Implicit Thinking

Reasoning in the Latent/Continuous Space

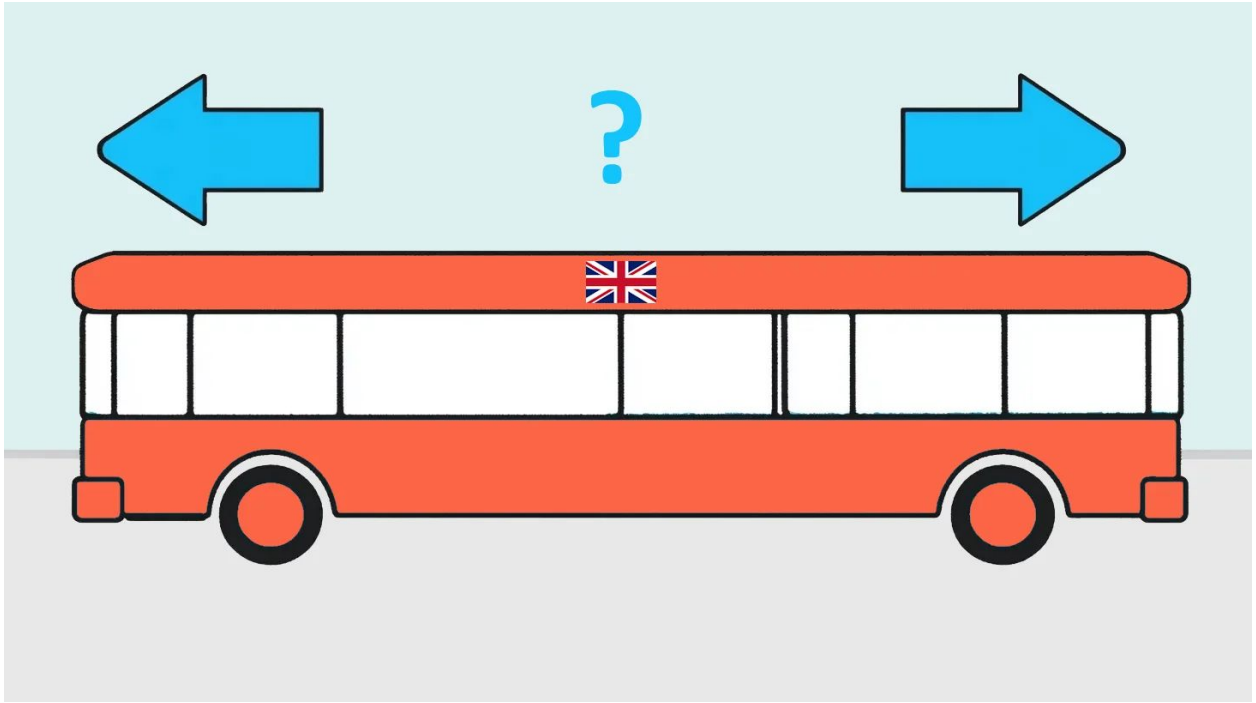
NAVER LABS Europe NLP Seminar
March 17, 2026

Thomas Palmeira Ferraz
Pierre Erbacher

Can you solve this problem?

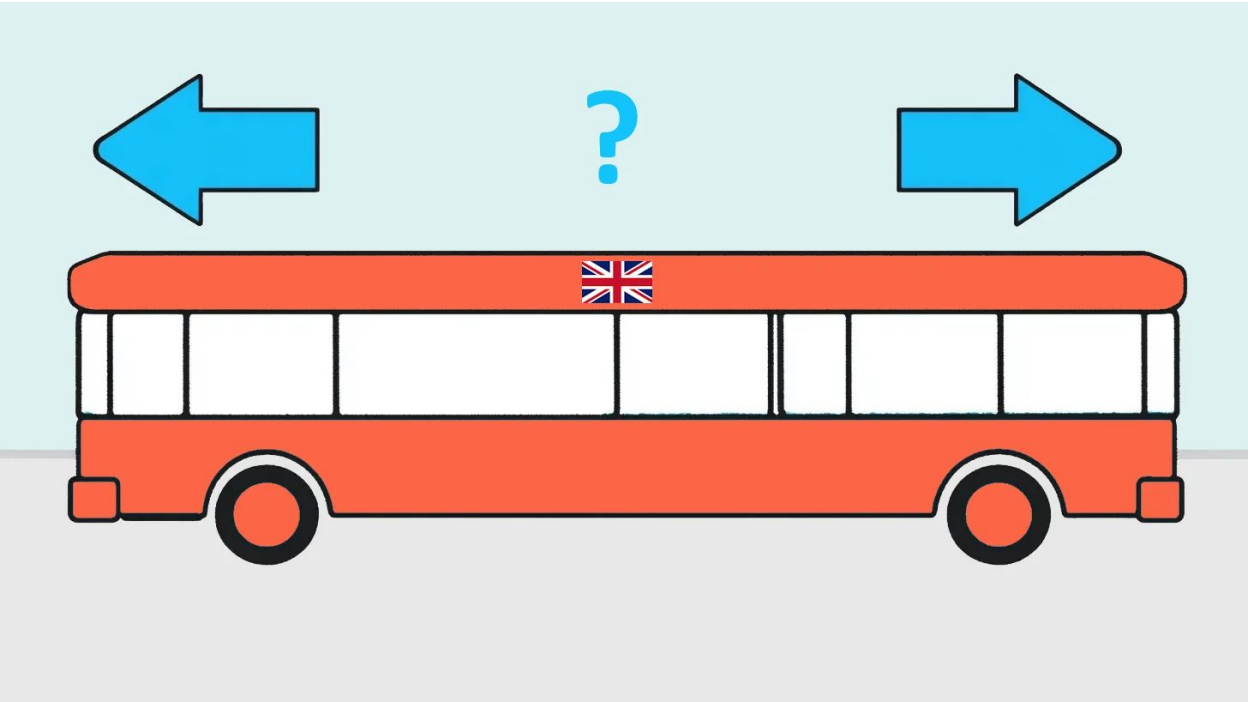
Can you solve this problem?

Which direction is the bus moving? Left or right?



Can you solve this problem?

Which direction is the bus moving? Left or right?



RIGHT

How LLMs Would Solve This? Qwen 3.5-VL-27B

The user wants me to solve a visual riddle based on an image of a bus.
1. Analyze the image:
Subject: A red single-decker bus. It looks like a classic London bus, indicated by the Union Jack flag.
Context: There are two blue arrows pointing in opposite directions (left and right) and a question mark above the bus. This is a classic "which way is the bus going?" puzzle.
Key Visual Cues:
Doors: I need to look for doors. In this specific drawing, there are no visible doors on the side facing the viewer. Wait, let me look closer. The windows are just panes of glass. There are no sliding doors visible on this side. This is a bit tricky. Usually, these puzzles rely on the absence of doors indicating the driver's side or passenger side.
Steering Wheel/Driver: I can't see inside the windows clearly enough to spot a steering wheel. The windows are just white rectangles.
Flag: There is a Union Jack flag in the middle. This suggests it's a UK bus. In the UK, you drive on the left side of the road. Therefore, the driver sits on the right side of the vehicle (from the driver's perspective), which means the doors are on the left side of the vehicle (from the driver's perspective). Wait, let me re-evaluate standard bus configurations.
UK/London Bus: Drivers sit on the right. Doors are on the left side of the bus (the curb side).
US/Europe Bus: Drivers sit on the left. Doors are on the right side of the bus (the curb side).
Re-evaluating the "Doors" clue: Let's look at the side of the bus again. Is it the front or back?
If it's moving Right: The front would be on the right. The driver would be on the right (in a US bus) or left (in a UK bus). If it's a UK bus, the driver is on the right. So if it's moving right, we are looking at the passenger side (where doors usually are). But I don't see doors. Wait, usually, doors are towards the front or middle.
If it's moving Left: The front is on the left.
Let's look at the shape: This is a very simplified drawing. It doesn't have distinct headlights, taillights, or a sloping front windshield vs a flat rear window. It's symmetrical except for the flag.
Wait, let me look really closely at the "doors". Actually, looking at the spacing of the vertical black lines... they represent window frames. There are no gaps that look like open or closed double doors.
Is there another clue? Ah, the flag. It's a Union Jack. This implies it's a UK bus.
Standard Riddle Logic: The most common version of this riddle shows a bus with *no doors* visible on the side facing you.
If there are no doors, it means the doors are on the *other* side.
Buses have doors on the sidewalk side (curb side).
If it's a UK bus (Union Jack), people drive on the left. So the driver is on the right. The doors are on the left side of the bus.
If the doors are on the other side (not visible), then the side we are seeing is the driver's side (right side).
If we are seeing the right side (driver's side), the bus is facing Left. Why? Because if the bus faces left, the right side is visible. Wait.
Let's try again.
Scenario A: Moving Right. Front is Right. Side visible is Left side. In UK, doors are on Left. So we should see doors. We don't see doors. So it's probably not moving right.
Scenario B: Moving Left. Front is Left. Side visible is Right side. In UK, doors are on Left (hidden). Driver is on Right (visible). So we are looking at the driver's side. We see no doors. This fits. So the bus is moving Left.
Alternative Interpretation (US Bus): What if I ignore the flag?
If it's a US bus, doors are on the Right side.
If moving Right -> Visible side is Left (No doors). This fits.
If moving Left -> Visible side is Right (Doors). We don't see doors. So not moving Left.
So for a US bus, answer is Right.
The Flag is the Key: The flag is a Union Jack. This places the bus in a country where they drive on the left (like the UK).
In the UK, the driver sits on the right-hand side.
The doors are on the left-hand side (curb side).
In the picture, we see a side of the bus with no doors.
Therefore, we are looking at the side *without* doors.
Since the doors are on the left (passenger side), the side without doors is the right side (driver's side).
If we are looking at the right side of the bus, the front must be to the Left. (Imagine sitting in the driver's seat on the right side facing forward - you are facing left in this 2D projection).
Wait, let's visualize.
Bus facing Left: [Front] -------- [Rear]. Right side is visible. (Driver is here). Doors are on the hidden Left side. -> Matches image (no doors visible).
Bus facing Right: [Rear] -------- [Front]. Left side is visible. Doors are on the visible Left side. -> Doesn't match image (should see doors).
Conclusion: The bus is moving to the Left.
Let's double check. Is there any other clue?
Look at the wheels. No help.
Look at the bumper. The front bumper usually sticks out more or has a different shape? Not really distinguishable here.
Look at the "nose". The front left corner looks slightly rounded vs the back left corner. Actually, both ends look identical. It's a symmetric drawing except for the flag.
Wait, is the flag oriented correctly? The flag is centered. It's symmetrical vertically but not horizontally (St Patrick's saltire is diagonal). The flag looks correct.
Let's reconsider the "No Doors" theory. It is the standard solution for this riddle.
Premise: Buses have doors on one side only.
Observation: We do not see doors on this side.
Inference: This is the non-door side.
Location Context: Union Jack = UK/Left-hand traffic.
Traffic Rule: In LHT countries (like UK), buses pick up passengers from the left side of the road. So doors are on the Left side of the bus.
Geometry: If doors are on the Left side of the bus, and we see the Right side of the bus (because we don't see doors), then the bus is facing Left.
Imagine the bus. Left side has doors. Right side has driver.
If bus faces Left: We see the Right side. Correct.
If bus faces Right: We see the Left side (with doors). Incorrect.
What if it's a right-hand traffic country? (e.g., USA, most of Europe).
Doors are on the Right side.
We see the Left side (no doors).
If we see the Left side, the bus is facing Right.
Which context applies? The flag is explicitly the Union Jack. This strongly implies a UK context. Therefore, Left-hand traffic rules apply.
Therefore: Doors are on the Left. Visible side is Right (Driver's side). Bus is facing Left.
Wait, let me look really closely at the image again. Is it possible that what I think are window dividers are actually doors?
No, they are evenly spaced vertical bars. Standard bus windows.
Is it possible there are doors but they are painted the same color? No, usually riddles rely on clear omission.
Is it possible the flag indicates direction? No, flags flap with wind, but here it's static. And it's just a sticker/decal likely.
Let's try a different angle. Look at the windshield.
The front windshield is usually slanted or larger. The rear window is usually smaller or split.
In this drawing, all windows look like uniform rectangles.
However, look at the very front edge (left side). It looks a bit more substantial, like a hood? No, it's flat.
Look at the very back edge (right side). Also flat.
This drawing is extremely stylized and ambiguous regarding aerodynamics. The "No Doors" rule is the intended logic.
Let's refine the "No Doors" logic.
Identify the object: A bus.
Identify the missing feature: Doors are not visible on the side shown.
Apply general knowledge: Buses have doors on the curb side (sidewalk side) for passenger safety.
Determine location: The Union Jack flag indicates the UK (or similar Commonwealth nation), where traffic drives on the left.
Apply traffic rules: In left-hand traffic, the curb is on the left. Therefore, bus doors are on the left side of the vehicle.
Deduce orientation: Since the doors are on the left (hidden), the side visible to us is the right side (driver's side).
Conclude direction: If the right side is visible, the front of the bus is pointing to the Left.
Counter-argument: What if it's just a generic bus illustration and the flag is just decoration?
If I assume generic/right-hand traffic (most of the world):
Doors on Right.
Visible side has no doors -> Visible side is Left.
If visible side is Left, bus is facing Right.
But the flag is specific. It's a deliberate clue. You wouldn't put a specific national flag unless it mattered. The flag dictates the traffic rules.
So, UK rules -> Doors on Left -> Visible side is Right -> Facing Left.
Let's check the flag orientation again.
The flag is the Union Jack.
Strictly speaking, a Union Jack has a "correct" way to fly. The wider white stripe of the St Andrew's cross should be upper-left (hoist side).
In the image, the blue diagonal stripes are symmetric. The white/red cross is central. It's hard to tell top/bottom/left/right perfectly because it's small, but it looks like a standard representation. It doesn't give directional info like a wind-blown flag. It's a decal.
Final Decision Path:
The riddle relies on the absence of doors.
The flag identifies the locale as the UK (driving on the left).
In the UK, bus doors are on the left side.
- We see a side with no doors. Thus, we are looking at the right side.

If we are looking at the right side of a vehicle, and the vehicle is on the road, it is facing Left. (Think about sitting in the driver's seat on the right... you face forward. To your right is the center line. To your left is the curb. Wait. If I am in the driver's seat on the right, facing forward (Left in the diagram), my right hand is towards the top/back of the page? No.
Let's orient ourselves.
Bus is horizontal.
Case 1: Moving Left. Front is Left. Rear is Right. Driver sits on Right side of bus. We are looking at the Right side of the bus. Correct.
Case 2: Moving Right. Front is Right. Rear is Left. Driver sits on Right side of bus. We are looking at the Left side of the bus. Incorrect (because we'd see doors).
Wait, let me re-verify the "Driver sits on Right" part for UK.
UK: Drive on Left. Driver sits on Right. Correct.
Doors are on the Left (curb side). Correct.
So, Left side of bus = Doors. Right side of bus = Driver/No Doors.
Image shows: No Doors.
Therefore, Image shows: Right side of bus.
If Image shows Right side of bus, and we see the profile...
If bus moves Left -> Front is Left. We see the side that is on the Right relative to the driver's forward motion. Yes, that's the right side of the vehicle.
If bus moves Right -> Front is Right. We see the side that is on the Left relative to the driver's forward motion. Yes, that's the left side of the vehicle.
So, since we see the Right side (no doors), the bus is moving Left.
Is there any possibility it's moving Right?
Only if it's a right-hand drive country (like France/Germany/USA).
But the flag says UK.
So the answer must be Left.
et's write down the final explanation.
Start with the main clue: The missing doors.
Explain the door placement rule: Doors are on the curb side.
Use the flag to determine traffic side: Union Jack = UK = Left-hand traffic.
Combine: Doors are on the left side of the bus.
Observe: We don't see doors, so we are looking at the right side (driver's side).
Conclusion: If we see the right side, the bus is facing left.
Wait, is it possible the "front" is defined by the arrow? No, the arrows are options.
Is it possible the shape gives it away? Some might argue the front nose is slightly more pronounced on the left? Looking at the image... the left end looks a tiny bit more rounded/protruding than the right end? It's very subtle if so.
The left bumper area looks a bit thicker. The right bumper area looks flatter. If the left is the front, it's moving Left. This supports the logical deduction.
Also, usually the driver's cabin window is slightly different or angled. Here, the first window pane on the left looks slightly taller/different? Hard to say. The "No Doors" logic is the robust one intended for this riddle.

Summary for the user:
Point out there are no doors visible.
Explain that buses have doors on the curb side.
Note the Union Jack flag indicates the UK.
Explain that in the UK, you drive on the left, so doors are on the left side of the bus.
Since doors aren't visible, we are looking at the right side (driver's side).
Therefore, the bus is facing/moving to the Left.
Let's double check the "Right side visible means moving Left" logic.
Vehicle coordinates: Front, Back, Left, Right.
If facing Left (West): Front is West. Right side is South (facing viewer). Left side is North (away from viewer).
If facing Right (East): Front is East. Left side is South (facing viewer). Right side is North (away from viewer).
Okay, so if we see the Right side, the bus is facing Left.
My previous logic holds up.
Wait, is it possible the flag is upside down?
If the flag is inverted, maybe it signals distress? Unlikely for a riddle.
Is the flag pointing? Flags usually fly from the front to the back. If the pole was at the front, the flag would trail behind. Here it's just a square patch. No directional info.
Final Answer Formulation:
Direct Answer: Left.
Reasoning:
Clue 1: No Doors. Notice there are no doors on the side of the bus shown.
Clue 2: The Flag. The Union Jack indicates this is a UK bus.
Logic: In the UK, traffic drives on the left side of the road. Therefore, the bus doors (for passengers to enter safely onto the sidewalk) are on the left side of the vehicle.
Conclusion: Since we are looking at the side *without* doors, we must be looking at the right-hand side (the driver's side). If the right side is visible, the front of the bus is pointing to the left.
Just to cover bases, I should mention that if it were a US bus (right-hand traffic), the answer would be Right. But the flag makes it specific.
Based on the visual clues, the bus is moving to the left.
Here is the step-by-step breakdown of the riddle:
1) Look for the doors: If you look closely at the side of the bus facing us, you will notice there are no doors.
2) Where do bus doors go? Bus doors are always on the curb side (the sidewalk side) so passengers can get on and off safely.
3) Check the location: There is a Union Jack flag on the roof. This tells us the bus is in the United Kingdom (or a country with similar driving laws).
4) Apply the traffic rules: In the UK, drivers drive on the left side of the road. This means the bus doors are located on the left side of the vehicle.
5) Conclusion: Since the doors are on the left side (hidden from view), the side we are seeing must be the right side(the driver's side). If you are looking at the right side of a car/bus, the front is pointing to the left.

LEFT?

How LLMs Would Solve This?

Model	# Words to Solve the Riddle	Correct
Kimi K-2.5 (1T)	7,143	Yes
Qwen 3.5-VL-27B	2,757	No
Qwen 3.5-VL-9B	1,951	No
Qwen 3.5-35B-A3B	3,897	Yes

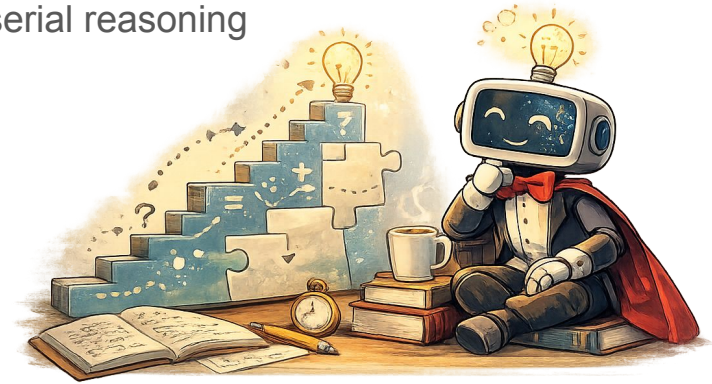
Agenda

- Why Latent Reasoning?
- L-SEQ: Latent Sequence Reasoning
- L-LOOP: Latent Looped Reasoning
- L-OPT: Latent Optimization
- Alternative Architectures
  - Large Concept Models
  - Diffusion Models
- Multimodality
- Evidences, Challenges, Limitations
- Open problems and future directions

Preliminaries

Explicit textual CoT: Breakthrough for TTS

- Test-time scaling
  - Human cognitive model: System 1 (autopilot) vs. System 2 (thinking)
  - System 2 LLMs improve performance by generating intermediate results.
- Explicit textual CoT was a breakthrough for TTS
  - Better performance on many multi-step tasks with *“Let’s think step-by-step”*
  - Supervision made straightforward: train on intermediate traces, not just answers
  - A human-readable interface for debugging and steering (interventions / RLVR)
  - A partially observable “working memory” channel for serial reasoning



Language is **the** bottleneck for reasoning progress

- Comparison with most efficient case: humans
  - Human cognition often transcends discrete linguistic symbols, involving abstract, continuous, multimodal or multi-conceptual representations that resist precise verbalization (Wittgenstein, 1922)
  - Complex emotions like **nostalgia** and **bittersweetness** are continuous blends of **joy**, **sadness**, and **multimodal** memory that defy expressions from a fixed vocabulary (Pinker, 1994)

Language is **the** bottleneck for reasoning progress

- Comparison with most efficient case: humans
  - Human cognition often transcends discrete linguistic symbols, involving abstract, continuous, multimodal or multi-conceptual representations that resist precise verbalization (Wittgenstein, 1922)
  - Complex emotions like **nostalgia** and **bittersweetness** are continuous blends of **joy**, **sadness**, and **multimodal** memory that defy expressions from a fixed vocabulary (Pinker, 1994)
- Consequences of reasoning on discrete textual space
  - **Expressive redundancy**: many reasoning tokens are syntactically needed but functionally weak (“so”, “the”, etc.).
  - **Stylistic overfitting**: models can learn artifacts of reasoning traces rather than reasoning itself.
  - **Semantic bottleneck**: forcing continuous or multi-concept states into a discrete linear chain of fixed vocabulary inevitably leads to information loss.
  - **Serialization bottleneck**:
    - reasoning must unfold one token at a time.
    - Verbose, expensive and limited parallel exploration.

The case for *de-linguistifying* reasoning

Promises:

- Less token-level computation → faster inference
- Richer, more compact reasoning representations
- Potential to explore multiple reasoning trajectories in parallel

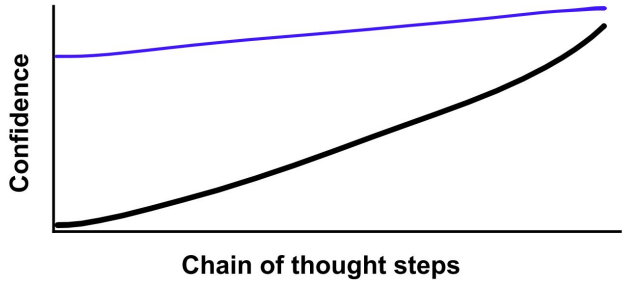
Challenges:

- Harder direct supervision
- Training and alignment difficulty
- Lower transparency / weaker debuggability
- Evaluation gap: did the model really reason, or just fit input–output patterns?

CoT and Performative Reasoning

Faithfulness

- **Unfaithful explanation:** CoT can systematically misrepresent LLM internal reason for a its prediction [1]
- **LLMs can fake reasoning:** producing coherent-looking rationales for contradictory / bias-driven answers [1,2]
- **(≠ Influence):** CoT can affect final answer even if not faithful to model's actual reasoning [3]
- **Fragile Monitorability**: CoT can help oversight, but it is imperfect and may fail to reveal important internal states [4]



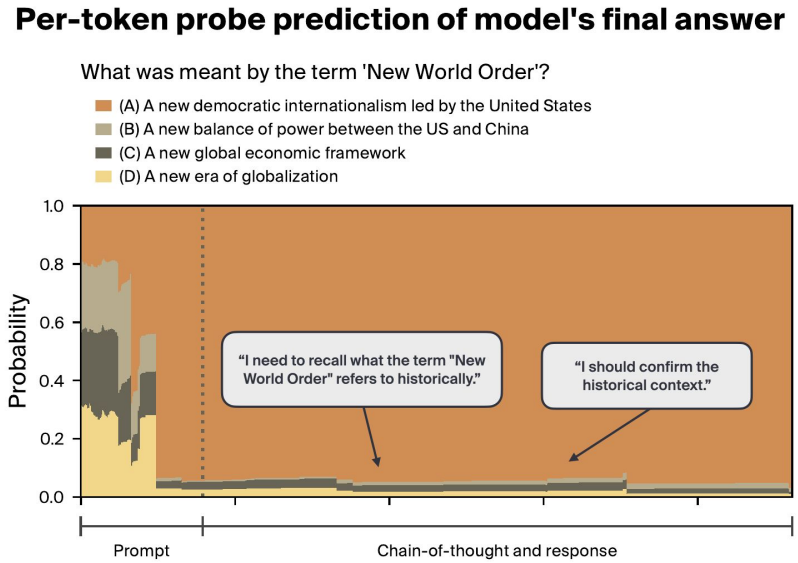
Source: Lewis-Lim et al. [3]

[1] Turpin et al., Language Models Don't Always Say What They Think: Unfaithful Explanations in Chain-of-Thought Prompting (NeurIPS, 2023)
[2] Arcuschin et al., Chain-of-Thought Reasoning in the Wild is not Always Faithful (2025)
[3] Lewis-Lim et al., Analysing Chain of Thought Dynamics: Active Guidance or Unfaithful Post-hoc Rationalisation? (EMNLP 2025)
[4] Korbak et al., Chain of Thought Monitorability: A New and Fragile Opportunity for AI Safety (2025)

CoT and Performative Reasoning

Performative (or decorative) reasoning:

- **Early decisions:** in some cases, the final answer becomes decodable well before the full CoT finishes [1]
- **Causality**: many CoT steps have little causal impact on the final prediction, even “aha” or self-verification steps can be decorative [2]

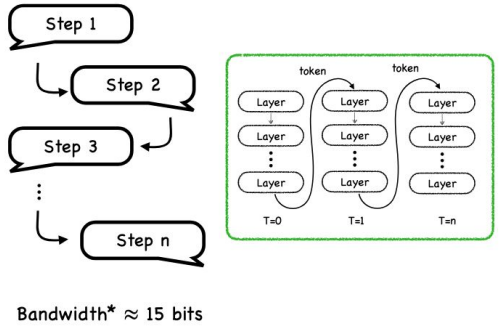


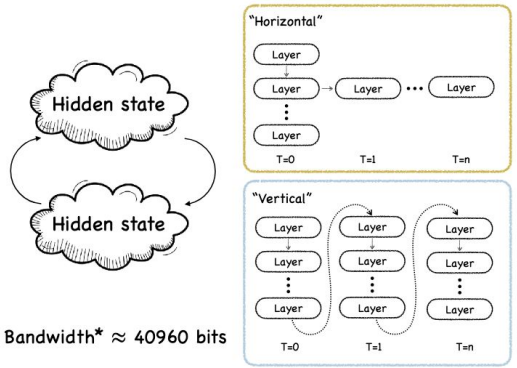
Source: Boppana et al. [1]

[1] Boppana et al., Reasoning Theater: Disentangling Model Beliefs from Chain-of-Thought (2026)
[2] Zhao et al., Can Aha Moments Be Fake? Identifying True and Decorative Thinking Steps in Chain-of-Thought (2025)

LLM Reasoning

- LLM Reasoning can be modeled as a **two-stage process**:
  - from an input, the model builds an intermediate trace, then produces the answer.
- **Explicit reasoning**: trace is verbalized as text.
- **Implicit reasoning**: trace is not verbalized or discretized.
  - **Latent reasoning**: internal reasoning explicitly in latent/hidden-state form.
- Key distinction is not whether there is a trace, but whether it is exposed to the user.





***For FP-16**: Explicit reasoning transmits discrete tokens **(≈ 15 bits / step),** whereas latent reasoning exchanges 2560-dimensional FP-16 hidden states **(≈ 40, 960 bits / step),** revealing a ~2.7 × 10^3-fold bandwidth gap between the two approaches. **Source**: Zhu et al. A Survey on Latent Reasoning (July, 2025)

Latent Reasoning: Taxonomy

- **L-SEQ - Latent Sequence Reasoning**
  - Generate a sequence of latent thoughts before producing the final answer
  - Inspiration from early Test-Time Scaling and Compression research
  - Also referred to as “token-wise horizontal level optimization”
- **L-LOOP - Latent Looped (Recurrent) Reasoning**
  - Methods that deepen reasoning through iterative computation across layers for each token
  - Inspiration from Sparsity / Early-Exit / MoE-MoD research
  - Also referred to as “layer-wise vertical level optimization”
- **L-OPT - Latent Optimization (Control)**
  - methods that focus on optimizing hidden reasoning
  - mainly without architecture modifications

Latent Reasoning: Goals

Usual goals reported in the literature can be summarized in:

- Accuracy / reasoning quality
  - Improve performance on reasoning-heavy tasks (math, logic, coding, algorithmic tasks) at a given compute budget.
- Efficiency / compute
  - Reduce FLOPs, latency, or token count while keeping accuracy roughly flat or slightly improved.
  - Includes: fewer generated tokens, chunked decoding, adaptive depth, parallel drafting, etc.
- Data efficiency / Pre-Training efficiency
  - Get more learning signal per token (e.g., using latent thoughts or concepts during pretraining to reduce data requirements or improve scaling).
- Out-of-distribution / systematic generalization
  - Improve robustness to distribution shifts and compositional generalization (e.g., new problem sizes, unseen graph structures, or systematically recombined primitives)

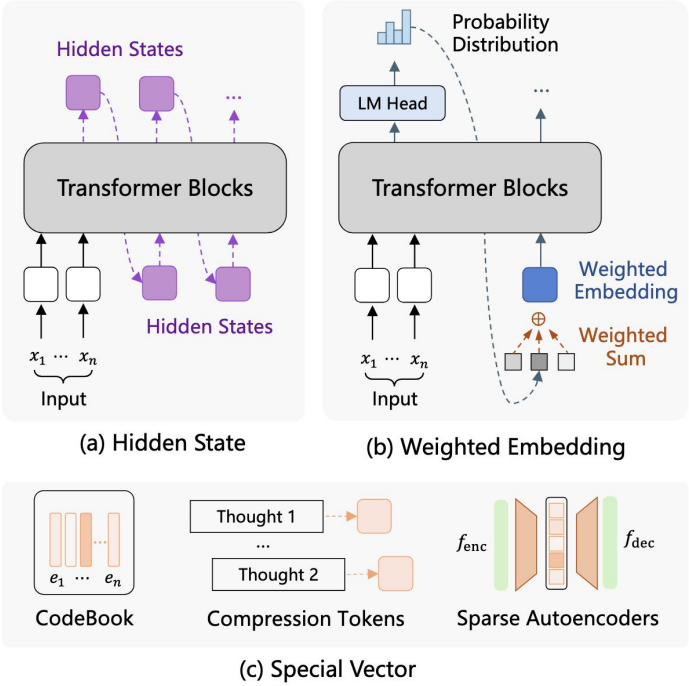
L-SEQ
Latent Sequence Reasoning

Sequential Latent Reasoning (L-SEQ)

- Early TTS efforts: padding / pause tokens / planning tokens
  - In parallel: trials to embed the logic of an explicit CoT directly into the model's latent representations without extending the visible token sequence.
- 3 key aspects:
  - Representation Initialization
  - Model Optimization
  - Inference Exploration

L-SEQ: Representation Initialization

- A main design choice!
  - dictates the nature of the latent space
  - strongly influences the model's capacity to reason
- Hidden State 🚀
- Weighted Embedding
- Special Vector



🚀 Indicates approaches with stronger empirical evidence of success

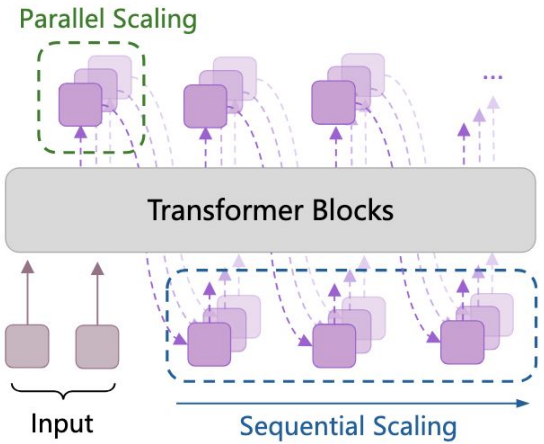
L-SEQ: Model Optimization

- Pre-Training: Beyond NTP
- Post-Training:
  - Indirect SFT: hope to induce learning useful latent representations
    - KL-Div / VAEs / Regularization
  - Direct SFT 🚀
    - Representation optimization with “gold” targets.
    - KV-Cache
  - RL 🚀
    - Does not rely on CoT traces for supervision
- Token-level vs. Trajectory-level

🚀 Indicates approaches with stronger empirical evidence of success

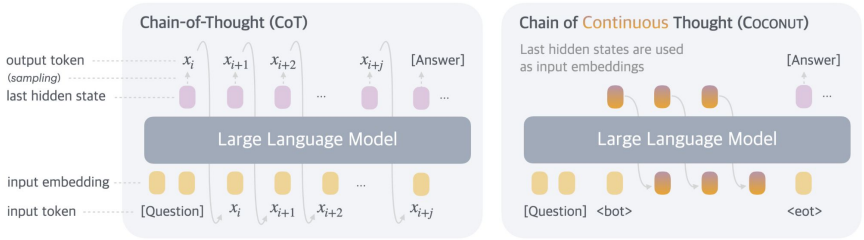
L-SEQ: Inference Exploration

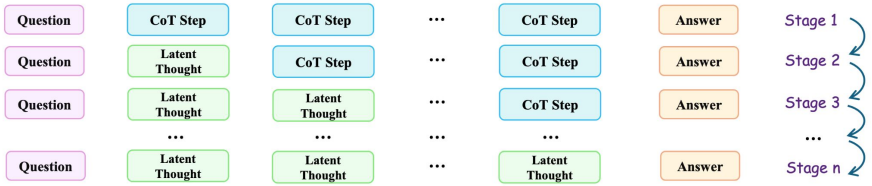
- Sequential Scaling vs. Parallel Scaling
- Underexplored: Budget / Adaptability
- Optimize tokens vs. trajectories.
    - Optimizing Trajectories allows adaptability (e.g. System 1.5)



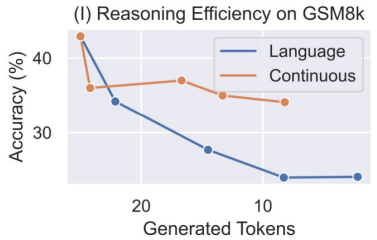
L-SEQ: Coconut

- Curriculum Learning to Compress CoT into continuous vectors.
- Surpassing Explicit CoT on Logical Reasoning with 10-100x faster inference
- Robustness to budget reduction.
- BFS-like Exploration allows improving performance on complex, planning-intensive tasks





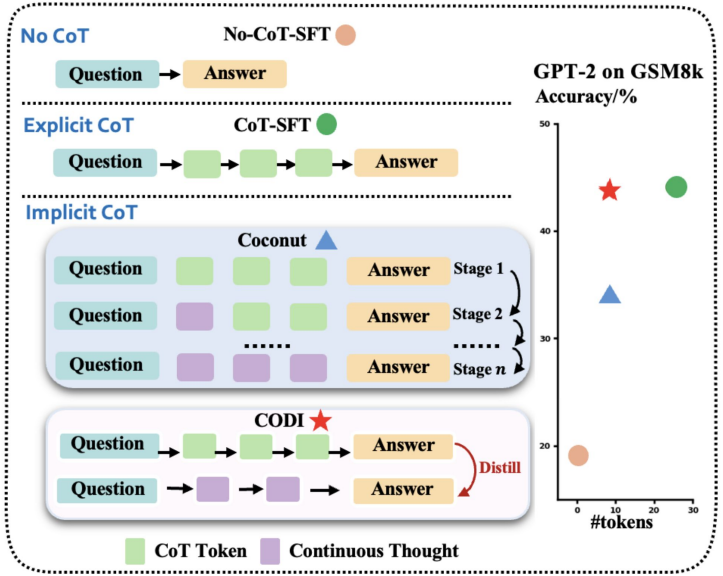
Method	GSM8k		ProntoQA		ProsQA	
	Acc. (%)	# Tokens	Acc. (%)	# Tokens	Acc. (%)	# Tokens
CoT	42.9 ±0.2	25.0	98.8 ±0.8	92.5	77.5 ±1.9	49.4
No-CoT	16.5 ±0.5	2.2	93.8 ±0.7	3.0	76.7 ±1.0	8.2
iCoT	30.0*	2.2	99.8 ±0.3	3.0	98.2 ±0.3	8.2
Pause Token	16.4 ±1.8	2.2	77.7 ±21.0	3.0	75.9 ±0.7	8.2
COCONUT (Ours)	34.1 ±1.5	8.2	99.8 ±0.2	9.0	97.0 ±0.3	14.2

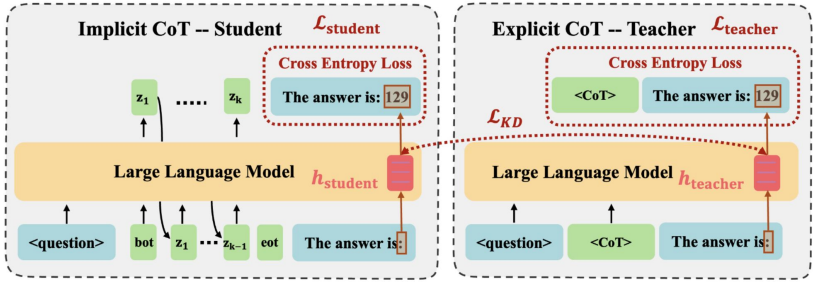


Hao et al., Training Large Language Models to Reason in a Continuous Latent Space (COLM 2025)

L-SEQ: CODI (Hero model)

- Supervision via distillation
- Comparable to Explicit CoT on Math

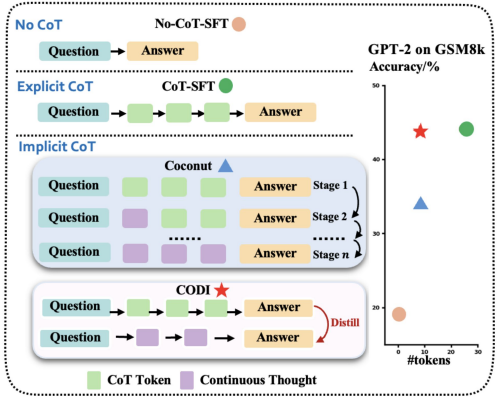


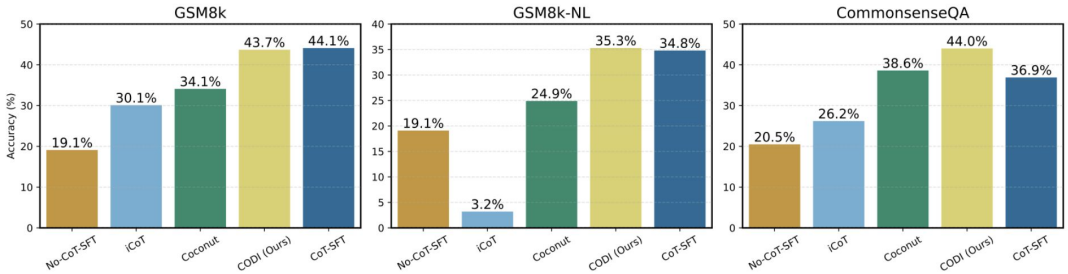


Hao et al., [CODI: Compressing Chain-of-Thought into Continuous Space via Self-Distillation](#) (EMNLP 2025)

L-SEQ: CODI (Hero model)

- Supervision via distillation
- Comparable to Explicit CoT on Math
- OOD Generalization
- However, unstable





In-Domain

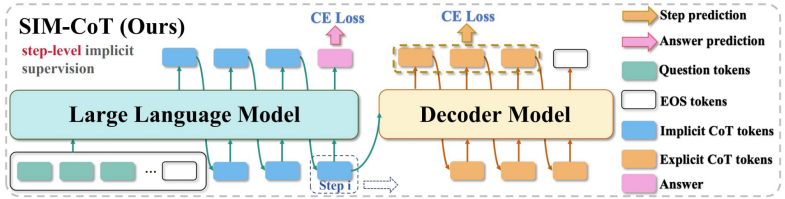
Models	SVAMP	GSM-Hard	MultiA
GPT-2			
No-CoT-SFT	16.4	4.3	41.1
CoT-SFT	41.8	9.8	90.7
iCoT	29.4	5.7	55.5
Coconut	36.4	7.9	82.2
CODI	**42.9**	**9.9**	**92.8**

Math (OOD)

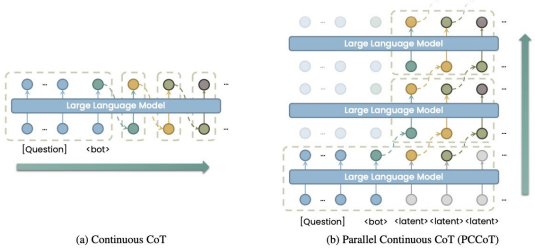
Hao et al., CODI: Compressing Chain-of-Thought into Continuous Space via Self-Distillation (EMNLP 2025)

L-SEQ: Supervision is the Key

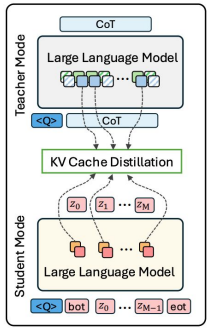
- Variants of CODI - Solve instability with supervision
  - Parallel Continuous CoT (PCCoT): Parallel Scaling
  - KaVA: KV-Cache Compression and Distillation
  - Sim-CoT: Direct Supervision with Thought Decoder
  - CoLaR: Embedding Compression + RL pipeline
  - Chained: PCCoT -> KaVA -> Sim-CoT



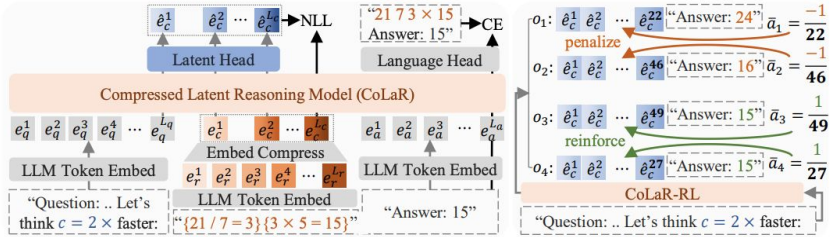
Sim-CoT



PCCoT



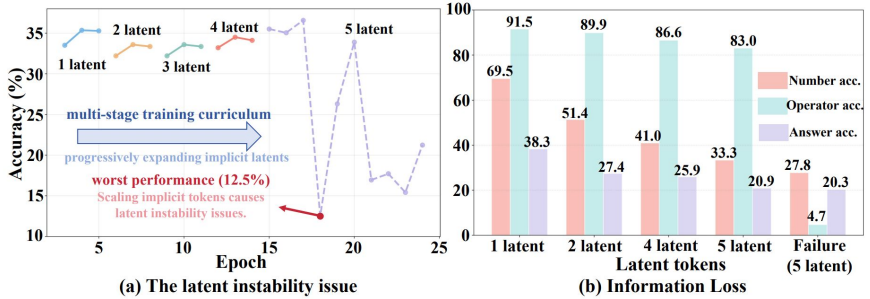
KaVA

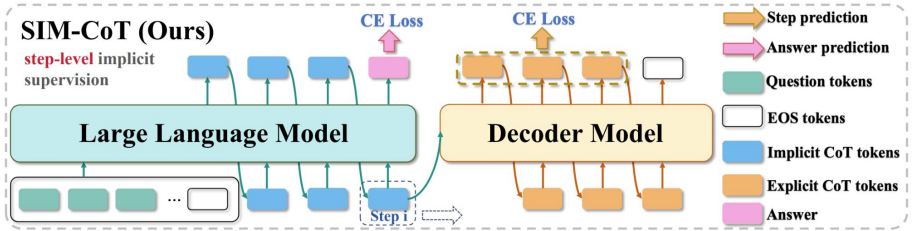


CoLaR

Wu et al., Parallel Continuous Chain-of-Thought with Jacobi Iteration (EMNLP 2025)
Kuzina et al., KaVa: Latent Reasoning via Compressed KV-Cache Distillation (2025)
Wei et al., SIM-CoT: Supervised Implicit Chain-of-Thought (2025)
Tan et al., Think Silently, Think Fast: Dynamic Latent Compression of LLM Reasoning Chains (NeurIPS 2025)

L-SEQ: Supervision is the Key

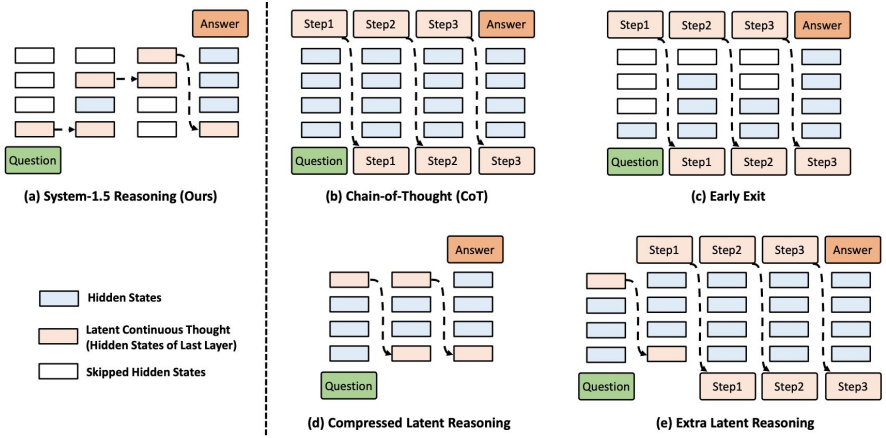




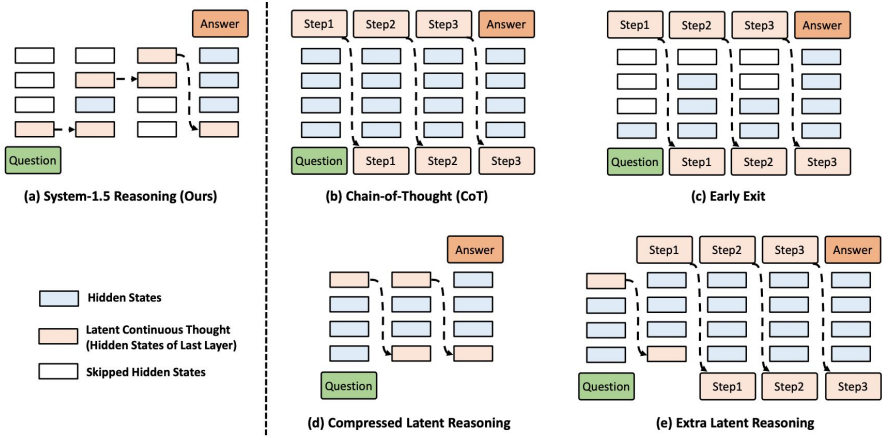
Method	SIM-CoT	In-domain		Out-of-domain				
		GSM8k-Aug		GSM-Hard	MultiArith	SVAMP	Average	# Average
		Acc. (%)	# Tokens	Acc. (%)	Acc. (%)	Acc. (%)	Acc. (%)	Tokens
SFT-CoT	✗	58.4	25.3	13.9	96.7	65.7	58.8	23.1
No-CoT	✗	28.8	1.2	6.3	50.3	26.7	27.8	1.9
iCoT	✗	19.0	1.2	4.4	39.0	40.9	28.1	1.9
Coconut	✗	33.2	13.2	7.0	63.3	43.7	38.0	11.9
	✓	**42.2** (+9.0)	13.2	**9.3**	**87.7**	**43.9**	**47.0** (+9.0)	11.9
CODI	✗	52.7	13.2	11.9	95.0	60.6	55.8	13.4
	✓	**56.1** (+3.4)	13.2	**12.7**	**96.2**	**61.5**	**56.8** (+1.0)	13.4

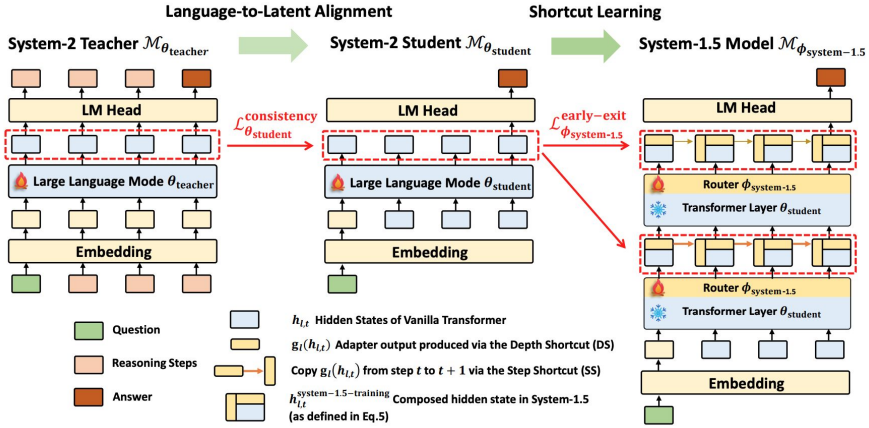
Wei et al., SIM-CoT: Supervised Implicit Chain-of-Thought (2025)

L-SEQ: System 1.5

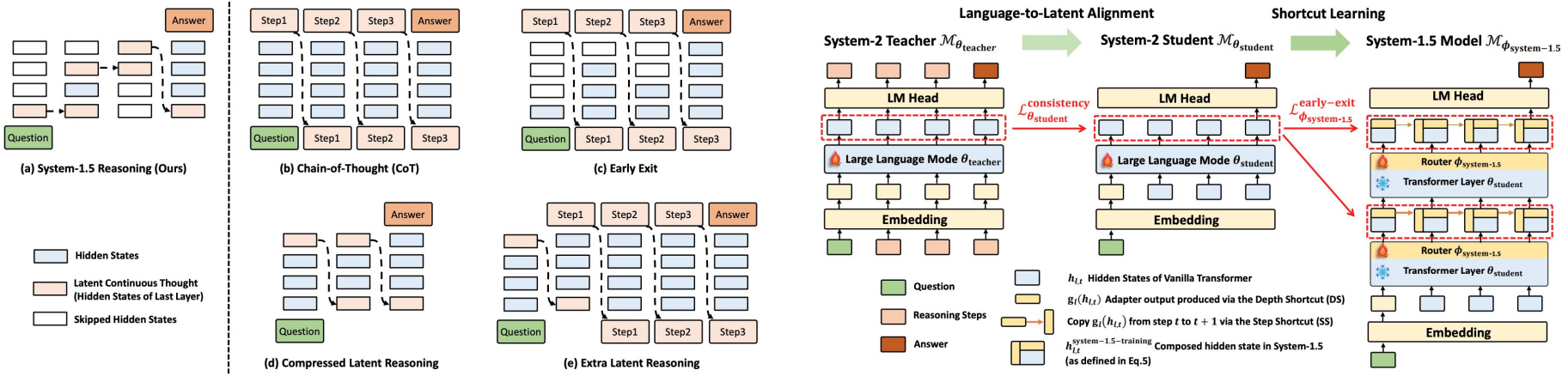


L-SEQ: System 1.5





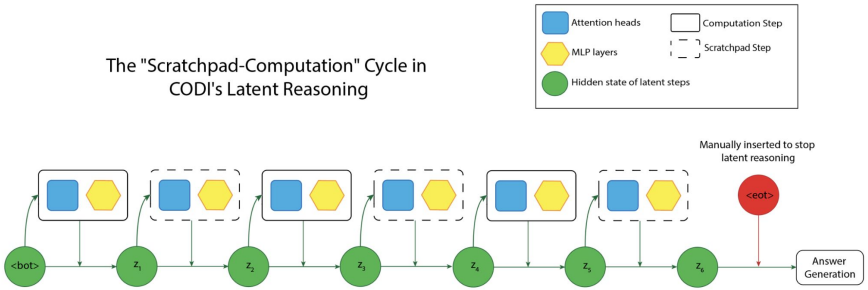
L-SEQ: System 1.5



Method	GSM8K				GSM-HARD				StrategyQA			
	Acc. (%)	# Steps	FLOPs r.	Speedup	Acc. (%)	# Steps	FLOPs r.	Speedup	Acc. (%)	# Steps	FLOPs r.	Speedup
CoT	**46.94**	26	-	-	**38.32**	26	-	-	47.62	52	-	-
LITE	44.51	28	1.84×	1.61×	37.01	28	1.56×	1.44×	46.15	42	1.96×	2.36×
LayerSkip	43.20	32	1.77×	1.45×	36.55	33	1.32×	1.03×	42.54	49	1.8×	1.86×
iCoT	32.14	2	1.02×	13.45×	23.17	4	1.02×	6.17×	34.42	2	1.02×	26.47×
Coconut	36.75	2	1.02×	11.98×	28.25	4	1.02×	7.07×	38.67	2	1.02×	27.25×
CODI	43.78	2	1.02×	13.37×	35.91	4	1.02×	6.93×	45.12	2	1.02×	25.96×
pause token	46.32	38	1.00×	0.8×	38.17	42	1.00×	0.89×	48.14	61	1.00×	0.82×
System-1.5	46.66	2	1.95×	**20.27×**	38.28	4	1.76×	**12.45×**	**48.61**	2	2.12×	**55.65×**

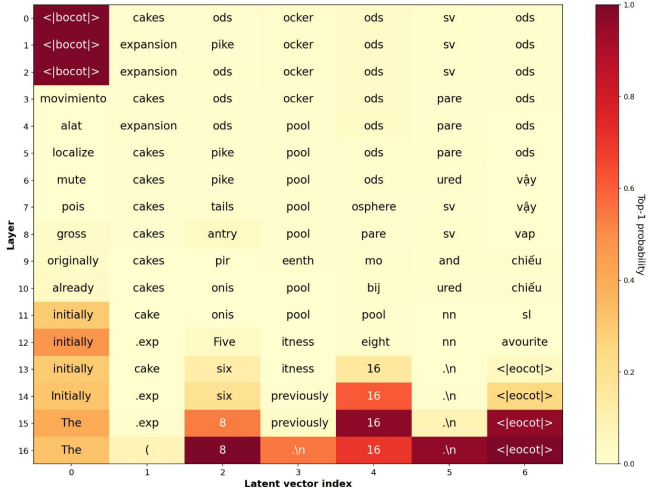
L-SEQ: Scratchpad Thinking Mechanistic Hypothesis

- **Even latent steps (scratchpad)**: hold numerical content, and are where number-related interventions matter most.
- **Odd latent steps (compute)**: where the main operation updates happen.
- **Supporting signals**: attention to numeric tokens and latent tokens peaks on even steps, early decoding, logitlens, patching, SAEs.
- Example (right): Model solves 3-step math problems with 6 latent vectors by storing the 2 intermediate values in specific vectors (3rd & 5th)



GSM8k: "*A team starts with 3 members. They recruit 5 new members. Then each current member recruits 2 additional people. How many people are there now on the team?*", we expect the reasoning should look like:

Step 1: 3 + 5 = 8
Step 2: Step 1 * 2 = 8 * 2 = 16
Step 3: Step 1 + Step 2 = 8 + 16 = 24



Goyal et al., Scratchpad Thinking: Alternation Between Storage and Computation in Latent Reasoning Models (2025)
Cywiński et al., Can we interpret latent reasoning using current mechanistic interpretability tools? (2025)

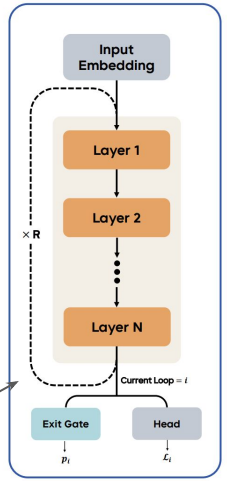
L-LOOP
Latent Looped Reasoning

L-LOOP

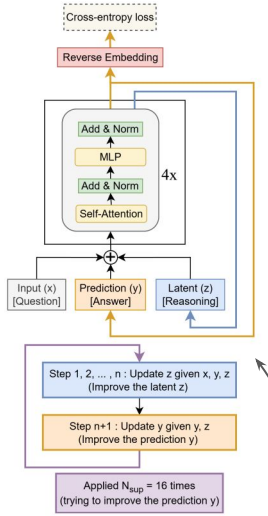
Recursive process that **iteratively refines internal representations**:

Fixed number of “think” **embeddings to refine**

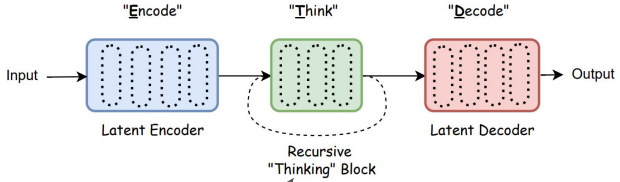
- Encode-Think-Decode (Recursion on specific intermediate latent representations)
- Scaling Latent Reasoning via Looped Language Models (loop over all embeddings)
- Recursion only on few specific representations (decouples answers from reasoning) TRM
- Denoise latent representation with Diffusion/flow matching



Recursion over all embeddings and all blocks



Decouple answer from reasoning



Recursion over specific intermediate block

Encode-Think-Decode

Adapt a pre-trained model (Olmo2) to loop over few middle layers

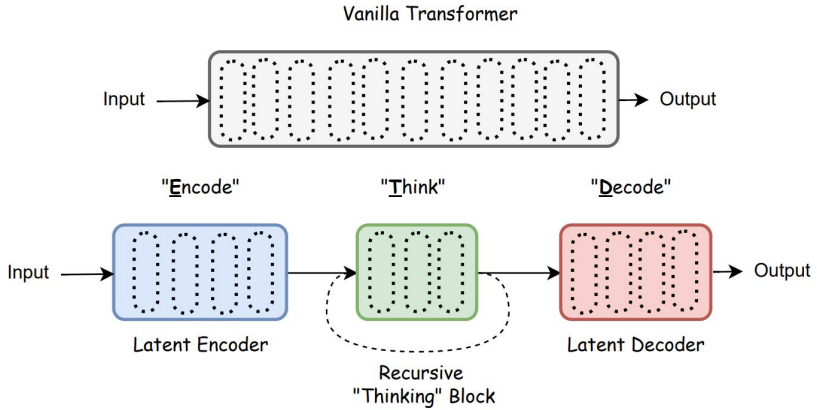
Observations from (Gromov et al 2024): dropping layers with low angular distances

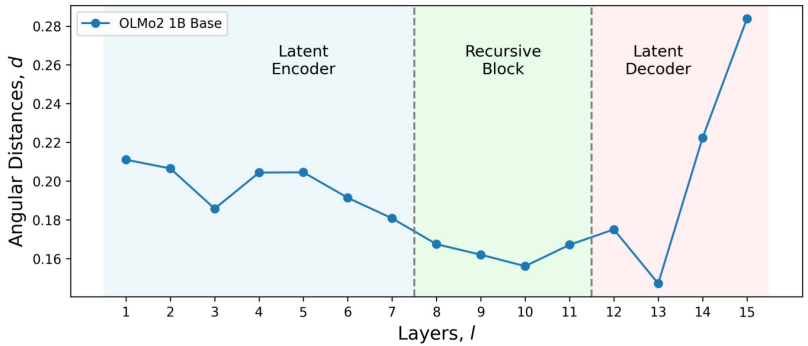
=> **No impact on knowledge retrieval tasks**

=> **High degradation on reasoning tasks**

Experiment:

Second stage training (high quality text), Olmo models with varying number of loop





Koishekenov et al., Encode, Think, Decode: Scaling test-time reasoning with recursive latent thoughts (2025)

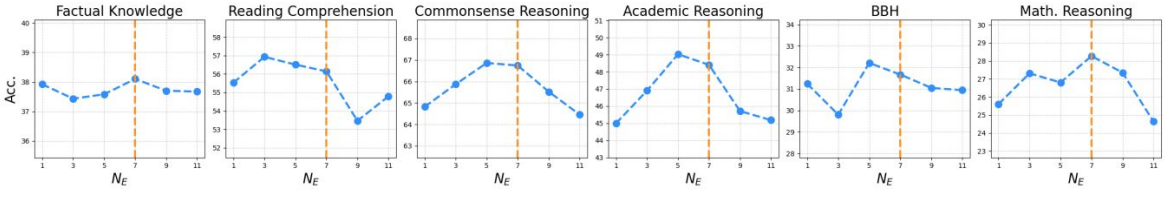
Results

No improvement for Factual Knowledge

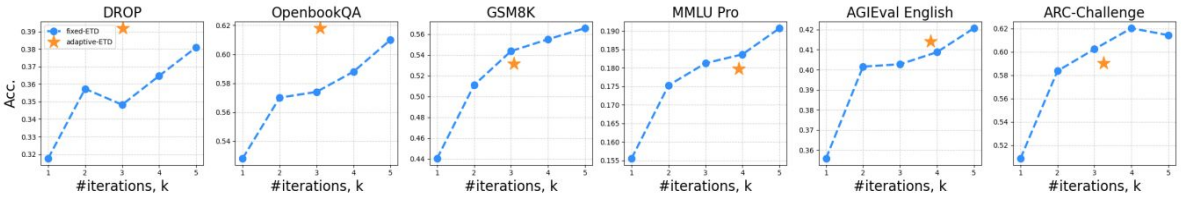
Large improvement for Math and reasoning tasks

		Factual Knowledge		Reading Comprehension		Commonsense Reasoning		Multi-Disciplinary Reasoning		BBH		Math. Reasoning	
Model	Params/FLOPs	Acc.	Δ(%)	Acc.	Δ(%)	Acc.	Δ(%)	Acc.	Δ(%)	Acc.	Δ(%)	Acc.	Δ(%)
OLMo 2 (k=1)	16 / 16	37.55	-	52.19	-	65.29	-	45	-	31.8	-	24.31	-
ETD (k=2)	16 / 20	38.1	(+1.5%)	56.14	(+7.6%)	66.74	(+2.2%)	48.41	(+7.6%)	31.67	(-0.4%)	28.27	(+16.3%)
ETD (k=3)	16 / 24	37.55	(0%)	56.07	(+7.4%)	67.75	(+3.77%)	49.55	(+10.1%)	32.62	(+2.6%)	30.29	(+24.6%)
ETD (k=4)	16 / 28	37.74	(0%)	57.76	(+10.7%)	68.16	(+4.4%)	50.18	(+11.5%)	33.01	(+3.8%)	29.62	(+21.8%)
ETD (k=5)	16 / 32	**38.23**	**(+1.8%)**	**58.5**	**(+12.1%)**	**68.41**	**(+4.8%)**	**50.58**	**(+12.4%)**	**33.49**	**(+5.3%)**	**30.45**	**(+25.3%)**

How does the choice of recursive layers change performance? Fix the recursive "thinking" block size and vary its starting position from layer 2 to 12



Adaptive test-time scaling: Train an exit gate tha predict when to exit the thinking loop



Scaling Latent Reasoning via Looped Language Models

Method:

Pretrain model from scratch to Loop representations multiple times over the model

Stage 1: CE with entropy regularization

$$\mathcal{L} = \underbrace{\sum_{t=1}^{T_{\max}} p_\phi(t \mid x)\, \mathcal{L}^{(t)}}_{\text{expected task loss}} - \underbrace{\beta\, H(p_\phi(\cdot \mid x))}_{\text{entropy regularization}}, \qquad H(p_\phi(\cdot \mid x)) = -\sum_{t=1}^{T_{\max}} p_\phi(t \mid x) \log p_\phi(t \mid x).$$

with Beta controlling how uniform the distribution p(. | x) is

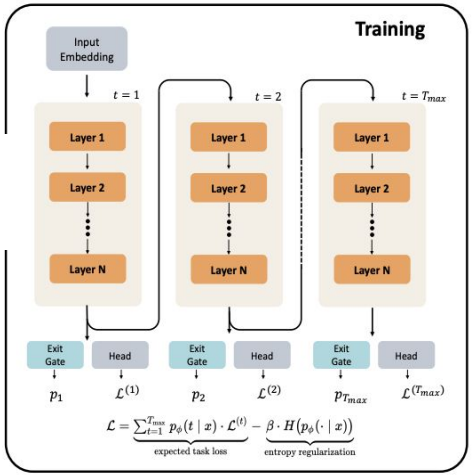
Stage 2: Focused Adaptive Gate Training

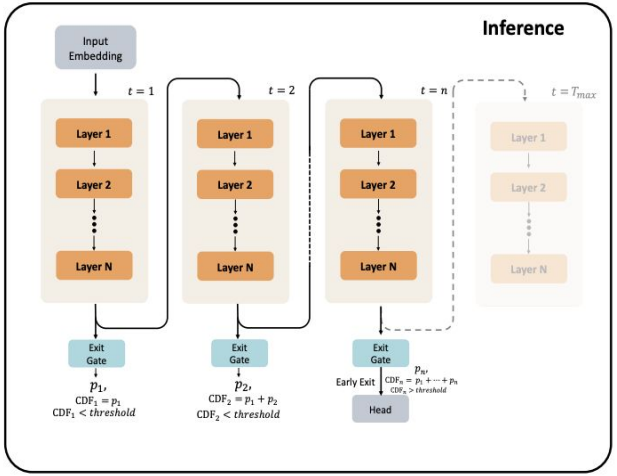
Build label for exit gate to predict correct exit probabilities using:

$$I_i^{(t)} = \max\left(0,\ \mathcal{L}_{i,\text{stop}}^{(t-1)} - \mathcal{L}_{i,\text{stop}}^{(t)}\right)$$

0 = should exit (higher loss in future steps)

1 = continue looping (lower loss in following steps)





Zhu et al., Scaling Latent Reasoning via Looped Language Models (2025)

Scaling Latent Reasoning via Looped Language Models

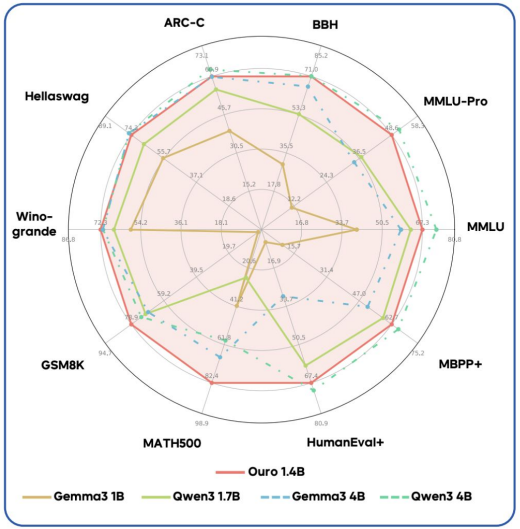
Contributions:

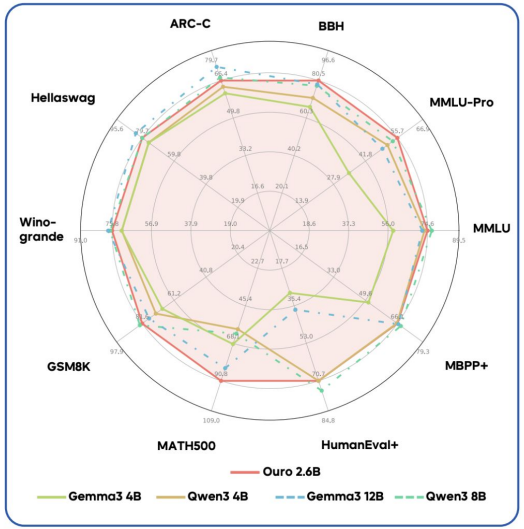
Parameter efficiency for reasoning tasks:
Pre-training on 7.7T tokens, 1.4B and 2.6B parameter LoopLMs match 4B and 8B standard transformers

Entropy-regularized adaptive computation:
Adaptive exits tend to collapse to shallow depths or overuse long loops **=>** added entropy reguarization under a uniform prior over exit steps for unbiased depth exploration

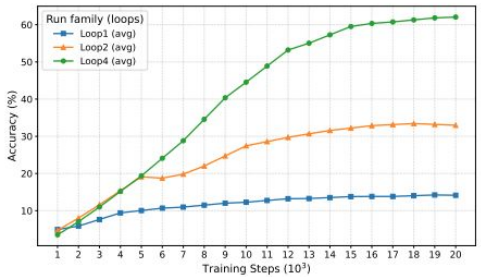
Understanding of recurrence:
Recurrence does not increase raw knowledge storage but enhances knowledge manipulation capabilities on tasks requiring fact composition and multi-hop reasoning.

Improved safety and faithfulness:
LoopLM reduces harmfulness on HEx-PHI





Ouro 1.4B and 2.6B models, both with 4 recurrent steps (red)



MultiHop question answering

	$L = 10$	$L = 16$	$L = 24$
Baseline model			
Base $(12 \otimes 1)$	93.6	94.4	34.8
2 layer model			
Base $(2 \otimes 1)$	21.5	8.4	7.5
Loop $(2 \otimes 6)$	**98.1**	**96.3**	78.0
3 layer model			
Base $(3 \otimes 1)$	75.4	29.8	11.0
Loop $(3 \otimes 4)$	**97.9**	**95.8**	**92.2**
6 layer model			
Base $(6 \otimes 1)$	84.7	59.5	20.0
Loop $(6 \otimes 2)$	93.4	88.5	35.1

Iso parameter study (Mano arithmetic task)

Zhu et al., Scaling Latent Reasoning via Looped Language Models (2025)

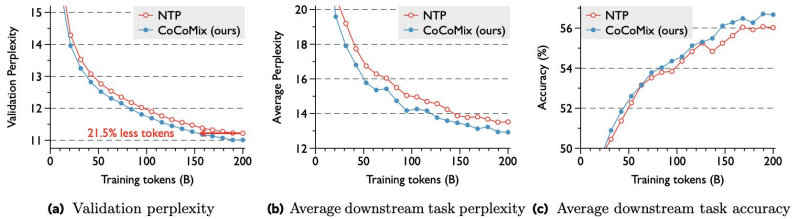
L-LOOP: Main takeaways

- Existing methods loop over a fixed number of embeddings
- Can learn strategies to adapt the number of loop
- Doesn’t improve on knowledge retrieval tasks
- Improves reasoning performance (knowledge manipulation)
- Looping on some layers seems to have more effect on reasoning tasks (routing mechanism)

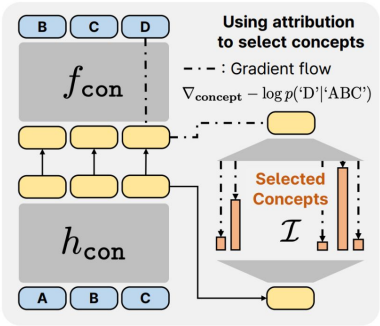
L-OPT
Latent Optimization

L-OPT: CoCoMix

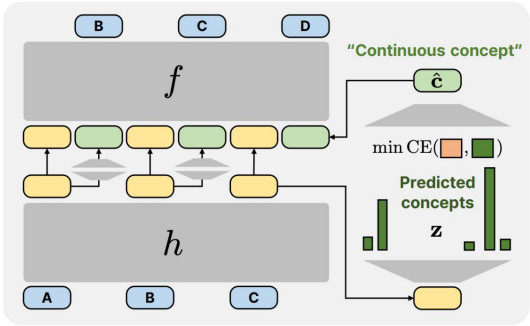
- L-OPT: methods that focus on optimizing hidden representations for reasoning
- Extract concepts from pre-trained model to guide learning.
- Model predicts from compressed continuous representation of predicted concepts
- Symbolic learning, interpretability and steerability



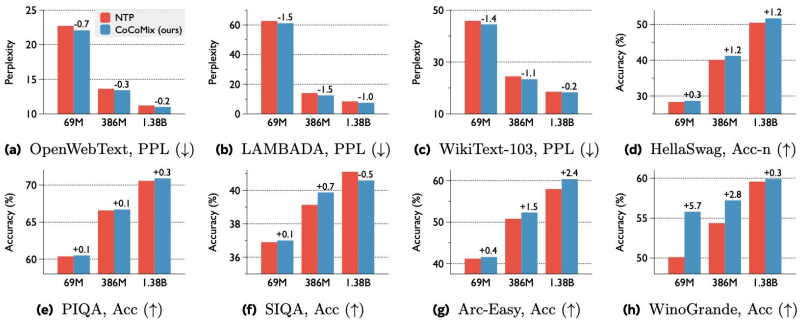
(a) Validation perplexity (b) Average downstream task perplexity (c) Average downstream task accuracy



Extracting concepts from a pretrained SAE model's hidden state



Learning to predict concepts & Mixing/Interleaving continuous concepts into the hidden state



(a) OpenWebText, PPL (↓) (b) LAMBADA, PPL (↓) (c) WikiText-103, PPL (↓) (d) HellaSwag, Acc-n (↑)

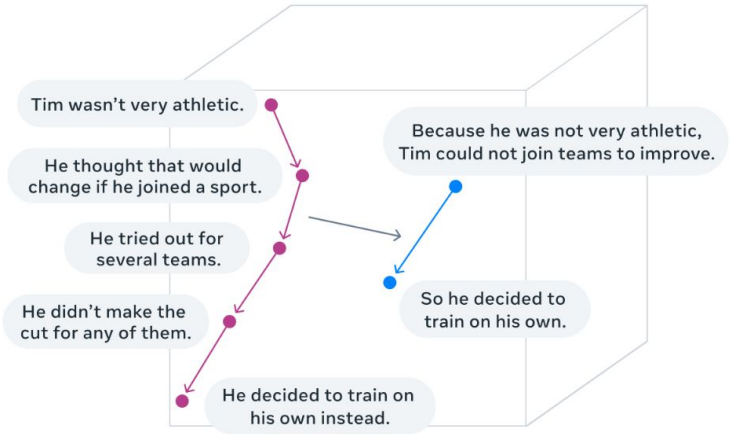
(e) PIQA, Acc (↑) (f) SIQA, Acc (↑) (g) Arc-Easy, Acc (↑) (h) WinoGrande, Acc (↑)

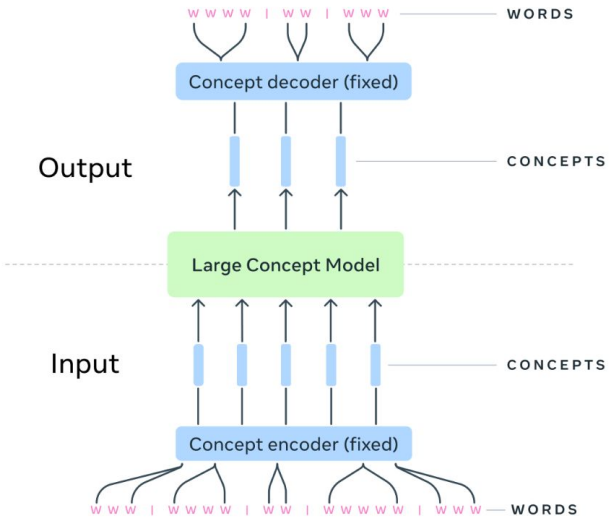
Tack et al., LLM Pretraining with Continuous Concepts (ICLR 2026)

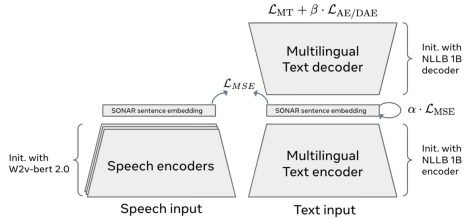
Alternative Architectures

Large Concept Models

- Reasoning in conceptual abstract **language- and modality-agnostic** representation space
- Hierarchical structure
- Handling of long context and long-form output.
- Modularity and extensibility
- Unparalleled zero-shot generalization





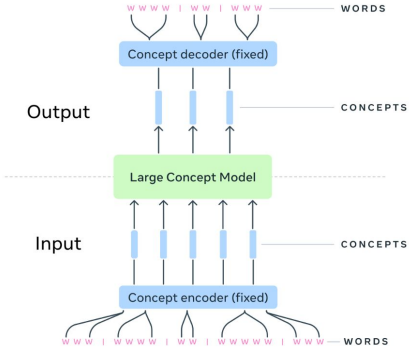


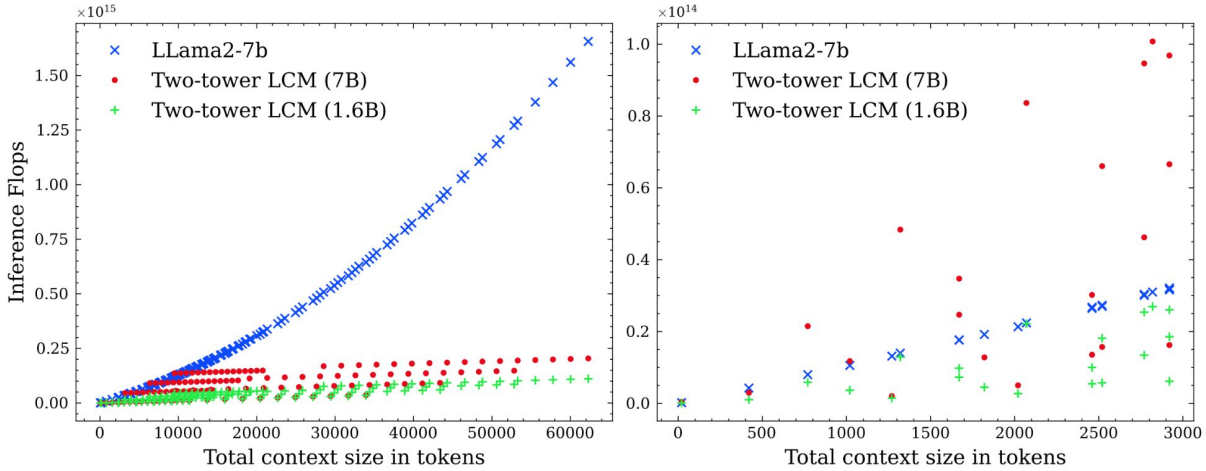
SONAR Encoders

Barrault et al., Large Concept Models: Language Modeling in a Sentence Representation Space (2024)

Large Concept Models

- FLOPs scale **sublinearly** with context
- **Unparalleled zero-shot generalization:** LCM w/ 2-Tower Diffusion beats Gemma 7B and Llama-3.1-8B in zero-shot multilingual tasks (summarization / expansion).

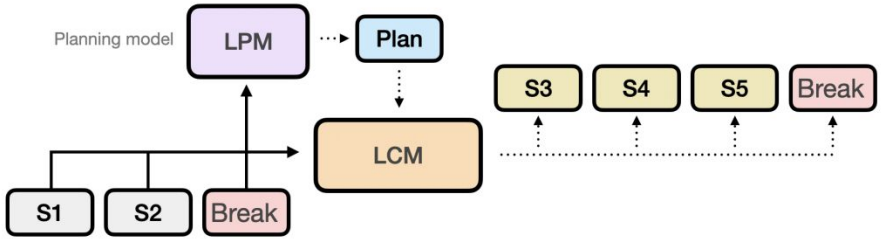




Barrault et al., Large Concept Models: Language Modeling in a Sentence Representation Space (2024)

Large Concept Models

Large Planning Model + Large Concept Model -> LPCM

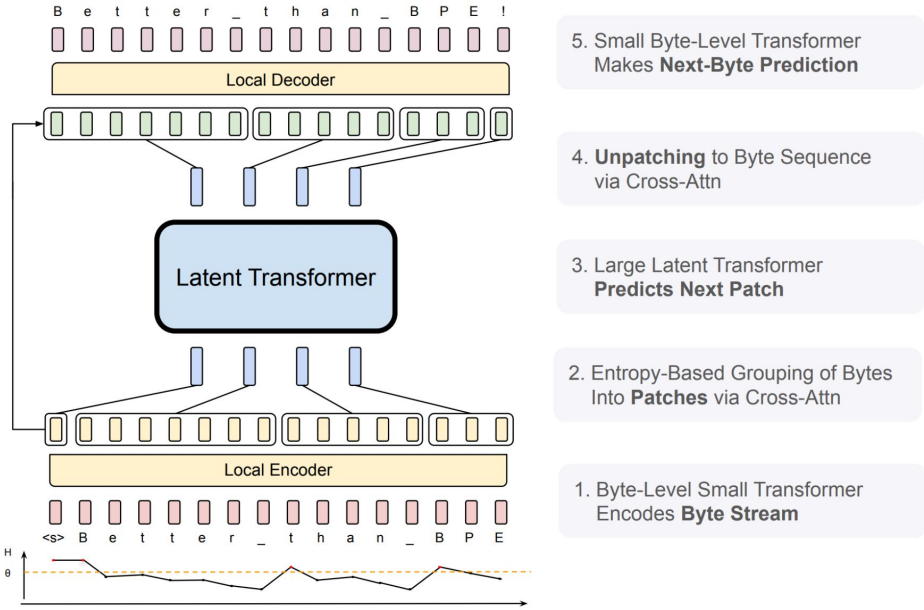


Barrault et al., Large Concept Models: Language Modeling in a Sentence Representation Space (2024)

Large Concept Models

Byte Latent Transformers

- Dynamic compute allocation
- Gains of up to 50% at 8B (vs. Llama)
- Better robustness and semantic generalization
- Improved long-context reasoning

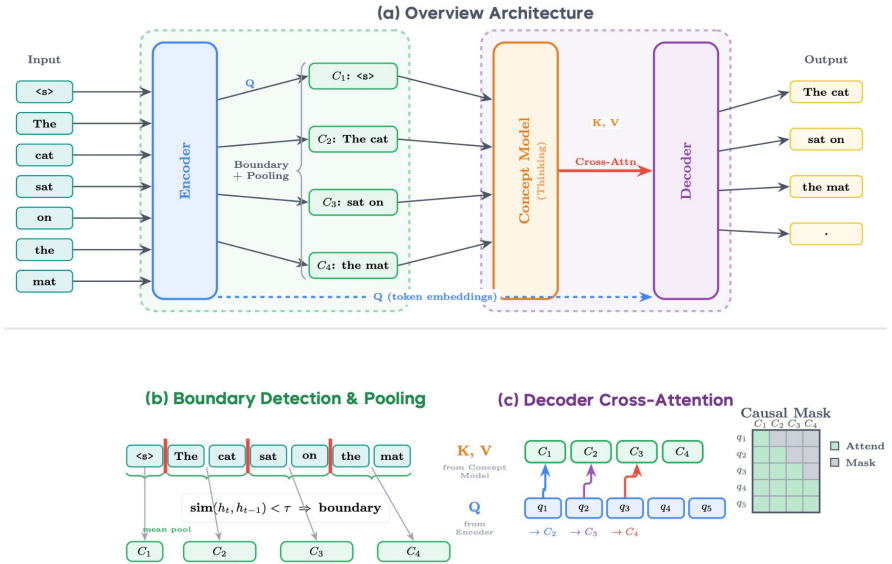


Pagnoni et al., [Byte Latent Transformer: Patches Scale Better Than Tokens](#) (2024)

Large Concept Models

Dynamic Large Concept Models

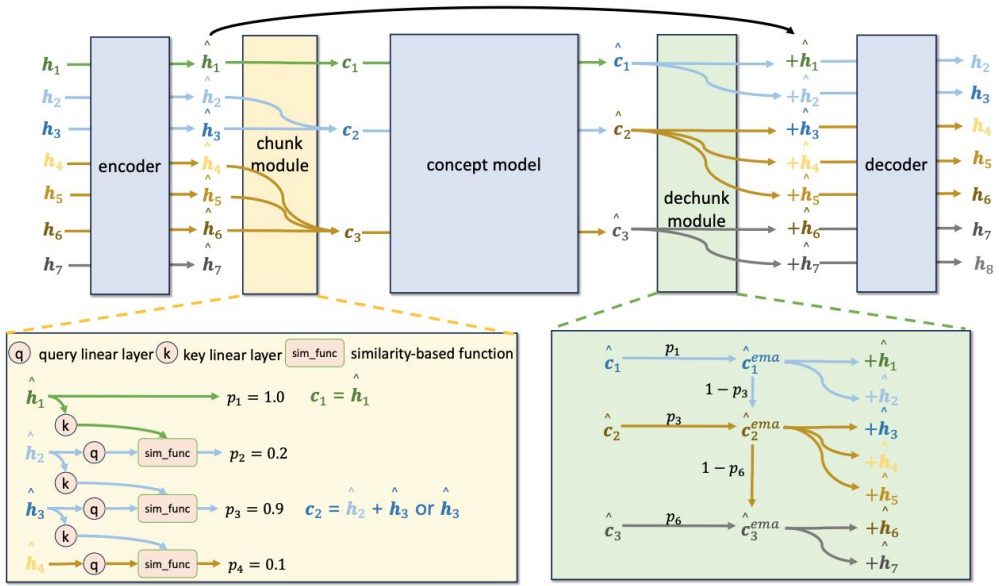
- End-to-End training with learned variable-length concepts
- Back to next-token prediction
- Great improvement from Llama on several tasks.



Qu et al., [Dynamic Large Concept Models: Latent Reasoning in an Adaptive Semantic Space](#) (2026)

Large Concept Models

ConceptMoE



Huang et al., [ConceptMoE: Adaptive Token-to-Concept Compression for Implicit Compute Allocation](#) (2026)

Diffusion Models

LaDiR: Latent Diffusion Enhances LLMs for Text Reasoning

Denoises latent tokens Z in parallel

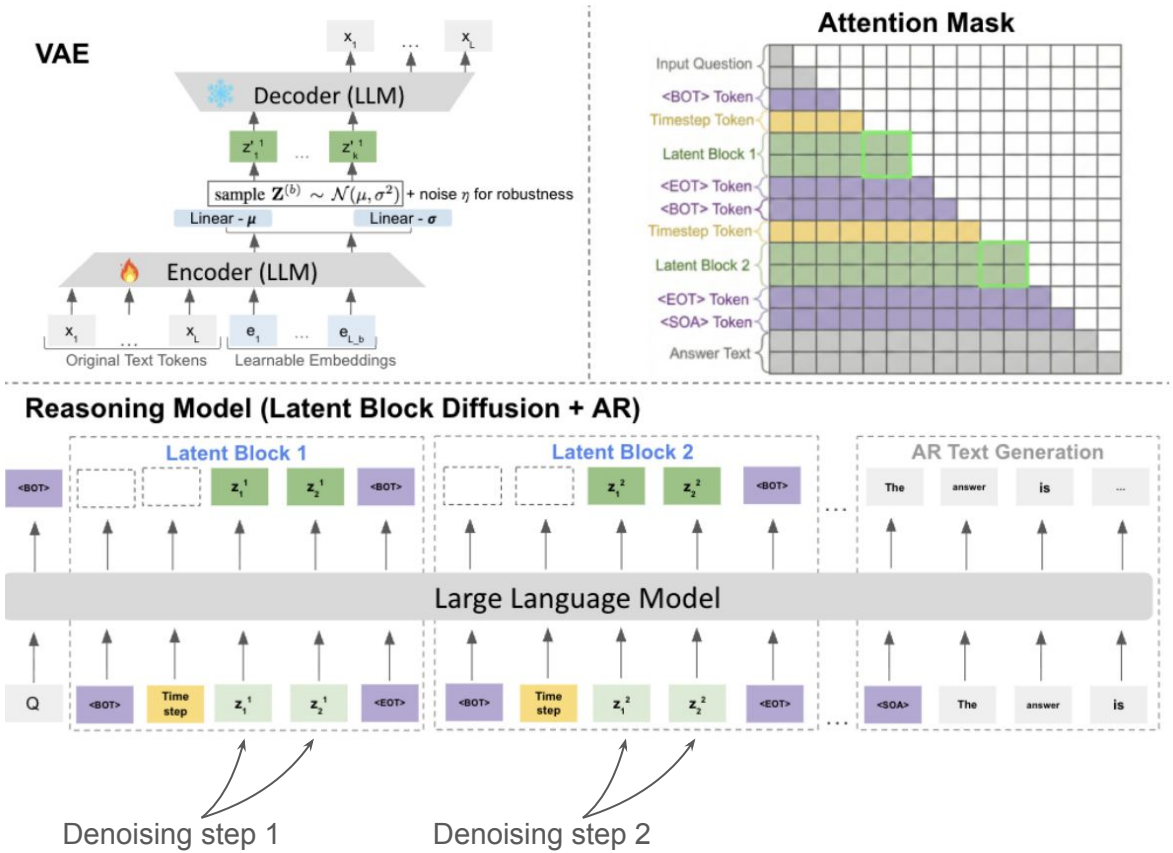
VAE: Trained to predict latent thought

LLM: Trained to diffuse latent thought in parallel

Stage1: Teacher Forcing

$$\mathcal{L} = \lambda_{\text{FM}}\,\mathcal{L}_{\text{FM}} + \lambda_{\text{Ans}}\,\mathcal{L}_{\text{Ans}} + \lambda_{\text{Spec}}\,\mathcal{L}_{\text{Spec}},$$

Stage2: FlowGRPO



Kang et al., LaDiR: Latent Diffusion Enhances LLMs for Text Reasoning (2025)

Results

Method	In-Domain		Out-of-Domain					Avg.
	MATH	GSM8K	Gaokao	DM-Math	College	Olympia	TheoremQA	
Coconut	37.3 / 39.3	68.3 / 74.3	26.8 / 29.3	33.5 / 36.9	40.2 / 42.9	5.8 / 6.3	11.4 / 14.9	31.9 / 34.8
CODI	38.5 / 45.1	76.3 / 81.7	27.5 / 35.2	38.8 / 44.9	43.2 / 49.0	7.6 / 14.8	8.8 / 15.7	34.3 / 40.9
SFT ($\alpha = 1$)	42.1 / 51.0	83.5 / 90.1	29.7 / 39.0	46.6 / 53.1	44.6 / 56.0	9.4 / 13.5	20.3 / 26.0	39.3 / 47.1
LaDiR	**46.2** / **63.7**	**84.8** / **93.7**	**35.4** / **45.8**	**52.3** / **54.2**	**48.6** / **60.3**	**12.9** / **15.3**	**24.7** / **30.7**	**43.5** / **52.0**
–w/o Stage 2	30.7 / 35.8	57.8 / 62.6	24.7 / 30.2	32.0 / 36.9	32.8 / 38.0	5.9 / 10.5	11.9 / 16.9	27.9 / 33.0

Pass@1 (left) / Pass@100 (right)

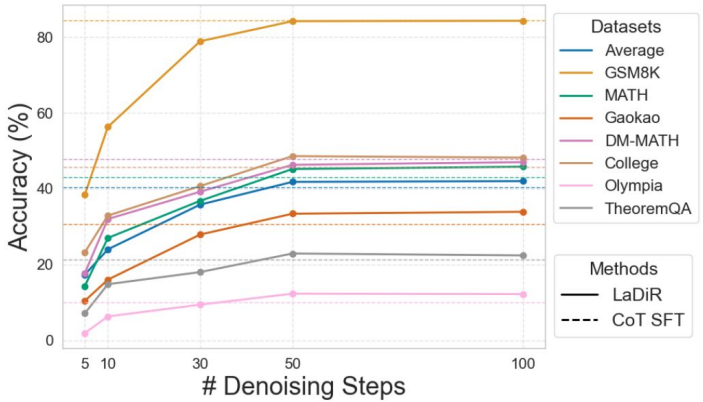
PUZZLE PLANNING – COUNTDOWN : {+, −, ×, ÷} x {a, b, c, d} => e

Model	CD-4 P@1	CD-4 P@100	CD-4 Div.	CD-5 P@1	CD-5 P@100	CD-5 Div.
Dream 7B Base*	16.0	24.7	4.1	4.2	10.3	5.6
MGDM†	**91.5**	95.2	3.2	**46.6**	70.4	4.9
LLaDA 8B SFT	51.2	75.2	5.4	34.4	45.2	6.2
LLaMA 8B SFT	46.7	65.3	3.0	8.9	15.4	3.5
LaDiR	76.6	**96.4**	**7.3**	38.5	**75.2**	**8.9**

Kang et al., LaDiR: Latent Diffusion Enhances LLMs for Text Reasoning (2025)

Results

Effect of number of denoising steps



VAE robustness Augmentation

Gaussian Noise (p=0.3)		**Token Substitution (k=3)**	
k (std)	Acc. (%)	p (prob.)	Acc. (%)
0	68.3	0.0	70.2
1	73.4	0.1	78.3
3	**84.2**	**0.3**	**84.2**
5	79.4	0.5	64.0
–	–	0.7	32.4

Interpretability

Block	**Text**
Question	*Billy sells DVDs. He has 8 customers on Tuesday. The first 3 buy one DVD each, the next 2 buy two DVDs each, the last 3 buy none. How many DVDs did Billy sell?*
Decode($Z^{(1)}$):	Billy's first 3 customers buy one DVD each, so that's $3 * 1 = \langle\langle 3 * 1 = 3\rangle\rangle 3$ DVDs.
Decode($Z^{(2)}$):	His next 2 customers buy 2 DVDs each, so that's $2 \times 2 = 4$ DVDs.
Decode($Z^{(3)}$):	His last 3 customers don't buy any DVDs, so that's 0 DVDs sold.
Decode($Z^{(4)}$):	Therefore, Billy sold a total of $3 + 4 + 0 = 7$ DVDs on Tuesday.
Answer	The answer is: 7.

Kang et al., LaDiR: Latent Diffusion Enhances LLMs for Text Reasoning (2025)

Multimodality: Reasoning with visual inputs

1- Align text and vision representations
2- Train to predict visual representations
3- Distil CoT from VLM teacher (multimodal CODI)

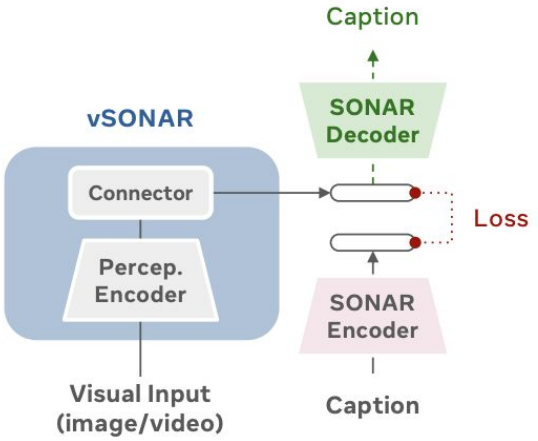
Align text and vision representations

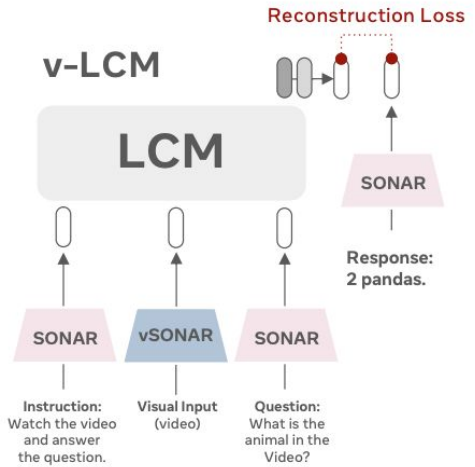
v-LCM

Latent diffusion language model operating directly in the Sonar/vSonar embedding space

Stage 1: Align perceptron (visual encoder) with SONAR (MSE + contrastive loss)

Stage2: Same training as LCM but visual and text representations





Qiu et al., Unified Vision-Language Modeling via Concept Space Alignment (2026)

Results

		Video Captioning / Summarization								M3IT Image				M3IT Video		
		PE-Video		Dream-1k		Vatex		VideoXum		COCO	VIQUAE	VisualMRC	ScienceQA	ActivNetQA	MSRVTT-QA	IVQA
		R-L	BS	R-L	BS	R-L	BS	R-L	BS	R-L	R-L	R-L	Acc.	R-L	R-L	R-L
Qwen2-VL	2B	31.2	**37.3**	18.5	13.9	16.4	30.8	23.6	29.8	24.9	**50.2**	56.1	54.5	53.7	39.6	49.4
	7B	26.9	32.6	19.8	18.1	28.5	**51.6**	26.0	32.4	23.7	49.7	**57.4**	70.4	41.9	22.7	39.1
Qwen2.5-VL	3B	**28.9**	34.4	15.9	8.6	15.0	27.6	26.0	32.9	25.1	48.3	55.7	55.0	52.1	41.6	48.5
	7B	22.2	25.9	15.7	10.5	27.5	50.8	24.1	28.9	18.5	34.5	45.0	61.6	46.0	41.4	54.2
Percep. LM	1B	26.6	31.0	19.3	15.5	19.1	29.6	21.8	33.2	27.5	30.8	45.5	73.6	27.8	14.5	39.1
	3B	26.4	31.3	20.4	19.0	19.3	30.8	**27.0**	**36.4**	34.3	23.7	51.1	89.8	28.0	19.4	26.1
	8B	27.4	31.9	**20.8**	**19.7**	19.0	30.8	26.2	33.7	36.3	31.0	50.0	87.7	40.5	25.3	41.4
LCM	LCM	25.5	27.9	18.5	16.6	23.8	30.8	21.5	22.1	18.0	34.3	33.5	44.7	51.7	36.0	48.9
	v-LCM	27.4	30.0	19.8	19.2	**28.8**	48.7	20.6	25.3	**38.8**	39.4	34.1	76.2	**63.6**	**48.7**	**63.9**

Qiu et al., Unified Vision-Language Modeling via Concept Space Alignment (2026)

Train to predict visual representations

Latent Visual Reasoning(A) and Mull-Tokens: Modality-Agnostic Latent Thinking(B)

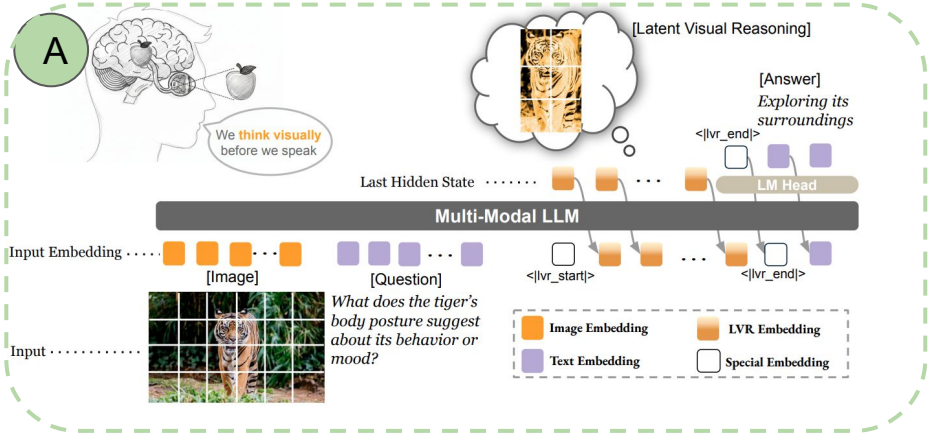
Stage1: Visual supervision of region of interest

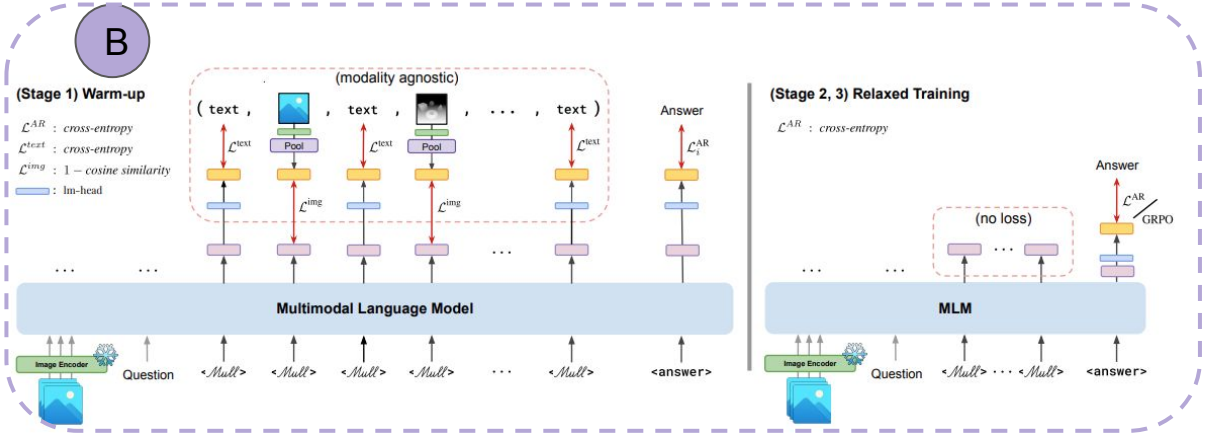
Visual Reconstruction Loss

A $\mathcal{L}_{\text{LVR}} = \frac{1}{T_v} \sum_{t=1}^{T_v} \|\mathbf{h}_t - \mathbf{v}_t\|_2^2.$

B $\mathcal{L}^{img}$: $1 - cosine\ similarity$

Stage2: GRPO

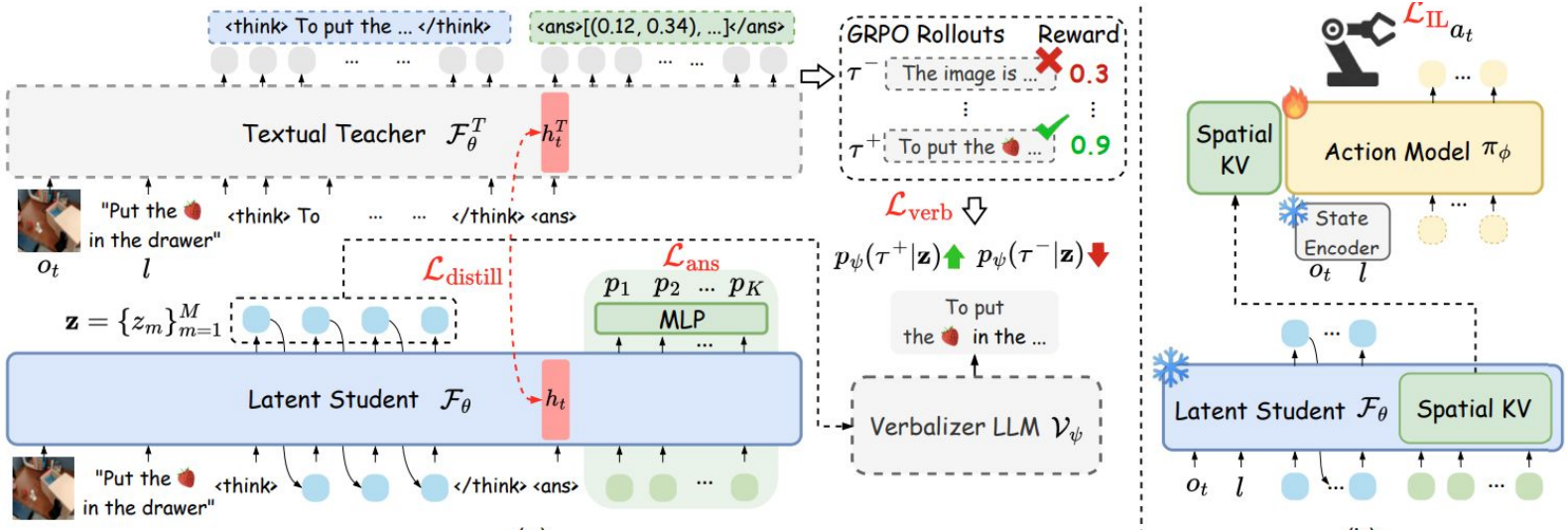




Li et al., Latent Visual Reasoning (2026)
Ray et al., Mull-Tokens: Modality-Agnostic Latent Thinking (2025)

Distil CoT from VLM teacher (multimodal CODI)

Fast-Think-Act: Efficient Vision-Language-Action Reasoning via Verbalizable Latent Planning



1 - Distill CoT hidden states into compact latent tokens z (similar to **CODI**) guided by reward preferences

$$\mathcal{L}_{\text{student}} = \mathcal{L}_{\text{verb}} + \mathcal{L}_{\text{distill}} + \mathcal{L}_{\text{ans}}$$

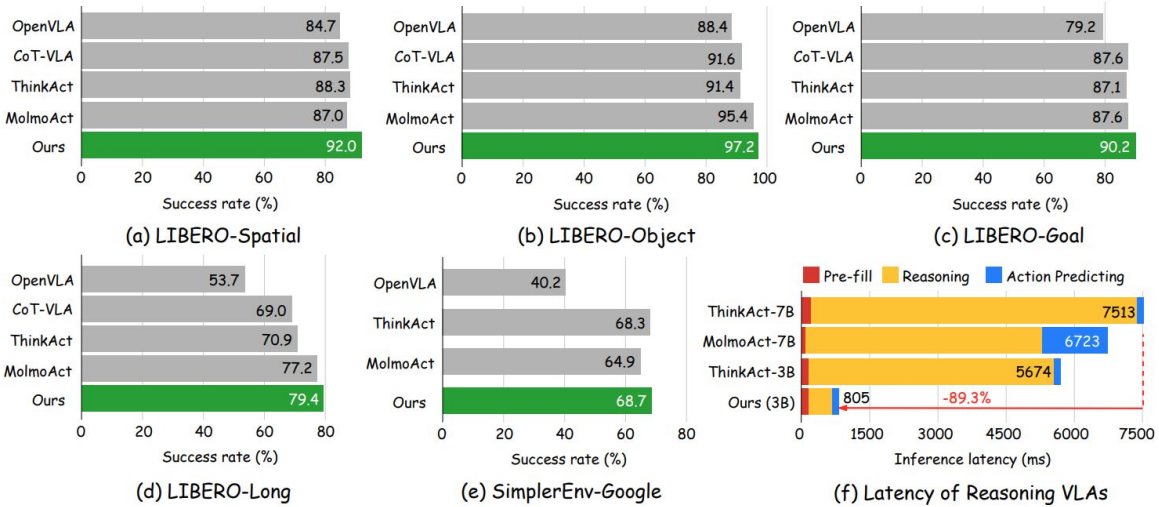
2 - Student answers representations are later used to train an action model

Huang et al., Fast-ThinkAct: Efficient Vision-Language-Action Reasoning via Verbalizable Latent Planning (CVPR 2026)

Results: Evaluation of robot manipulation and reasoning efficiency

Method	EgoPlan-Bench2					RoboVQA					OpenEQA	Overall
	Daily.	Work.	Rec.	Hobbies	Avg.	B-1	B-2	B-3	B-4	B-Avg.	Score	Avg.
GPT-4V Achiam et al. (2023)	36.7	27.7	33.9	32.5	32.6	32.2	26.5	24.7	23.9	26.8	49.6	36.4
Gemini-2.5-Flash Comanici et al. (2025)	44.2	42.3	43.2	39.1	42.4	39.1	31.6	22.9	22.1	28.9	45.3	38.9
InternVL2.5-2B Chen et al. (2024)	30.9	27.8	28.6	33.1	30.1	36.6	33.7	31.0	29.4	32.7	47.1	36.6
InternVL3-2B Zhu et al. (2025)	36.9	29.9	35.6	31.5	33.4	34.4	33.9	33.5	33.3	33.8	48.8	38.7
NVILA-2B Liu et al. (2024)	34.6	26.7	33.3	31.6	31.4	38.7	34.3	31.1	29.2	33.3	47.0	37.2
Qwen2.5-VL-3B Bai et al. (2025)	29.0	27.0	30.2	28.9	28.5	42.5	36.3	28.7	31.8	34.8	43.4	35.6
Magma-8B Yang et al. (2025)	32.1	25.7	34.4	29.3	29.8	38.6	31.5	28.1	26.7	31.2	49.1	36.7
RoboBrain2.0-3B Team et al. (2025)	45.3	37.6	45.9	39.7	41.8	54.4	47.7	43.1	41.0	46.5	50.1	46.1
ThinkAct-3B Huang et al. (2025)	46.6	41.4	45.9	42.5	44.0	62.4	57.3	52.0	49.6	55.3	48.9	49.4
Fast-ThinkAct-3B	**50.3**	**44.3**	**46.4**	**43.2**	**46.4**	**70.1**	**63.0**	**57.2**	**53.0**	**60.8**	**51.2**	**52.8**

Qwen2.5B-VL-3B backbone



Huang et al., Fast-ThinkAct: Efficient Vision-Language-Action Reasoning via Verbalizable Latent Planning (CVPR 2026)

Mechanistic evidences, challenges, limitations

Theoretical Comparison

Previously in CoT Prompting [1]: m steps -> O(m log m) CoT steps

Theorem 1. A **single fixed finite-size Transformer** can be prompted to simulate **any algorithm that runs in m steps** on a Turing machine using $O(m \log m)$ chain-of-thought steps;

L-SEQ [2]: m steps -> m continuous CoT

Theorem 2. A **2-layer Transformer** can solve **directed graph reachability** with D steps of **continuous CoT**, where D is the **graph diameter**.

- Intuitively, each latent step performs one **parallel BFS expansion**.
- D-graph reachability <-> D-hop induction problem

L-LOOP [3]: m steps -> log m + 2 loops

Theorem 3. A **looped 1-layer Transformer** can simulate the effect of a deeper Transformer by reusing the same block across multiple loops, with only modest overhead in width and attention heads.

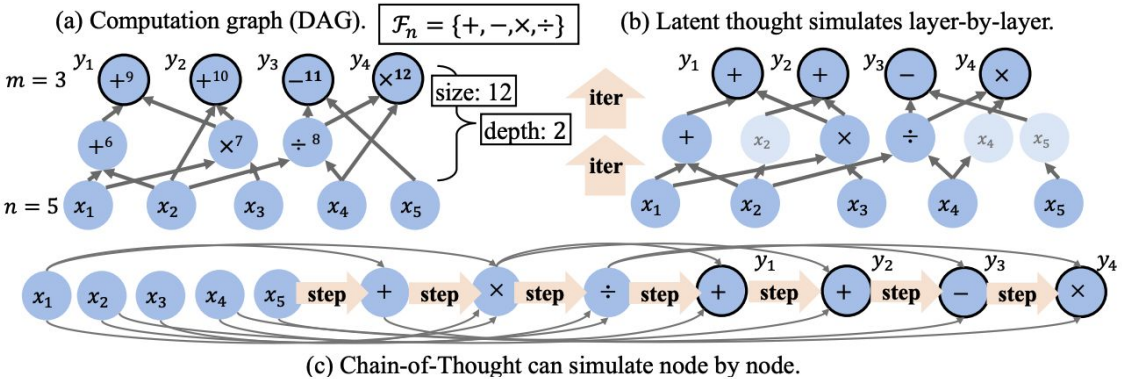
- Consequence: the **p-hop induction problem** can be solved by looping a **1-layer Transformer** only $\lfloor \log_2 p \rfloor + 2$ times.

Theorem 4. A looped Transformer can reproduce the output of an L-layer non-looped Transformer after m CoT steps using m **loops**, with only constant extra architectural overhead.

- non-looped: $(L \text{ layers}) + (m \text{ CoT steps}) \implies$ looped: $(L + O(1) \text{ layers per loop}) \times m \text{ loops}.$
- In general: **1 loop simulates 1 CoT step**.

[1] Qiu et al., Ask, and it shall be given: On the Turing completeness of prompting (ICLR 2025)
[2] Zhu et al., Reasoning by Superposition: A Theoretical Perspective on Chain of Continuous Thought (NeurIPS 2025)
[3] Saunshi et al., Reasoning with Latent Thoughts: On the Power of Looped Transformers (ICLR 2025)

Theoretical Comparison



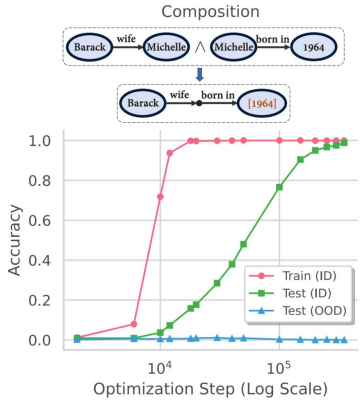
a. A computation graph Gn
b. Latent thought can simulate the computation layer by layer in parallel, using a number of loops equal to the depth of the graph, depth(Gn).
c. CoT can sequentially simulate the computation node by node, using a number of steps proportional to the size of the graph, O(size(Gn)).

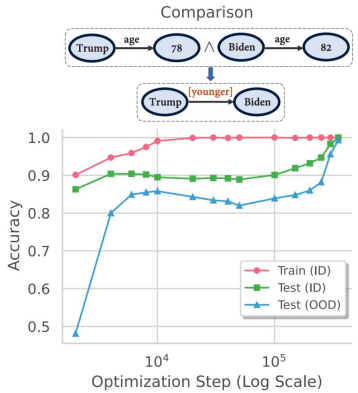
- Latent thought scales with **graph depth** / CoT scales with **graph size**
  - Generalizes the earlier “parallel BFS / reachability” intuition
- Latent thought wins on deterministic parallel computation.
  - CoT can be strictly stronger when stochastic intermediate-token generation matters
    - especially for **approximate counting** and **sampling**

Xu & Sato. A Formal Comparison Between Chain of Thought and Latent Thought (2026)

How does latent multi-step reasoning emerge during training?

- **Training dynamics**
  - **Stage 1 — Memorization**: fit training patterns, weak compositional generalization
  - **Stage 2 — Transition / *grokking***: memorizing behavior is replaced by more generalizing circuits
  - **Stage 3 — Structured reasoning**: reusable intermediate representations support multi-step inference and better OOD generalization, though later hops remain harder than earlier ones
- What makes this emergence easier?
  - More compositional / connected data
  - More training data: needed data grows quickly with hop count (eg. 4-hop x100)
  - Optimization bias toward simple rule-like solutions:
    - e.g. complexity control can favor reasoning over memorized mappings
- Latent reasoning is **learned**, but not guaranteed





Wang et al., Grokked Transformers are Implicit Reasoners: A Mechanistic Journey to the Edge of Generalization (NeurIPS, 2024)
Qu et al., How do Transformers Learn Implicit Reasoning? (NeurIPS, 2025)
Pan et al., Opening the Black Box: A Survey on the Mechanisms of Multi-Step Reasoning in Large Language Models (2026)
Zhang et al., Complexity Control Facilitates Reasoning-Based Compositional Generalization in Transformers (IEEE PAMI, 2025)
Yao et al., Language models can learn implicit multi-hop reasoning, but only if they have lots of training data (EMNLP 2025)

Which tasks?

- Benchmarks? Any
- Tasks in which we have evidence that implicit reasoning can do better (and more efficiently) than Explicit CoT:
  - General Knowledge/Commonsense Reasoning:
    - several works demonstrate great improvements (e.g. CommonsenseQA)
  - Multi-hop Search / Compositional Reasoning (e.g. Strategy QA)
  - Logical / Symbolic Reasoning (e.g. ProverQA, PrOntoQA, ProsQA)
  - Multilingual Comprehension and Reasoning (e.g. LCM on XLSum)
  - Multimodal Reasoning
  - Tasks that require planning
  - LLM-based Agents (still underinvestigated):
    - Support planning, deeper thinking, compositional tasks
    - Compressed Memory
    - KV Cache Sharing across Agents
    - Parallel exploration and execution (e.g. tool-calling)

Comparison Implicit vs. Explicit CoT

Dimension	Explicit Reasoning	Implicit Reasoning
Reasoning Visibility	States verbalized in text, transparent	States hidden in latent space, invisible
Reasoning Efficiency	Verbose, high cost and latency	Compact, faster, resource-efficient
Interpretability	Directly observable and checkable ❌	Indirect, via probing or attribution
Control	Easier to guide, inspect, and correct	No built-in mechanism to steer or verify ❓
Reasoning Diversity	Commits to one trajectory	Encodes multiple alternatives
Supervision Granularity	Explicit, step-aware supervision	Guided by latent objectives
Alignment with Human Thinking	Explains thoughts aloud	Thinking silently

Future directions: Open Problems

- Limited Interpretability and Latent Opacity
  - probing and attribution techniques getting less effective with time
- Limited Control and Reliability
  - Models fail silently
- Performance Gap Compared to Explicit Reasoning in some tasks
  - Latent Reasoning relies on shortcut heuristics (before *grokking*)
  - Hybrid strategies? How to profit from the best of each configuration?
- Lack of Standardized Evaluation
  - difficult to diagnose failure modes or distinguish genuine reasoning from shallow heuristics
  - lack of fair comparability and reproducibility
  - latent annotations, standardized probing protocols, and metrics that assess internal consistency, trajectory depth, and robustness to distributional shifts?

Future directions: Open Problems

- Architecture and Generalization Constraints
  - Most works on small LMs
  - Need for architecture-specific components, such as loop controllers and task-specific latent heads
  - Future work: architecture-agnostic designs and clear reasoning objectives
- Dependence on Explicit Supervision
  - Most methods still need Explicit CoT to guide training
    - restricts independence and scalability
  - Future research: supervision signals operating directly on latent trajectories
    - self-consistency constraints
    - implicit verification objectives
    - unsupervised discovery of latent reasoning structures

Thanks!

Questions?

References

Much of this literature review was inspired by these surveys:

- Li et al., Implicit Reasoning in Large Language Models: A Comprehensive Survey (Sep, 2025)
- Zhu et al. A Survey on Latent Reasoning (July, 2025)
- Chen et al., Reasoning Beyond Language: A Comprehensive Survey on Latent Chain-of-Thought Reasoning (Nov, 2025)
- Pan et al., Opening the Black Box: A Survey on the Mechanisms of Multi-Step Reasoning in Large Language Models (Jan, 2026)