves its underlying principles largely unexplained.*



HRM, 2025, from Sapient Intelligence



Coconut, 2025, from Meta

...

Scale over insight: Extend reasoning chains and trajectory depth to squeeze out better answers — without asking what those trajectories mean.

No principled framework: We lack a theory for what latent dynamics represent and how computation unfolds step by step inside the model.

These methods improve what the model can do — not our understanding of how it works.

Representation Analysis

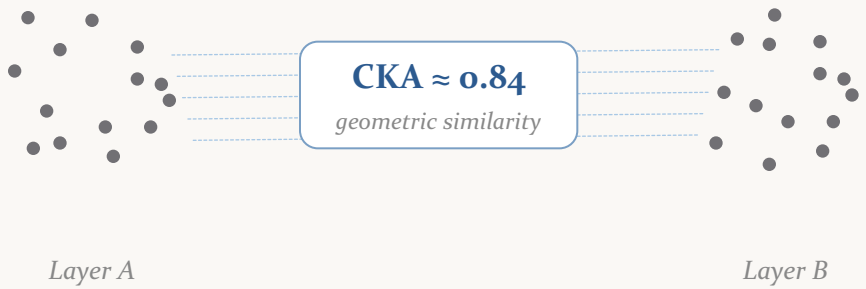
*Representation analysis opens the black box, but **what you measure determines what you see**.*

How classical tools measure

CCA · CKA · Procrustes

Compare two static states.

Ask: are their representational geometries aligned?

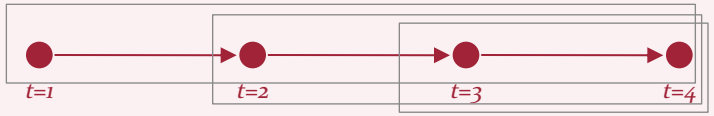


Why it falls short on LLMs (e.g., reasoning)

Computation is reflected by a series of states

1 *Temporal dependency*

LLM states at each step depend on all prior steps.



2 *Curse of scaling*

CNN features: thousands of dims. LLM states: tens of thousands, millions of tokens.

What we need:

- ***Study the dynamics of representation — not just its geometry at a point in time.***
- ***Capture structure that scales with the representation — tools built for high-dimensional, long-sequence regimes.***

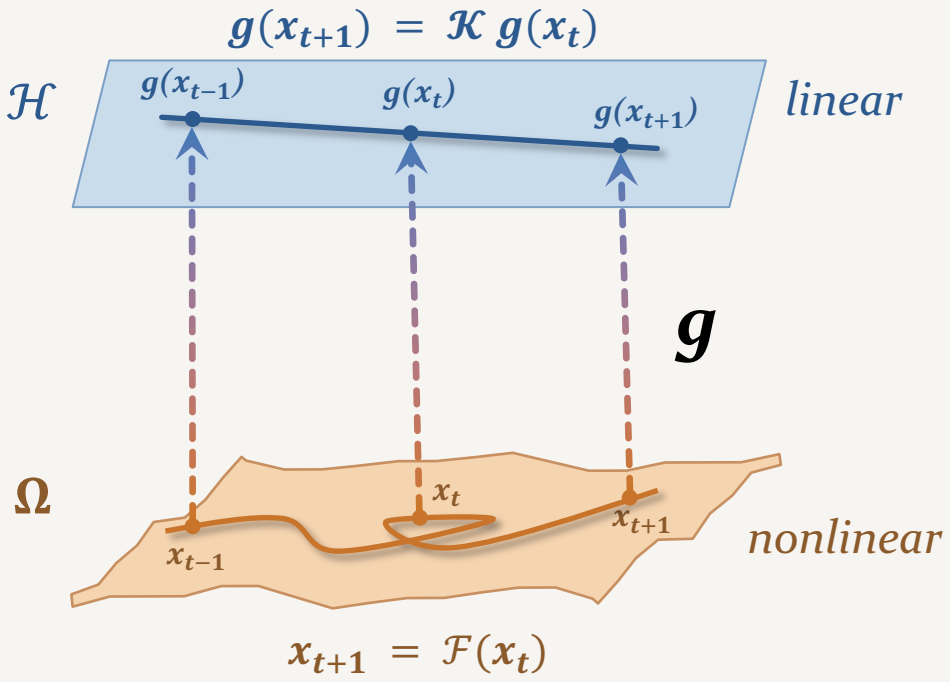
Theoretical Foundation: Koopman Operator Theory

Linearizing the nonlinearity: *instead of chasing the nonlinear state* $\boldsymbol{x}$*, track an infinite family of observables* $\boldsymbol{g}(\boldsymbol{x})$ *— and their evolution is* ***exactly linear****.*

Two key concepts:

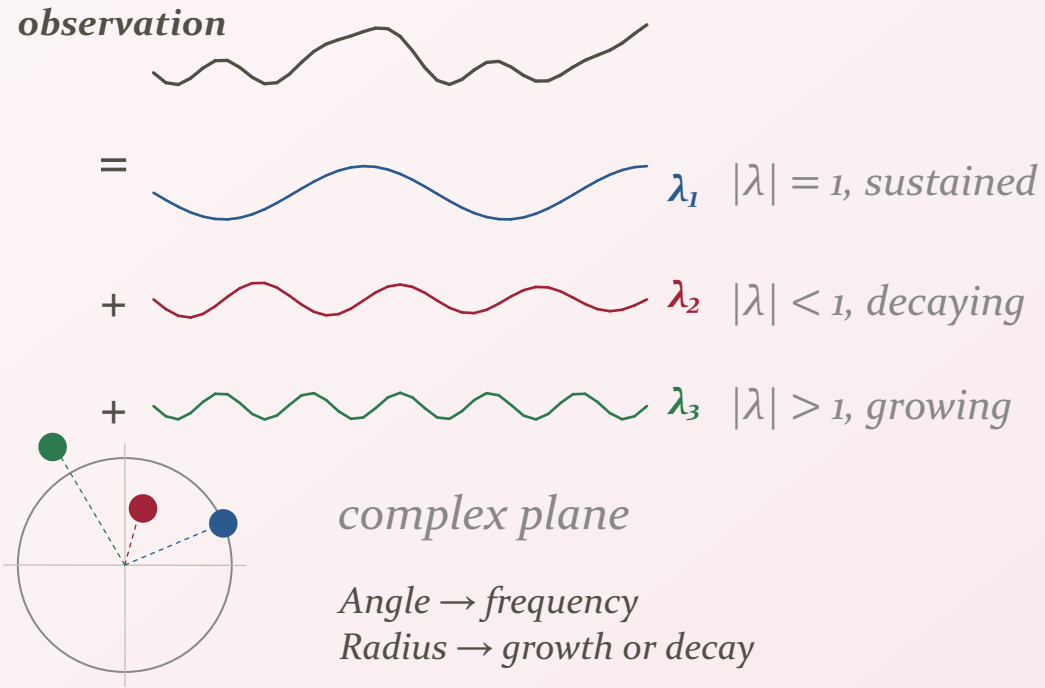
Koopman Dictionary

nonlinear dynamics → linear observation



Koopman Spectrum

A set of dynamical modes, each oscillates with its own λ



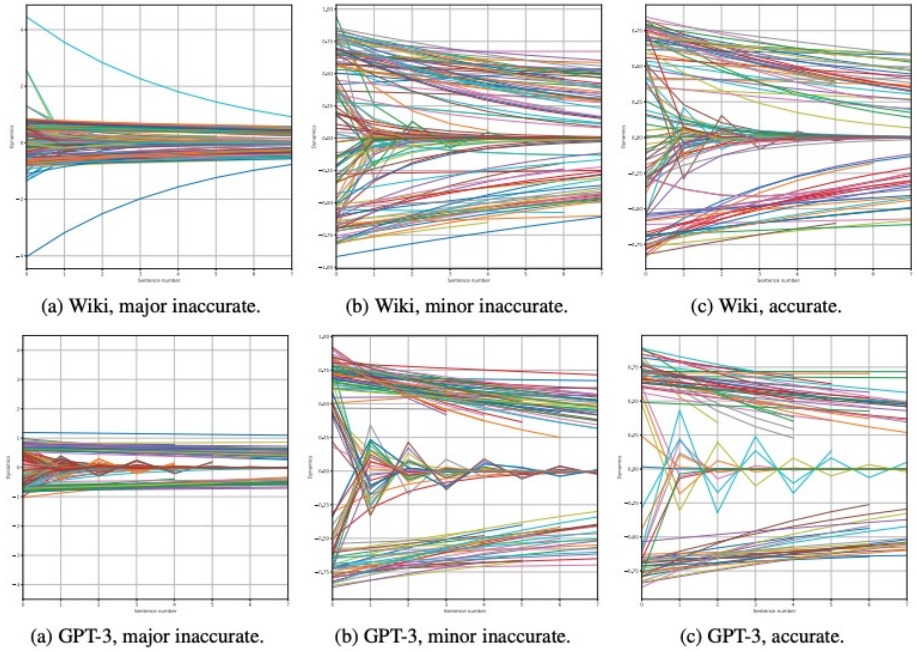
Koopman for LLM Representation

The use of Koopman operator theory in LLM research remains largely underexplored and not yet well established …

Hallucination Detecting 2023

How the spectrum of sentence dynamics can uncover LLM hallucinations?

Dynamical Modes Difference
GT (top) vs GPT3 (bottom)



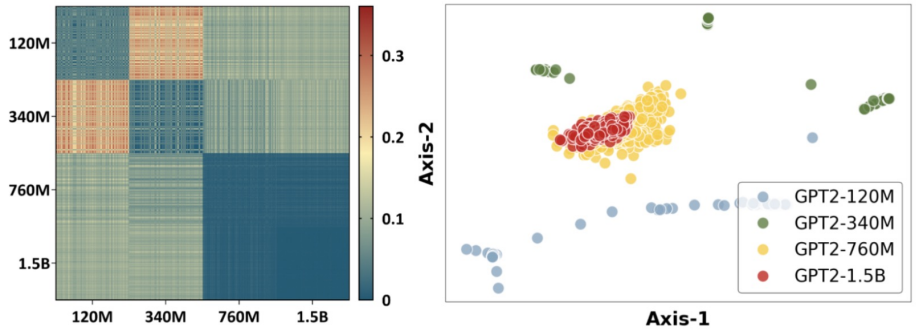
(a) Wiki, major inaccurate. (b) Wiki, minor inaccurate. (c) Wiki, accurate.

(a) GPT-3, major inaccurate. (b) GPT-3, minor inaccurate. (c) GPT-3, accurate.

KoopSTD *ICML 2025*

Reliable similarity analysis between dynamical systems.

GPT-2 Last-layer Hidden States



Clustered
Coherent dynamics

Scattered
Inconsistent dynamics

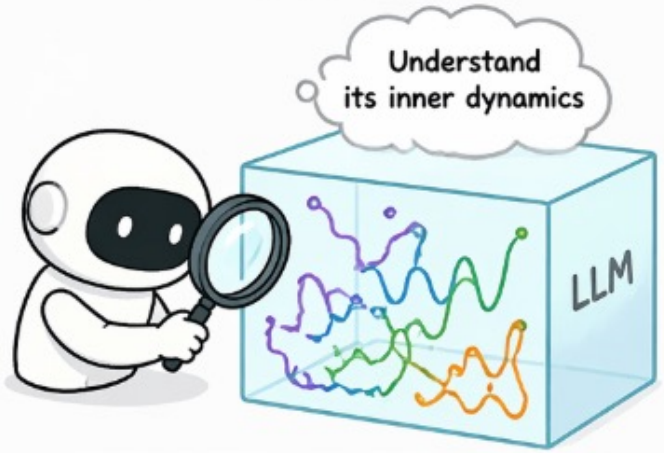
A dynamical scaling law: larger models exhibit more stable and coherent internal computations across diverse inputs, whereas smaller models display more variable and inconsistent dynamics.

Let's Think…

Beyond analysis, how can we seek the opportunity of improving the performance of an LLM?

1. ANALYZE & UNDERSTAND

Look inside the model



Patterns | Mechanisms | Dynamics

Get insights from the model's internal computation

2. IMPROVE & LEVERAGE

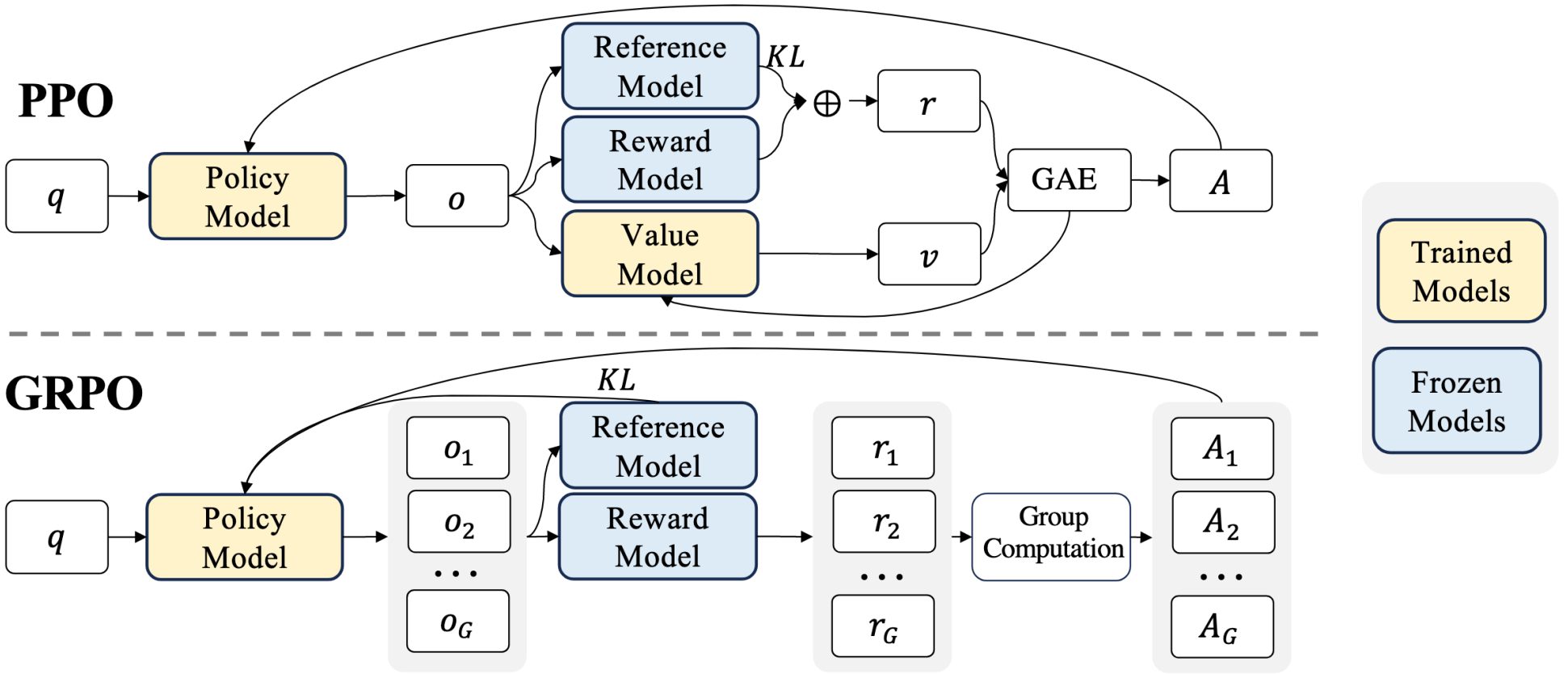
Turn insights into better performance

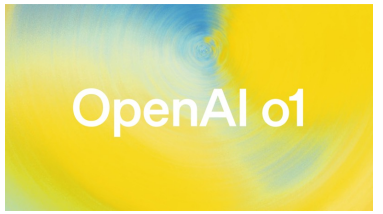
Better reasoning | More robust | Higher performance

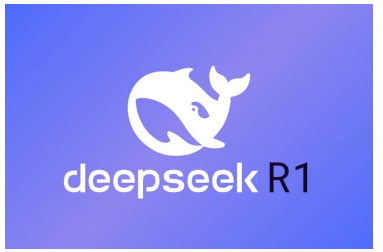
From understanding to improvement: turn insights into practical gains.

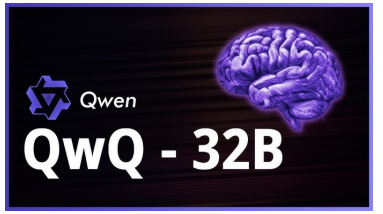
Reinforcement Learning with Verifiable Rewards (RLVR)

Recently, RLVR paradigm has been proved effective for improving the reasoning performance of LLMs.



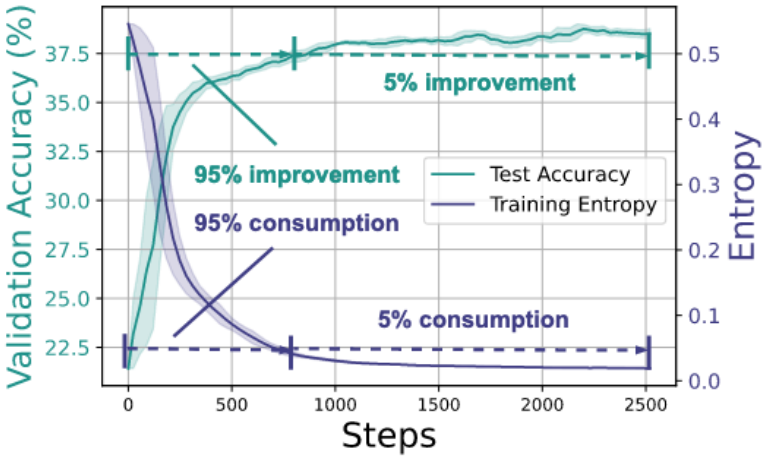


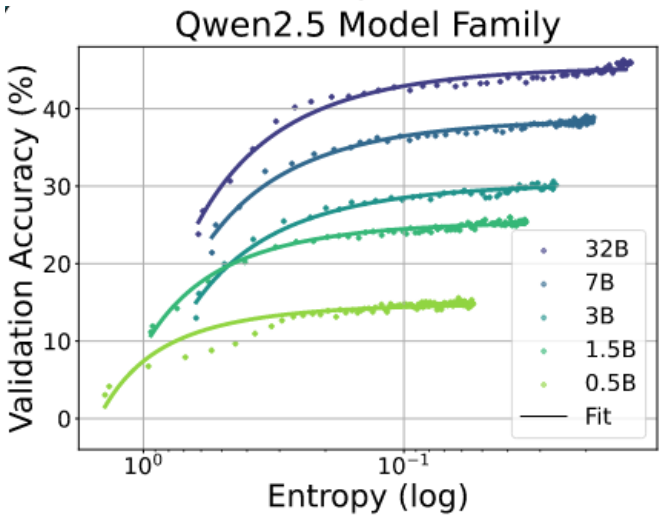




Exploration-Exploitation Tradeoff

However, RLVR algorithms like GRPO often drive policies toward over determinism for robustness — ***at the cost of output diversity****.*





Cui et al., 2025

1 RLVR fine-tuning

verifiable rewards · group-relative sampling

Fine-tune LLMs on math/code with verifiable rewards. GRPO samples R responses per prompt and normalizes rewards group-wise.

↓ *policy gradient → low entropy*

2 Entropy collapse

reward ↑ forces entropy ↓

Sparse rewards accelerate the collapse. *Cui et al.* fit an exponential law to the trend:

$$R \approx -a \cdot e^{H} + b$$

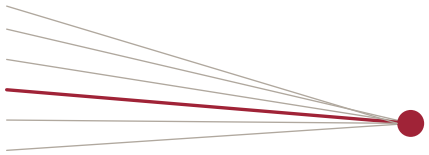
empirical exponential law

↓ *narrow policy distribution*

3 Imbalanced Exploration & Exploitation

over-exploitation → premature convergence

Gradient trapped in a narrow subspace → model settles into one reasoning mode. Seen in both text LLMs and multimodal VLMs.



all paths collapse to one

The exploration–exploitation tradeoff is the central bottleneck in scalable RL.

Related Work: Token-Level Diversity

The community has converged on token entropy as the lever — but this view is fundamentally limited.

What's been tried

token-level diversity levers

Objective

Redesign training objective

To keep sampling token-level diversity, including:

- **Reward reshaping:** replace Pass@1 with Pass@K
- **Entropy encouraging:** incorporate entropy-related metric into policy loss or reward

Tokens

Anchor to salient tokens

To focus on critical tokens that related to diversity and give them more credits.
For example, DAPO, KL-Cov, FR3E, ...

Why it partially works

short-term wins on math & code

Delay

Slows convergence

Keeps training from locking in too early on one reasoning mode.

Breadth

Prevents mode collapse

Distribution stays wider — Pass@K metrics improve on standard math benchmarks.

Gains

Measurable math gains

GSM8K, MATH, AIME: a few points of headroom vs. vanilla RLVR.

Why it hits a wall

the symptom isn't the cause

Conflict

Fights the RL objective

Entropy bonus pushes against the very gradient that makes RL learn — each side fights the other.

VLMs

Breaks on multimodal

Token entropy is a poor proxy once images enter the loop; per-token signal is dominated by text surface.

Proxy

Noisy, surface-level

Token entropy mixes semantic branching with mere lexical variation — you regularize noise along with signal.

We hypothesize that token entropy collapse is rather a superficial symptom than the root cause.

Dynamic Analysis on Latent Exploration

Grounded in Koopman operator theory, we connect latent dynamics with exploration in LLMs and introduce DSD, ***a spectral diagnostic of dynamical diversity****.*

LLM a Stochastic Autonomous System

A stochastic dynamical view of LLM latent space computation:

$$x_t = \mathcal{F}(x_{t-1}, \omega_t), \text{where } \omega_t \sim \mathrm{P}_\omega .$$

Where the stochasticity ω is determined by decoding knobs including temperature, top-p and top-k.

When does this noise become meaningful exploration?

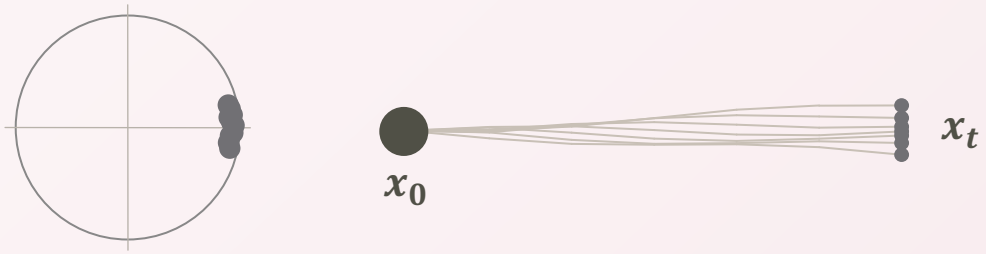
Stochastic inputs → diverse trajectories only if ***the intrinsic hidden state dynamics are rich enough to propagate and amplify them****.*

Ture exploration lives the geometry of latent dynamics — the capacity of $\mathcal{F}$ to translate noise into diverse reasoning trajectories.

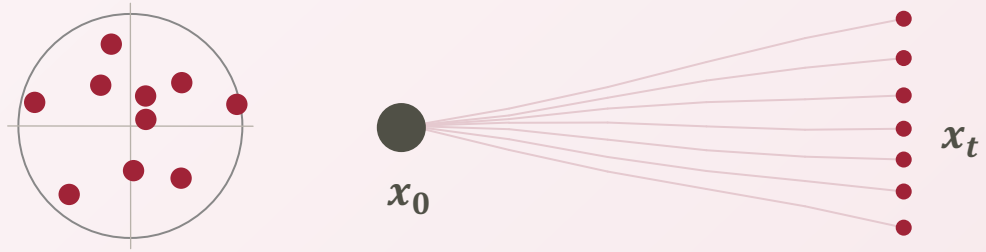
Dynamic Spectral Dispersion (DSD)

We propose a new metric for describing the heterogeneity of dynamical modes, mathematically defined as:

$$DSD(x) \triangleq Var(|\Lambda|), where\ \mathcal{K}\,\Phi = \Phi\,\Lambda .$$

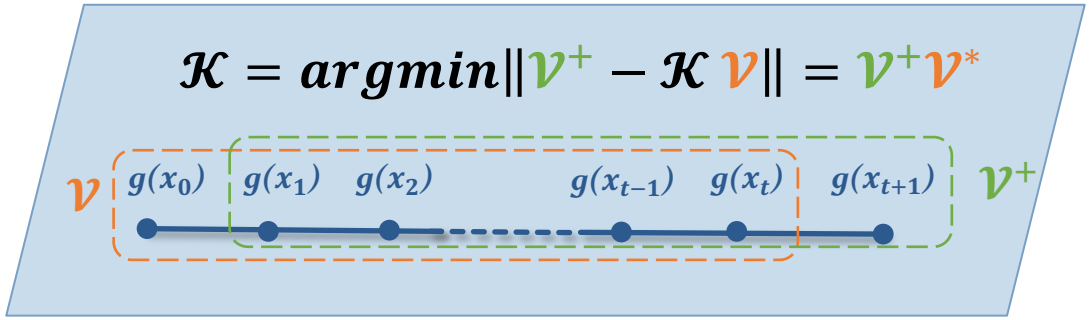


$|\lambda_i|$ *bunched together ⇒ low DSD ⇒ poor exploration*



$|\lambda_i|$ *spread across scales ⇒ high DSD ⇒ rich exploration*

Fitting Koopman Dictionary for Hidden States



Dynamical Mode Decomposition (DMD)

Specifically, the Koopman observables g are parameterized by a single linear layer W followed by a sigmoid activation $\sigma(\cdot)$:

$$g(x) = \sigma(Wx), \quad W \in \mathbb{R}^{d\times m}, \tag{8}$$

where m represents the dimensionality of the approximated Koopman operator. The dictionary W is optimized using latent trajectories $\{x_i\}_{i=1}^{B\times R}$ collected from the last hidden-layer of initial policy, where B denotes the policy training batch size. The optimization objective for W is to minimize the spectral residual of the Koopman operator:

$$W = \arg\min \frac{1}{BR} \|(\mathcal{V}^+ - \mathcal{K}\mathcal{V})\Phi\|_F^2, \tag{9}$$

Two key issues of performing Koopman-based analysis on LLM hidden states:

Find a proper Koopman dictionary This requires learning expressive yet tractable observables that can capture the underlying nonlinear dynamics without suffering from the curse of dimensionality.

Accurate Koopman spectrum approximation This for principled criteria and robust estimation techniques to distinguish true dynamical modes from numerical artifacts introduced by finite-dimensional approximations.

ResKoopNet — *ICML 2025*

Replace the traditional hand-crafted Koopman dictionary with a parameterized MLP.

Optimize the MLP through an objective derived from the residual loss in ResDMD.

Residual DMD — *CPAM 2024*

Use Galerkin method to estimate the residual of each approximated Koopman modes.

Filter out the modes that exceed the certain threshold and preserve remain accurate modes.

Method: ReLaX

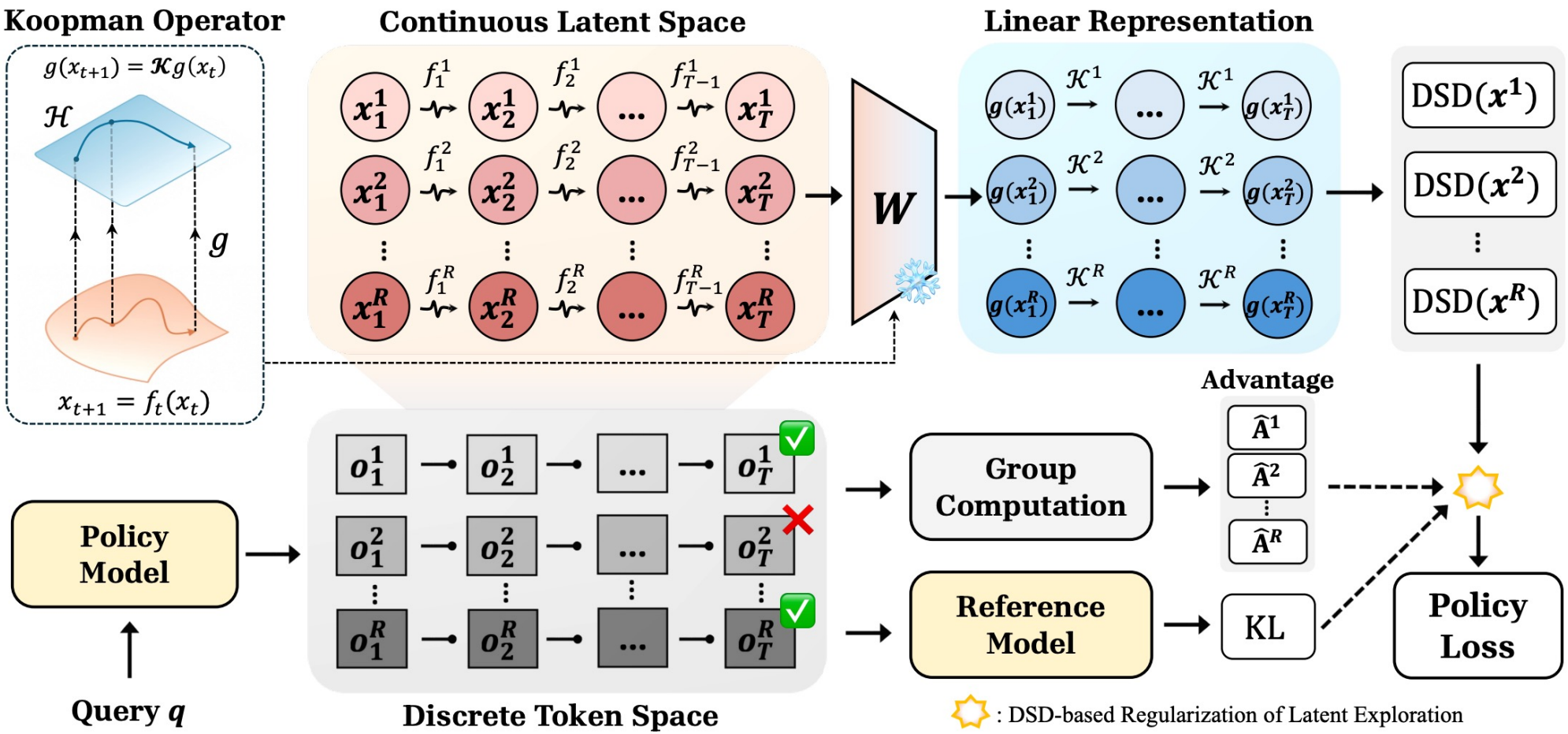
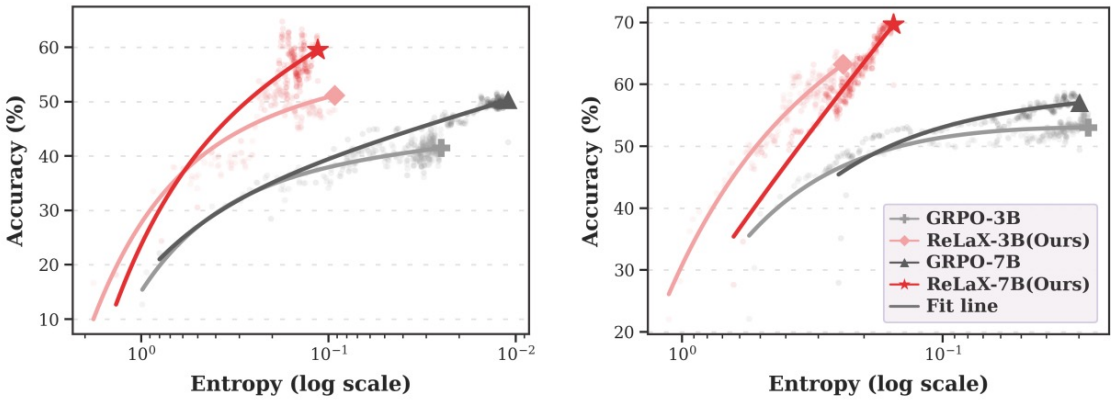


Figure 2. **Overview of ReLaX.** Grounded in Koopman operator theory (upper left), ReLaX employs a neural Koopman dictionary (frozen after one step of learning) during policy optimization to linearize the latent dynamics of last-layer hidden states. This transformation allows us to assess the flexibility of policy's internal computations through the proposed DSD. The DSD score for each trajectory is subsequently integrated into the GRPO objective, mitigating computational rigidity and enabling a more effective exploration–exploitation tradeoff.

Experimental Results Across 3B & 7B Scales



(a) LLMs (Text-only) (b) MLLMs (Vision-language)

Performance-Entropy dynamics

Model	MathVista *testmini*	MathVerse *testmini*	MathVision *test*	DynaMath *overall*	MMMU *val*	MMStar *overall*	EMMA *overall*	Average
General Multimodal LLMs								
Qwen2-VL-7B [39]	58.2	19.7	16.3	42.1	54.1	60.7	20.2	38.8
Qwen2.5-VL-3B [2]	62.3	33.5†	21.2	40.0†	46.3	55.9	19.2†	39.8
Qwen2.5-VL-7B [2]	68.2	49.2	25.1	53.2	54.3	63.9	21.5	47.9
Qwen2.5-VL-72B [2]	74.8	57.6	38.1	-	70.2	70.8	-	-
Intern2-VL-8B [5]	58.3	22.8	17.4	39.7	51.2	61.5	19.8	38.7
Intern2.5-VL-8B [5]	64.4	39.5	19.7	-	56.0	-	20.6	-
Llava-OV-7B [22]	63.2	26.2	18.5	-	48.8	-	18.3	-
Kimi-VL-16B [35]	68.7	44.9	21.4	-	55.7	-	-	-
Reasoning Multimodal LLMs								
MM-Eureka-7B [27]	73.0	50.3	26.9	54.4†	55.2	64.3†	23.5	49.7
MM-Eureka-8B [27]	67.1	40.4	22.2	-	49.2	-	21.5	-
Vision-R1-7B [50]	73.5	52.4	27.2	52.0†	54.7	60.9†	22.4	49.0
R1-VL-7B [51]	63.5	40.0	24.7	45.2	44.5	60.0	8.3	40.9
OpenVLThinker-7B [12]	70.2	47.9	25.3	38.6†	52.5	56.3†	26.6	45.3
VL-Rethinker-7B [36]	74.9	54.2	32.3	55.2†	56.7	64.2†	29.7	52.5
SRPO-7B [18]	75.8	55.8	32.9	-	57.1	-	29.6	-
ReLaX-VL-3B (Ours)	70.7	46.2	27.6	52.2	52.2	60.7	26.9	**48.1**
ReLaX-VL-7B (Ours)	77.1	55.7	30.2	55.9	57.4	65.5	30.6	**53.2**

VLM (Qwen2.5-VL) benchmarking results

Model	Size	MATH500 *Mean@1*	Minerva *Mean@1*	AMC22 *Mean@32*	AMC23 *Mean@32*	AIME24 *Mean@32*	AIME25 *Mean@32*	Average
Qwen2.5-base	3B	18.4	2.6	12.5†	7.0	0.6	0.3	5.9
+Vanilla GRPO[31]	3B	69.2	23.5†	25.9†	51.6	7.5	4.5	26.0
+DCPO[44]	3B	71.2	-	-	55.8	7.5	4.7	-
+ReLaX (Ours)	3B	71.6	26.1	44.8	63.5	8.8	6.2	**31.6**
Qwen2.5-base	7B	64.6	5.7	21.4†	30.0	0.3	0.1	17.4
+SimpleRL[49]	7B	78.2	38.6	39.5†	62.5	15.6	8.9†	34.8
+DAPO[46]	7B	77.8	35.3	-	60.0	18.1	11.5	-
+KL-Cov[10]	7B	80.8	38.2	-	61.4	22.6	12.9	-
+FR3E[54]	7B	79.0	39.0	49.2†	67.5	25.2	14.8†	39.2
+ReLaX (Ours)	7B	82.4	39.1	65.4	84.1	19.7	13.8	**43.5**
Qwen2.5-Math-base	7B	52.4	10.7	25.4†	52.2	16.6	6.3	23.4
+SimpleRL[49]	7B	77.4	32.0	57.0†	60.8	26.7	9.3	37.6
+DAPO[46]	7B	81.6	38.2	-	62.5	31.6	14.9	-
+DCPO[44]	7B	82.5	-	-	79.8	38.8	17.2	-
+R1-zero-Div[45]	7B	77.2	31.6†	50.2†	58.0†	17.5†	10.0†	34.9
+FR3E[54]	7B	82.2	40.8	64.1†	67.5	26.7	18.1†	42.8
+ReLaX (Ours)	7B	85.6	43.4	71.9	88.9	36.9	17.3	**49.1**

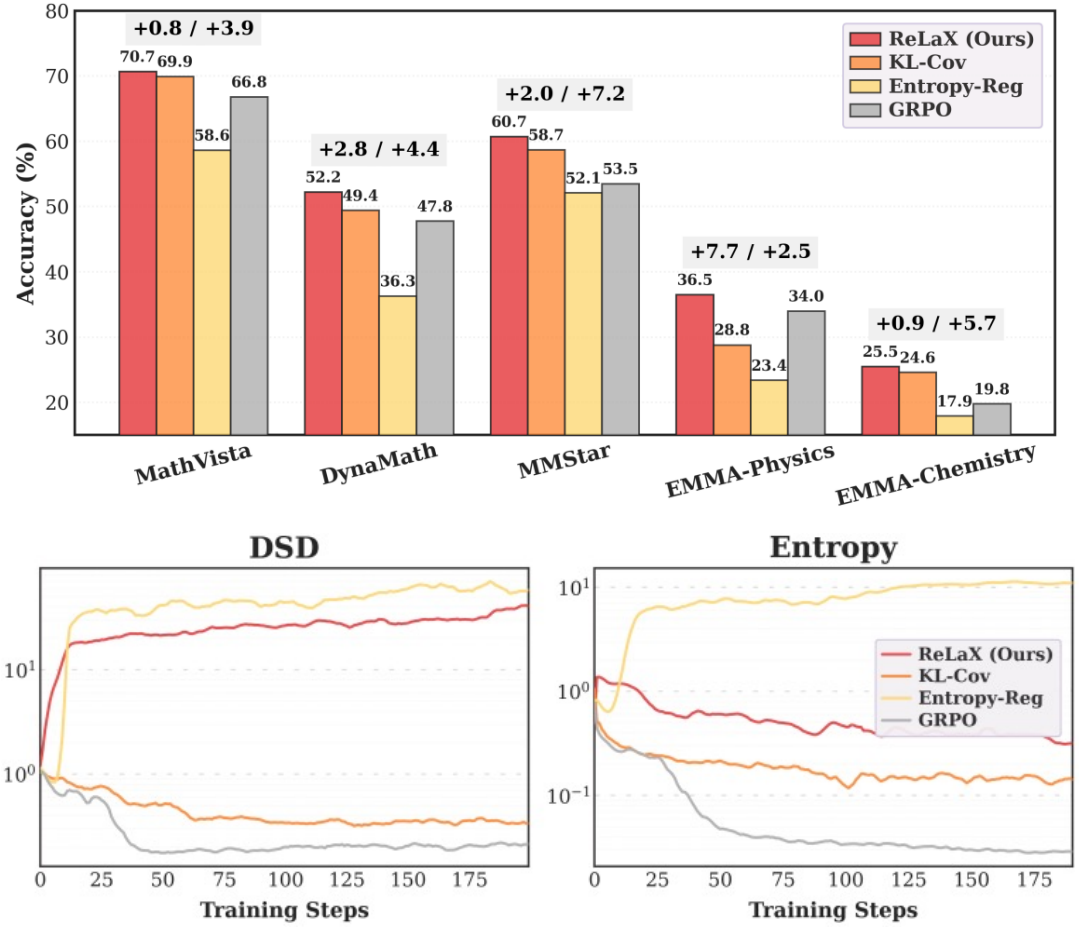
Model	Size	MATH500 *Mean@1*	Minerva *Mean@1*	AMC22 *Mean@32*	AMC23 *Mean@32*	AIME24 *Mean@32*	AIME25 *Mean@32*	Average
Llama3.2-Instruct	3B	46.8	15.4	16.1†	22.0	8.5	0	18.1
+Vanilla GRPO[32]	3B	55.4	22.8	26.4	40.0	16.3	1.4	27.1
+ReLaX (Ours)	3B	57.0	23.5	39.0	52.8	18.9	3.3	32.4
Δ (ReLaX - GRPO)	3B	+1.6	+0.7	+12.6	+12.8	+2.6	+1.9	+5.3
Qwen3-Base	4B	63.8	28.3	29.1†	38.9	9.4	5.3	29.1
+Vanilla GRPO[32]	4B	83.0	38.9	42.6	51.2	24.9	23.8	44.1
+HICRA [39]	4B	89.0	42.5	-	54.0	31.0	27.6	-
+ReLaX (Ours)	4B	90.2	48.5	52.6	64.5	30.9	27.6	52.3
Δ (ReLaX - GRPO)	4B	+6.2	+9.6	+10.0	+13.3	+6.0	+3.8	+8.2

LLM benchmarking results, including Qwen2.5, Qwen3 and Llama3.2 model families

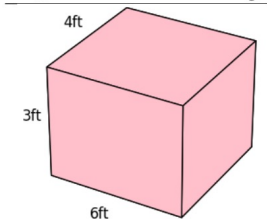
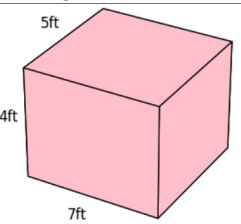
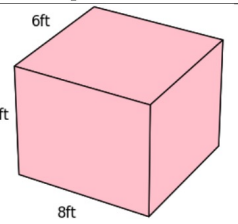
Comparative Analysis on VLMs

On VLMs, we also compare with existing methods that focus on token-level diversity, specifically:

- ***Entropy regularization***
- ***KL-Cov***



Question: Given a rectangular prism with length l, width w, and height h, calculate its perimeter.

 Question Variant 1	 Question Variant 2	 Question Variant 3
ReLaX: To find the perimeter of a rectangular prism, ... Given: - Length (L) = 6 ft - Width (W) = 3 ft - Height (H) = 4 ft The formula for the perimeter P of a rectangular prism is: $P = 4(L + W + H)$ Substituting the given values into the formula: $P = 4(6 + 3 + 4)$ $P = 4(13)$ $P = 52$ So, the perimeter of the rectangular prism is $\boxed{52}$ feet. **[CORRECT]**	**ReLaX:** To find the perimeter of a rectangular prism, ... Given: - Length (L) = 7 ft - Width (W) = 4 ft - Height (H) = 5 ft The formula for the perimeter P of a rectangular prism is: $P = 4(L + W + H)$ Substituting the given values into the formula: $P = 4(7 + 4 + 5)$ $P = 4(16)$ $P = 64$ So, the perimeter of the rectangular prism is $\boxed{64}$ feet. **[CORRECT]**	**ReLaX:** To find the perimeter of a rectangular prism, ... Given: - Length (L) = 8 ft - Width (W) = 3 ft - Height (H) = 6 ft The formula for the perimeter P of a rectangular prism is: $P = 4(L + W + H)$ Substituting the given values into the formula: $P = 4(8 + 3 + 6)$ $P = 4(17)$ $P = 68$ So, the perimeter of the rectangular prism is $\boxed{68}$ feet. **[CORRECT]**
KL-Cov: To find the perimeter of the rectangular prism, ... The formula for the perimeter of the edges of a rectangular prism is: $P = 4(l + w + h)$ where l, w, and h are the length, width, and height of the prism, respectively. Substituting the given dimensions (6 feet, 3 feet, 4 feet): $P = 4(6+3+4) = 4 \times 13 = 52$ feet However, the problem seems to be asking for the perimeter of the faces, which is not a standard geometric term either. Given the dimensions provided, if we assume the problem is asking for the perimeter of the faces that would form a net of the prism, we need to consider the perimeter of the faces, ... Therefore, the correct answer is: $\boxed{52}$ **[CORRECT]**	**KL-Cov:** To find the perimeter of the rectangular prism, ... **The dimensions given for the base of the rectangular prism are 7 feet and 4 feet.** The perimeter P of a rectangle is calculated using the formula: $P = 2 \times (length + width)$ Here, the length is 7 feet and the width is 4 feet. Substituting these values into the formula, we get: $P = 2 \times (7+4) = 2 \times 11 = 22$ feet So, the perimeter of the rectangular prism (interpreted as the perimeter of the base) is: $\boxed{22}$ **[WRONG]**	**KL-Cov:** To find the perimeter of the rectangular prism, **we need to calculate the perimeter of each pair of opposite faces and then sum them up**. The perimeter of each pair of opposite faces is calculated as follows: 1. For the faces with dimensions 8 feet by 3 feet, the perimeter of one face is $2 \times (8 + 3) = 2 \times 11 = 22$ feet. Since there are two such faces, the total contribution to the perimeter from these faces is $2 \times 22 = 44$ feet. 2. ..., the total contribution to the perimeter from these faces is $2 \times 28 = 56$ feet. 3. ..., the total contribution to the perimeter from these faces is $2 \times 18 = 36$ feet. Adding these contributions together gives us the total perimeter of the rectangular prism: $44 + 56 + 36 = 136$ Therefore, the perimeter of the rectangular prism is $\boxed{136}$. **[WRONG]**

Summary & Future Directions

Viewing LLMs as dynamical systems opens both a deeper understanding and a clear path forward.

Infrastructure

- Hidden state-centric research demands new infrastructure as models scale
- Storing full latent trajectories across long CoT and large batches requires more memory
- Latent analysis operations like Koopman learning need optimized kernels

Scaling the tools, not just models.

Beyond Koopman

- Explore broader tools from dynamical systems and control theory
- Goal: a unified theoretical toolkit for analyzing and regulating LLM internal computation

Koopman is just a starting point.

Complex Agentic Scenarios

- Scaling the perspective to multi-step, multi-agent systems.

Challenging but interesting!

Thank you !