



AAAI
2026

<https://srl4llm.github.io/>

Structured Representation Learning: Interpretability, Robustness, and Transferability for LLMs

20th Jan 14:00 - 18:00
Peridot 205, Singapore EXPO



Hanqi Yan
King's College London

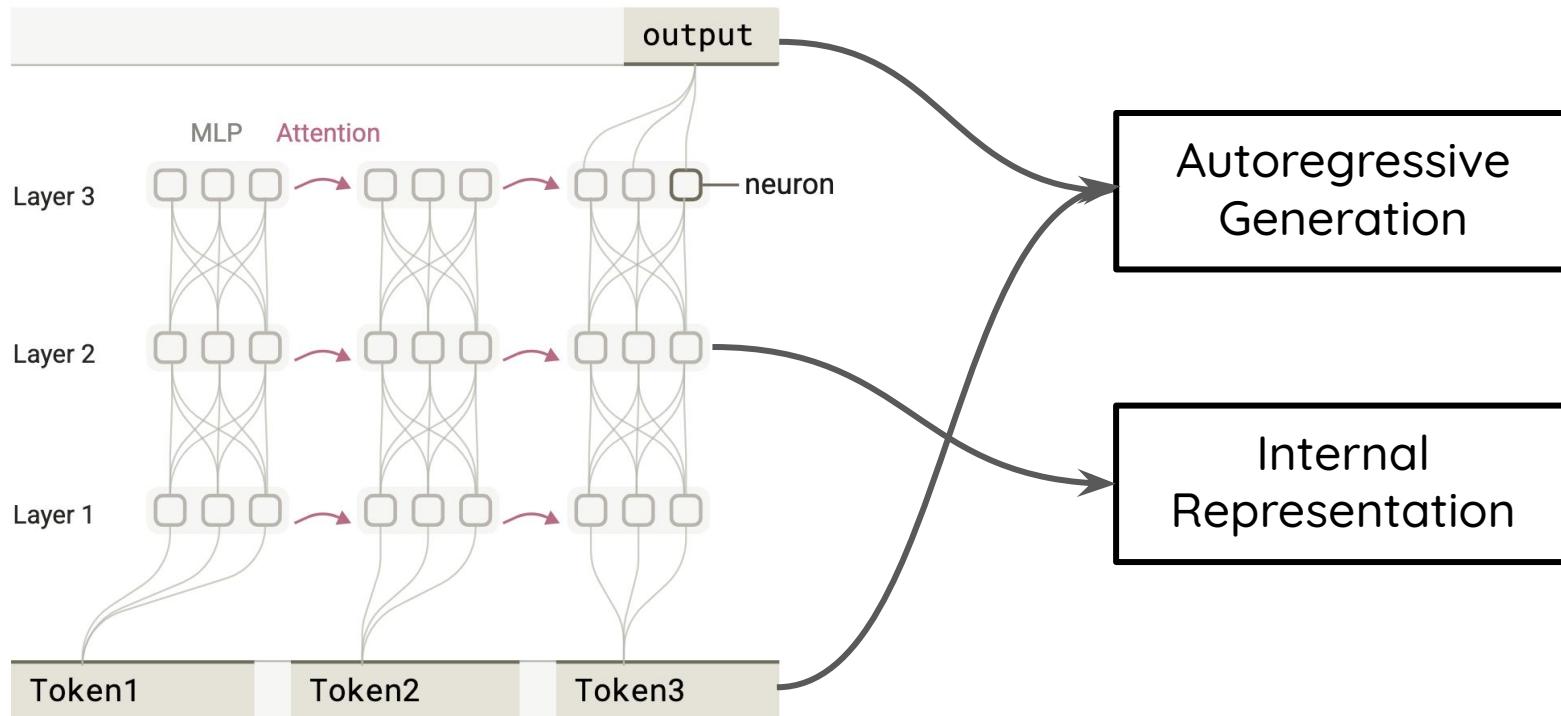


Guangyi Chen
Carnegie Mellon University
MBZUAI



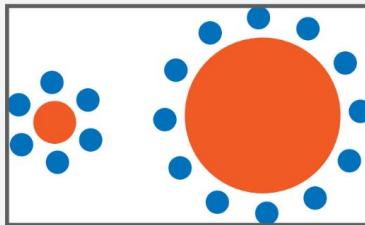
Jonathan Richard Schwarz
Imperial College London
Thomson Reuters

Surface behavior vs internal understanding



Common LLM failures are internal state failures

Hallucination



User

In this image, which orange circle is larger? Think step by step before answering.



OpenAI-01

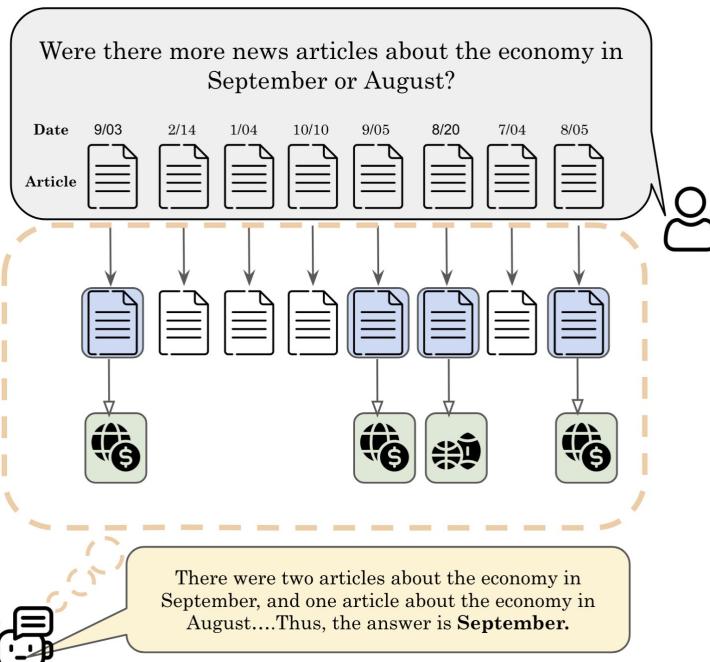
This image is a variant of the Ebbinghaus illusion. Although the circle on the right appears larger due to being surrounded by smaller blue circles, **both orange circles are actually the same size**. The arrangement of the surrounding circles creates a visual context that tricks our perception, making one orange circle look bigger than the other even though they are identical in diameter.



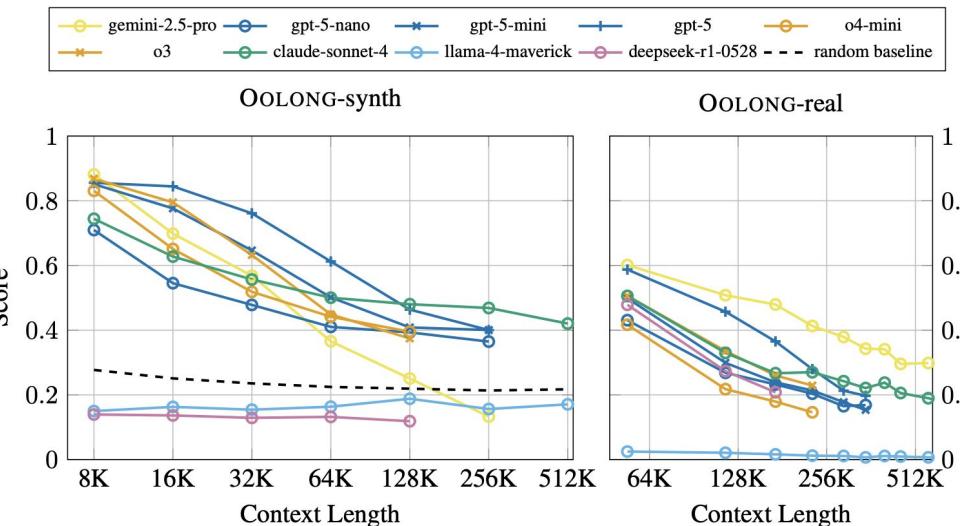
SGP-BENCH (Qiu
et al., ICLR'25)

Common LLM failures are internal state failures

Context rot



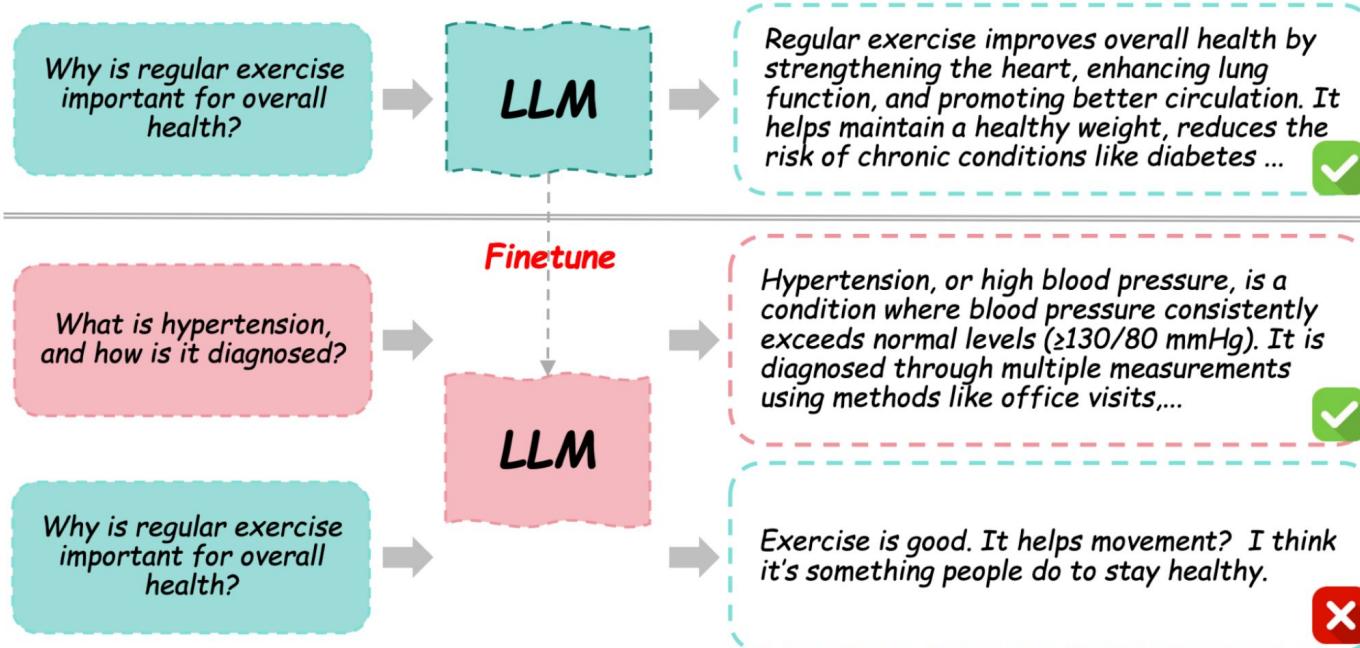
Context didn't disappear – its influence did.



OOLONG (Bertsch et al. 2025)

Common LLM failures are internal state failures

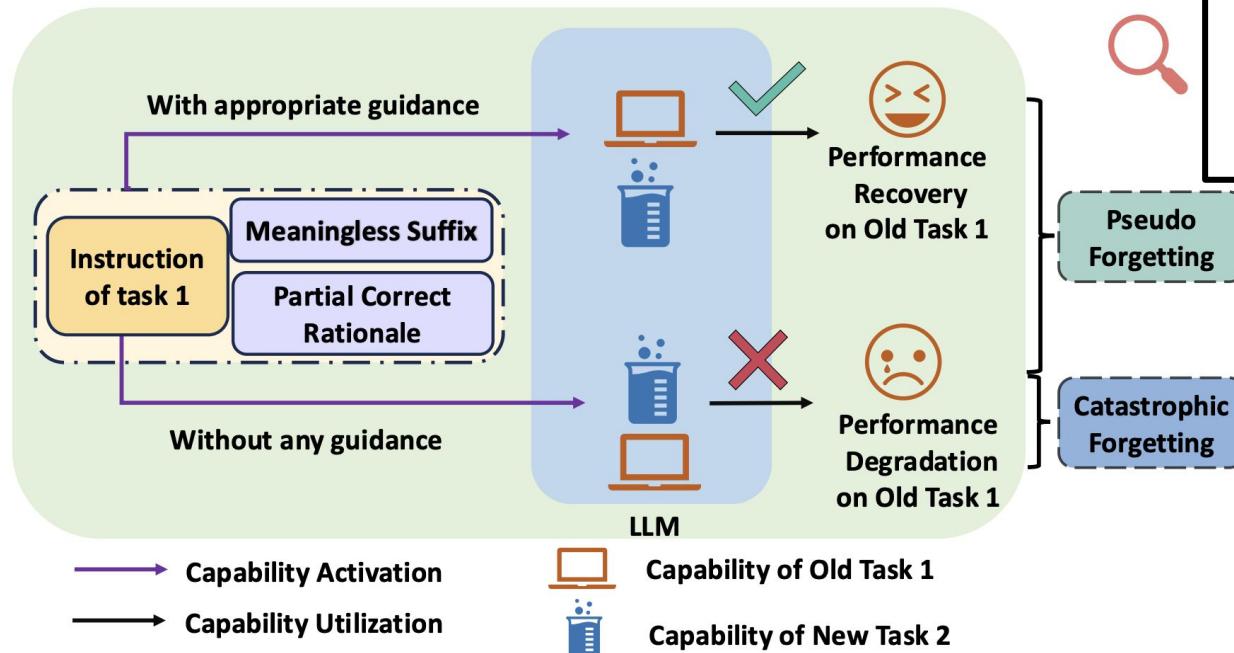
Catastrophic Forgetting



(Song et al., arXiv'25)

Common LLM failures are internal state failures

Catastrophic Forgetting



Forgetting can be restored through appropriate prompts, showing that no actual forgetting occurs.

(Sun et al., ACL'25)

Where things go wrong: latent dominance

The model generates based on what dominates the hidden state.

Hallucination

- The fact is in the context.
- The prior “common sense” is stronger.
- The outputs looks smooth but not based on the fact.

Context rot

- Early context remains in the context window.
- The representation decays and loses dominance.
- Subsequent generation is no longer guided.

Forgetting

- The task knowledge is in the internal representation.
- Fine-tuning reshapes representation space, don’t erase previous knowledge.

What information is represented?

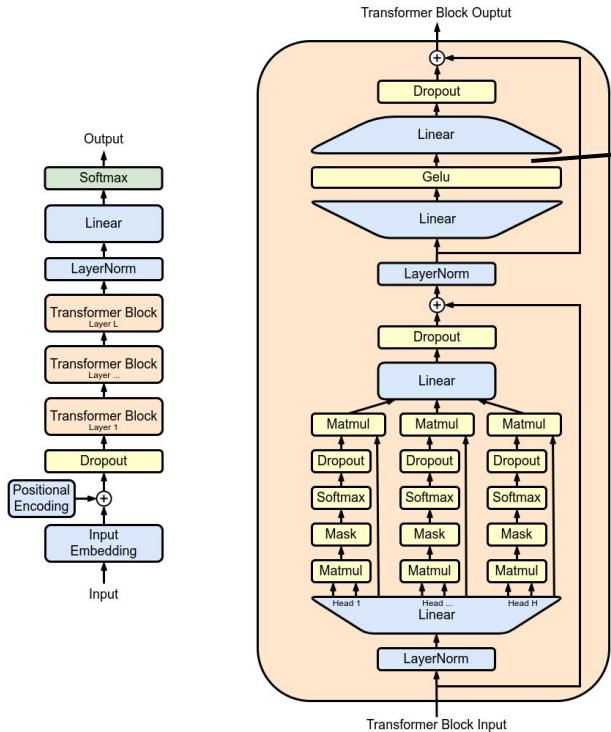
Concepts

Entities

Tasks

Rules

Where and how is it represented?

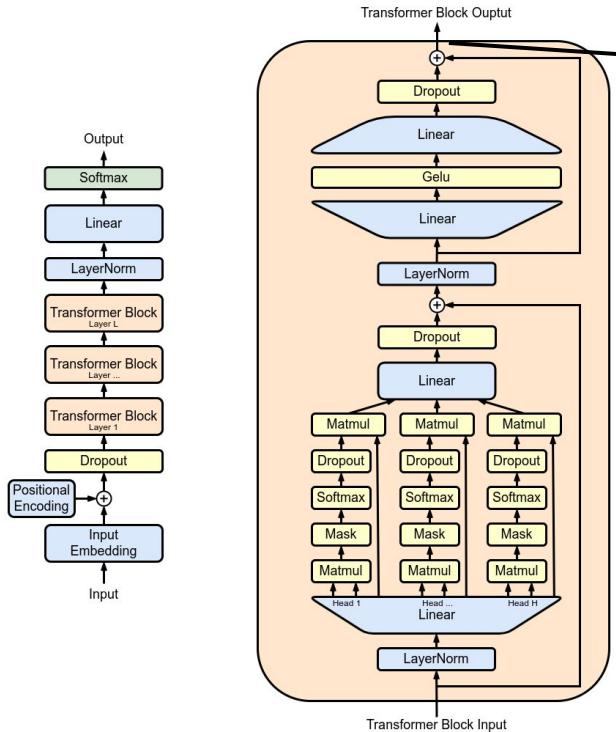


Some methods take the activations in the transformer as the representation for understanding and interpretation.

Geva et al., EMNLP' 22;
Gurnee et al., TMLR' 23;
Wang et al.; KDD' 24;

...

Where and how is it represented?

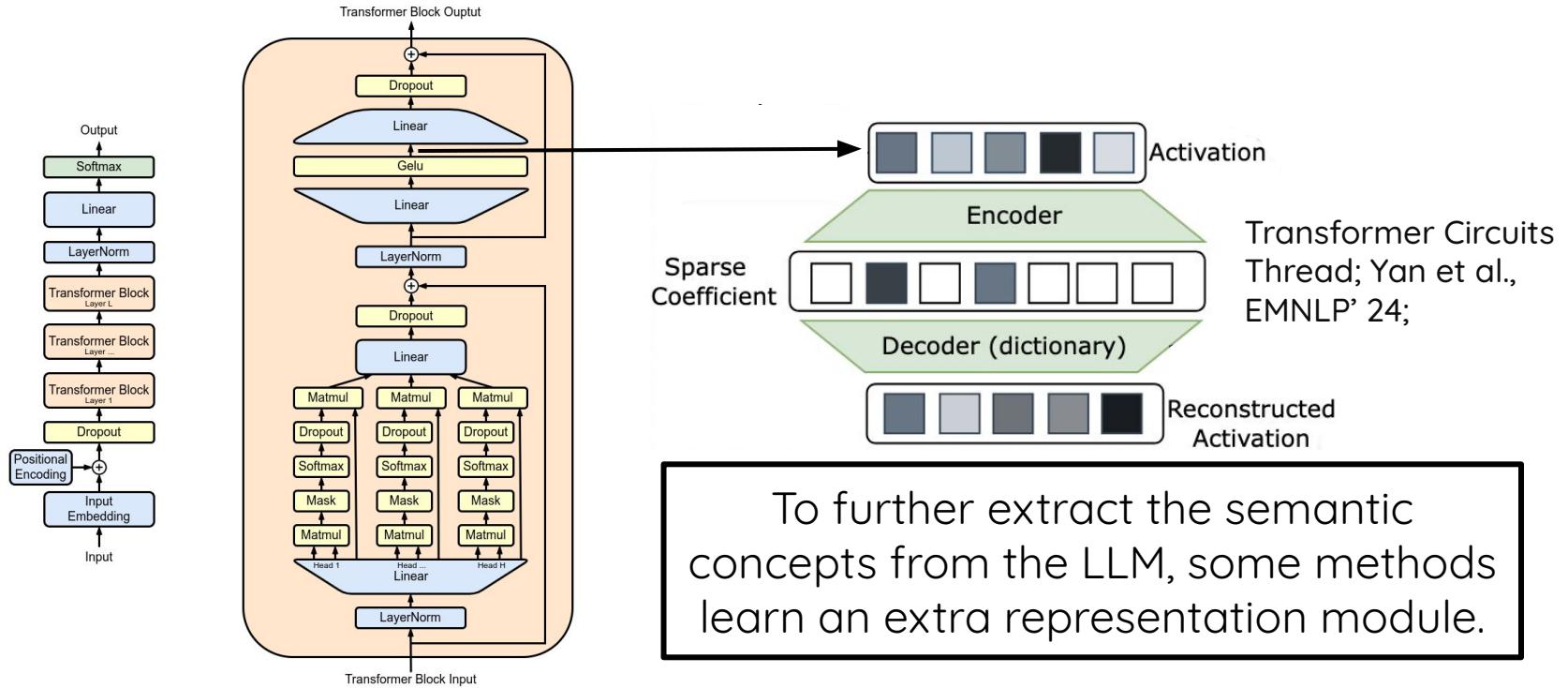


Some methods use the layer outputs as the representations which encodes the context information for predictions.

MEDUSA, Cai et al., ICML' 24;
EAGLE, Li et al., ICML' 24;
COCONUT, Hao et al., COLM' 25

...

Where and how is it represented?



Goals of this tutorial

- ❑ Promote a representation-centric view of LLMs beyond prompt-output behavior.
- ❑ Understand core principles of representation learning and how these principles apply specifically to LLMs.
- ❑ Introduce algorithmic approaches for learning and shaping structured representations in LLMs.
- ❑ Apply representation learning methods to improve: reasoning, editing, interpretability and generalization.

Today's Tutorial Overview

Session 1 Introduction

Session 2 The Principles of Representation Learning

Session 3 Representations for Reasoning

Coffee Break



Session 4 Understand and Model Edit via Representation learning

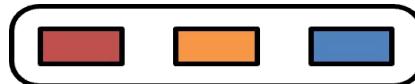
Session 5 Integrate Models Internals for Self-Improvements

Session 6 Conclusion and Future Work

The Principles of Representation Learning

What are good representations

Discriminative



\neq



“Cat”



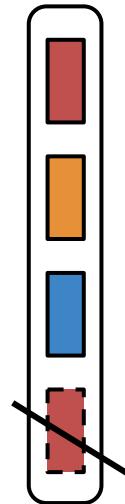
“Dog”



What are good representations

Compact

“There is a
cute cat on
the blanket”

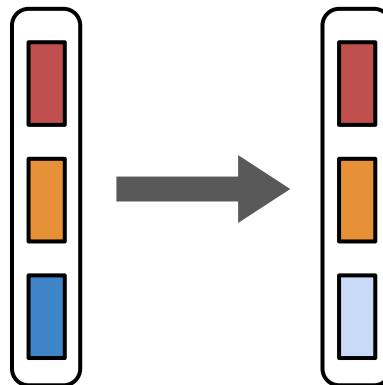


“There is a
cute cat on
the blanket”

What are good representations

Transferable

“There is a
cute cat on
the blanket”

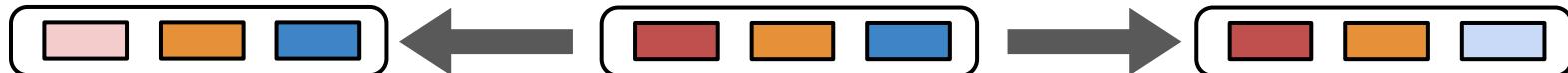


“有一只可爱
的猫在地毯
上”

What are good representations?

Controllable (Disentangled)

Emotion Event Language



“Oh wow! A cat
on the blanket!
So cute!”

“There is a
cute cat on
the blanket.”

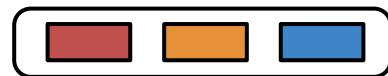
“有一只可爱的猫
在地毯上。”

Identify the latent representation

Data generation process

Emotion Event Language

z



$$x = g(z)$$



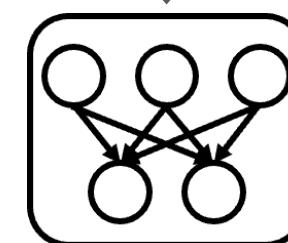
x

“There is a cute
cat on the
blanket.”

Representation learning

“There is a cute cat on the
blanket.”

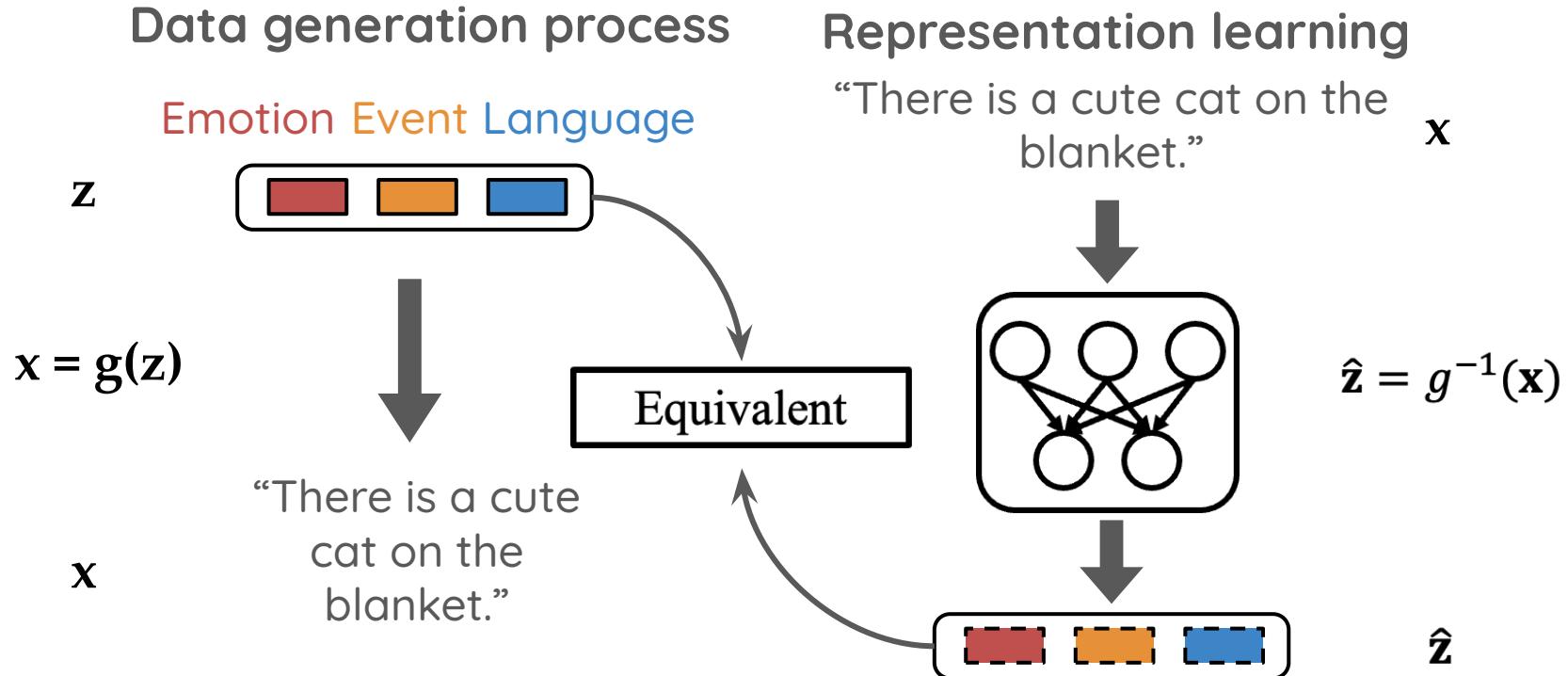
x



$$\hat{z} = g^{-1}(x)$$

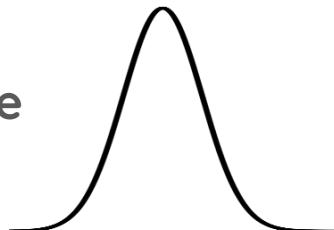
\hat{z}

Identify the latent representation

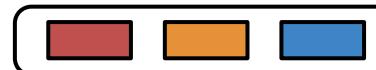


Identifiability

“There is a
cute cat on the
blanket.”



Observed data distribution

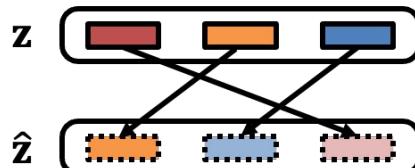


Parameter (representation) space

Definition: A statistical model $\{\mathbb{P}_z \mid z \in \mathbb{Z}\}$ is identifiable if

$$\forall z, \hat{z} \in \mathbb{Z}, \quad \mathbb{P}_z = \mathbb{P}_{\hat{z}} \Rightarrow z \sim \hat{z}$$

**Component-wise
Identifiability**



$$\hat{z}_i = g^{-1} \circ g_i(z_i)$$

$$[\begin{array}{ccc} 1 & 2 & 3 \end{array}]$$

$$[f_a(2) f_b(3) f_c(1)]$$

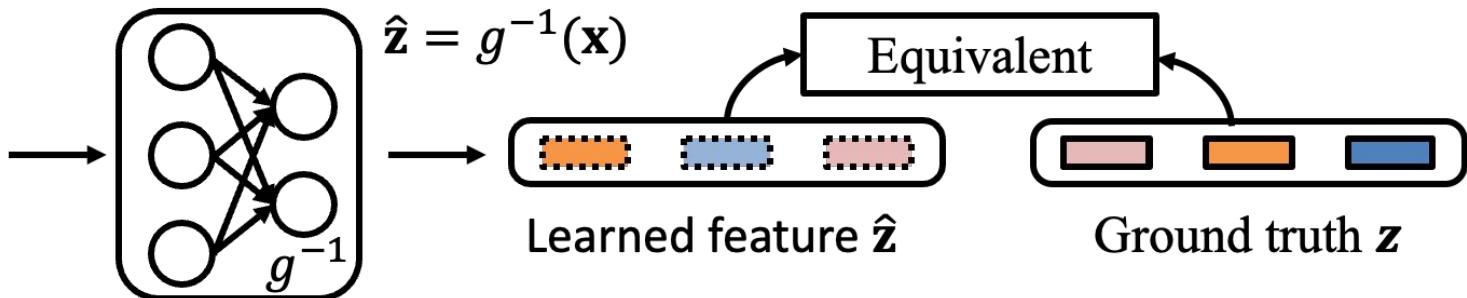
The overall process

$$\mathbf{z} = f_{\mathbf{u}}(\epsilon), \quad \mathbf{x} = g(\mathbf{z})$$



Causal Representation Learning

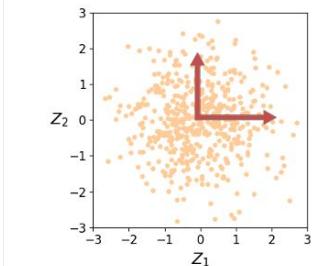
“There is a
cute cat on
the
blanket.”



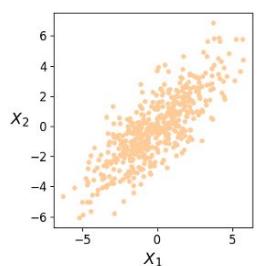
Intuition: a simple linear case $\mathbf{X}=\mathbf{A}\mathbf{Z}$

Linear Gaussian

Latent variable (\mathbf{z})



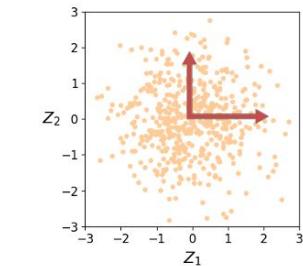
Observation (\mathbf{x})



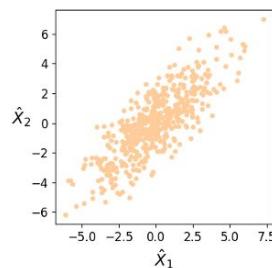
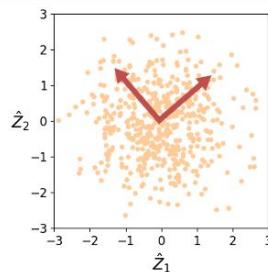
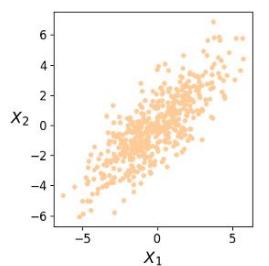
Intuition: a simple linear case $X=AZ$

Linear Gaussian

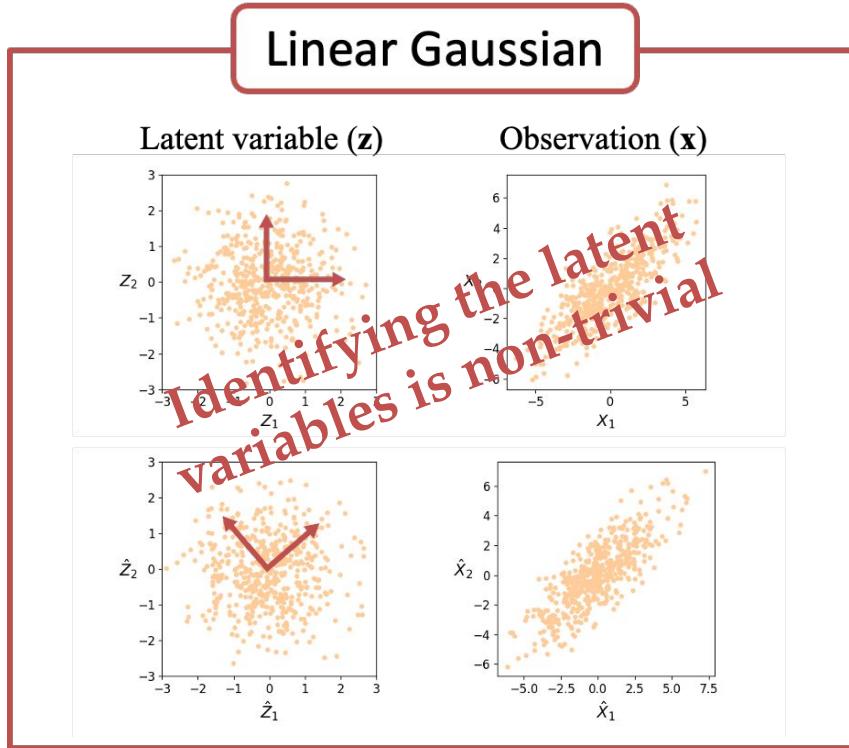
Latent variable (\mathbf{z})



Observation (\mathbf{x})



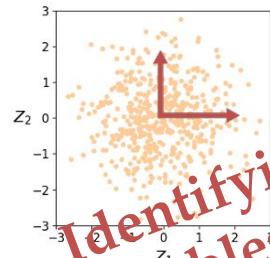
Intuition: a simple linear case $X=AZ$



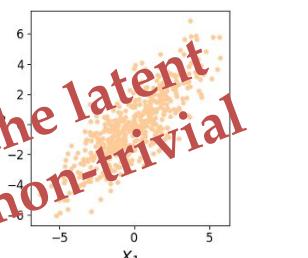
Intuition: a simple linear case $X=AZ$

Linear Gaussian

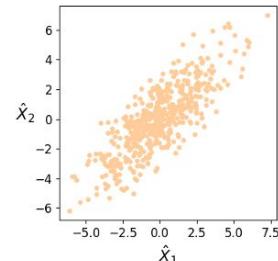
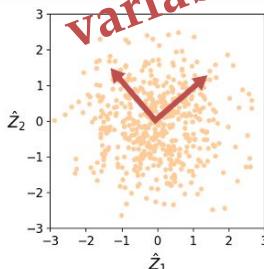
Latent variable (\mathbf{z})



Observation (\mathbf{x})

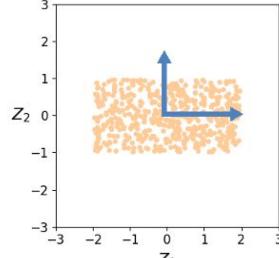


Identifying the latent variables is non-trivial

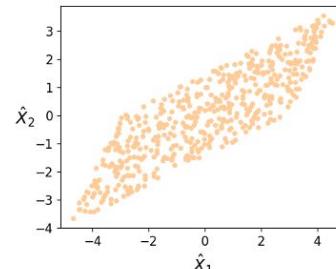
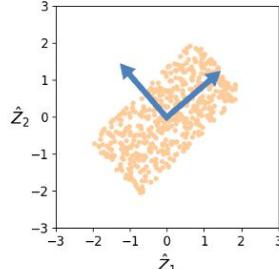
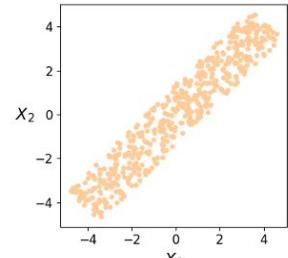


Linear Non-Gaussian

Latent variable (\mathbf{z})



Observation (\mathbf{x})



What principles can we use to learn representations?

★ **Sufficient Change Principle**

- ★ The Sparsity Principle
- ★ Learning Framework
- ★ Application Showcase

Sufficient change principle

$$\mathbf{z} = f_{\mathbf{u}}(\epsilon), \quad \mathbf{x} = g(\mathbf{z})$$



Identification Condition

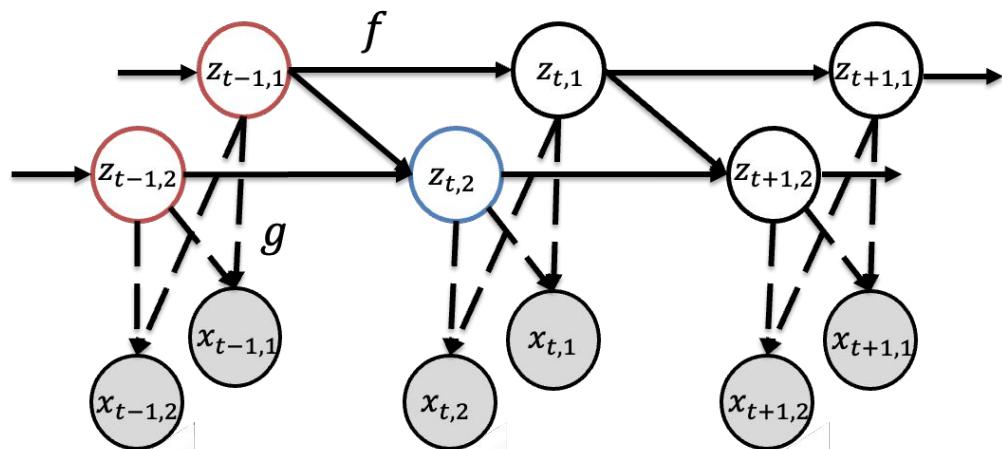
If the data is generated as described above and meets the following criteria:

- **[Sufficient change]:** There are enough values of \mathbf{u} to observe distributional changes, and the changes should be sufficiently large.
- **[Invertible function and smooth density]:** g is invertible, and $p_{\mathbf{z}|\mathbf{u}}$ is smooth.
- **[Conditional independence]:** $\log p_{\mathbf{z}|\mathbf{u}}(\mathbf{z}|\mathbf{u}) = \sum_i \log p_{z_i|\mathbf{u}}(z_i|\mathbf{u})$
- **[Marginal distribution matching]:** $p_{\mathbf{x}|\mathbf{u}} = p_{\hat{\mathbf{x}}|\mathbf{u}}$

Then, the learned latent variables are component-wise identifiable.

Temporal dynamics provide changes

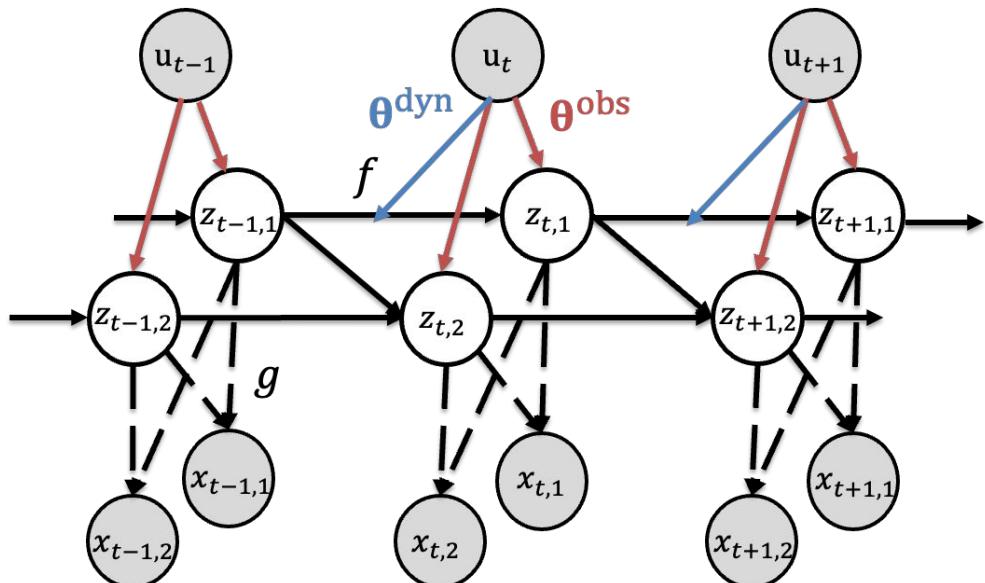
- In the temporal scenario, we can use the historical information $\{\mathbf{z}_{<t}\}$ as  an auxiliary to provide **sufficient change** and thus establish identifiability.



There is a ...

- **Stationary** latent causal process:
 $\mathbf{z}_{t,i} = f_i(\text{Pa}(\mathbf{z}_{t,i}), \varepsilon_{t,i})$
- **Invertible** data generation process:
 $\mathbf{x}_t = g(\mathbf{z}_t)$
- **Conditional independence** given historical information:
 $\mathbf{z}_{t,i} \perp\!\!\!\perp \mathbf{z}_{t,j} \mid \mathbf{z}_{<t}$

Dynamic with known non-stationary



The known non-stationarity can serve as an auxiliary variable, providing change information.

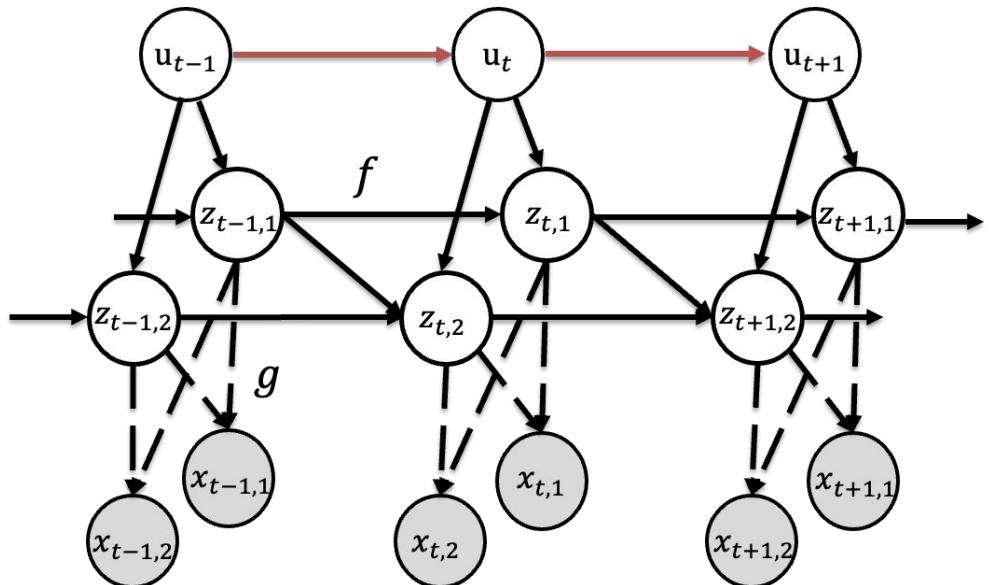
a cat.

However,

we ...

$$\left\{ \begin{array}{l} \mathbf{z}_{t,i}^{\text{fix}} = f_i(\mathbf{Pa}(\mathbf{z}_{t,i}^{\text{fix}}), \varepsilon_{t,i}) \\ \mathbf{z}_{t,j}^{\text{chg}} = f_j(\mathbf{Pa}(\mathbf{z}_{t,j}^{\text{chg}}), \theta^{\text{dyn}}, \varepsilon_{t,j}) \\ \mathbf{z}_{t,k}^{\text{obs}} = f_k(\theta^{\text{obs}}, \varepsilon_{t,k}) \\ \mathbf{x}_t = g(\mathbf{z}_t = [\mathbf{z}_t^{\text{fix}}, \mathbf{z}_t^{\text{chg}}, \mathbf{z}_t^{\text{obs}}]) \end{array} \right.$$

Dynamic with unknown non-stationary



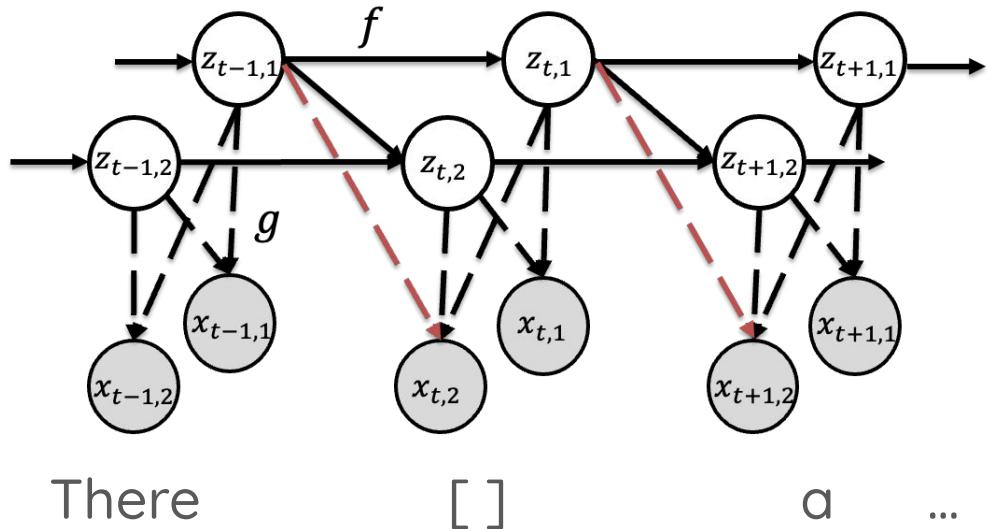
We can leverage extra assumptions to help estimate the non-stationarity, e.g., the Markov assumption.

$$\left\{ \begin{array}{l} \mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_t \sim \text{Markov Chain}(\mathbf{A}) \\ \mathbf{z}_{t,i} = f_i(\mathbf{Pa}(\mathbf{z}_{t,i}), \mathbf{u}_t, \varepsilon_{t,i}) \\ \mathbf{x}_t = g(\mathbf{z}_t) \end{array} \right.$$

“This rule applies until an exception is learned.”

Non-Invertible generation process

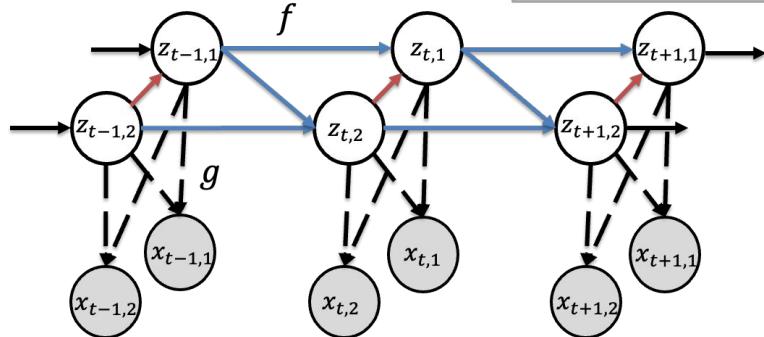
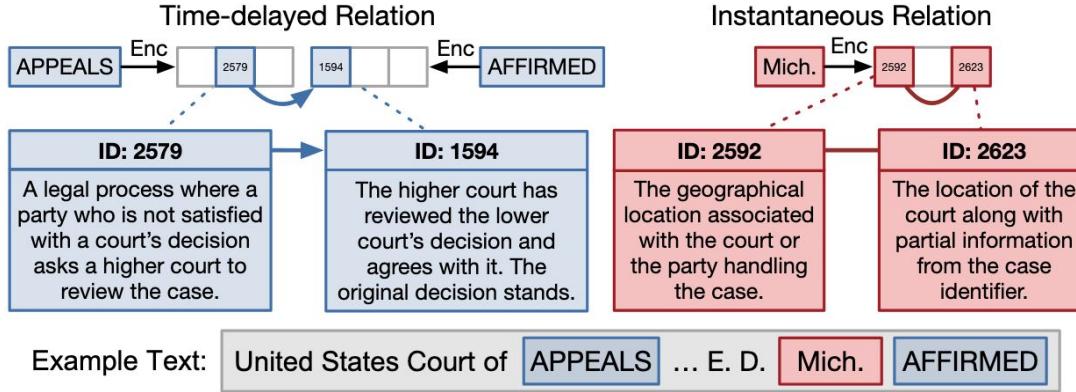
The generation process may be non-invertible, caused by typos/masks.



We can leverage the context to recover the lost information

$$\begin{cases} \mathbf{z}_t = m(\mathbf{x}_{t:t-u}) \\ \mathbf{z}_{t,i} = f_i(\mathbf{Pa}(z_{t,i}), \varepsilon_{t,i}) \\ \mathbf{x}_t = g(\mathbf{z}_{t:t-\tau}) \end{cases}$$

Instantaneous dependency



$$\mathbf{x}_t = g(\mathbf{z}_t)$$

$$z_{it} = f_i(\mathbf{Pa}_d(z_{it}), \mathbf{Pa}_e(z_{it}), \varepsilon_{it})$$

What principles can we use to learn representations?

★ Sufficient Change Principle

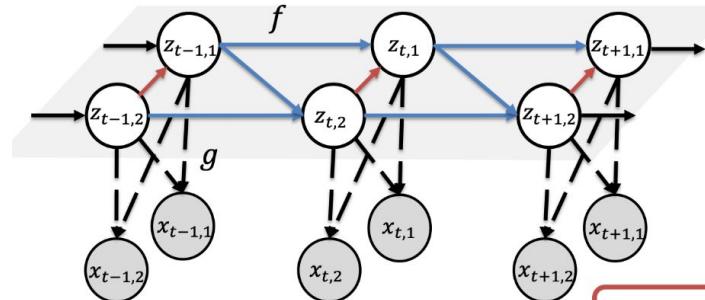
★ The Sparsity Principle

★ Learning Framework

★ Application Showcase

The sparsity principle

$$\mathbf{x}_t = g(\mathbf{z}_t) \quad z_{it} = f_i(\mathbf{Pa}_d(z_{it}), \mathbf{Pa}_e(z_{it}), \varepsilon_{it})$$



Identification Condition

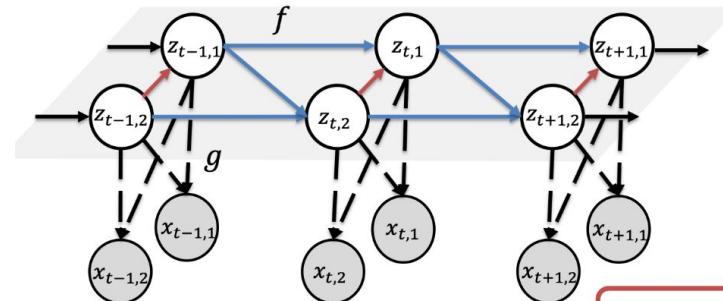
If the data is generated as described above and meets the following criteria:

- [Sparse Latent Process]: sparse enough to provide *effect change*.
- [Sparse Constraint]: the edges of the estimated Markov Network $|\tilde{\mathcal{M}}|$ is minimal.
- [Invertible function and smooth density]: g is invertible, and $p_{\mathbf{z}_t | \mathbf{z}_{<t}}$ is smooth.
- [Sufficient change]: there exist enough values of $\{\mathbf{z}_{<t}\}$ providing enough changes.
- [Marginal distribution matching]: $p_{\mathbf{x}_t | \mathbf{x}_{<t}} = p_{\hat{\mathbf{x}}_t | \hat{\mathbf{x}}_{<t}}$

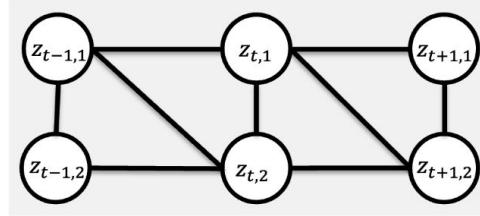
Then, the learned latent variables are component-wise identifiable.

The sparsity principle

$$\mathbf{x}_t = g(\mathbf{z}_t) \quad z_{it} = f_i(\mathbf{Pa}_d(z_{it}), \mathbf{Pa}_e(z_{it}), \varepsilon_{it})$$



Markov Network \mathcal{M}



Identification Condition

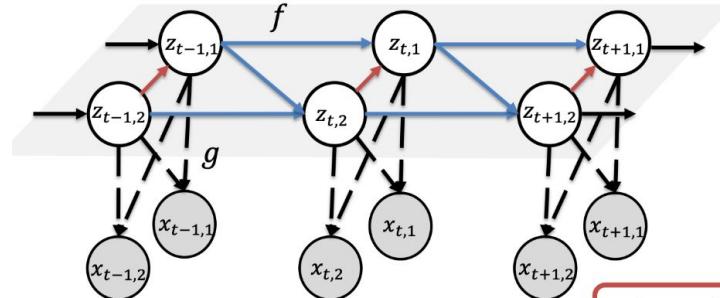
If the data is generated as described above and meets the following criteria:

- [Sparse Latent Process]: sparse enough to provide *effect change*.
- [Sparse Constraint]: the edges of the estimated Markov Network $|\tilde{\mathcal{M}}|$ is minimal.
- [Invertible function and smooth density]: g is invertible, and $p_{\mathbf{z}_t | \mathbf{z}_{<t}}$ is smooth.
- [Sufficient change]: there exist enough values of $\{\mathbf{z}_{<t}\}$ providing enough changes.
- [Marginal distribution matching]: $p_{\mathbf{x}_t | \mathbf{x}_{<t}} = p_{\hat{\mathbf{x}}_t | \hat{\mathbf{x}}_{<t}}$

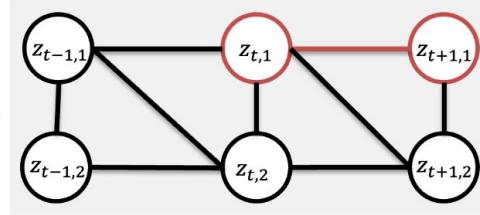
Then, the learned latent variables are component-wise identifiable.

The sparsity principle

$$\mathbf{x}_t = g(\mathbf{z}_t) \quad z_{it} = f_i(\mathbf{Pa}_d(z_{it}), \mathbf{Pa}_e(z_{it}), \varepsilon_{it})$$



Markov Network \mathcal{M}



Identification Condition

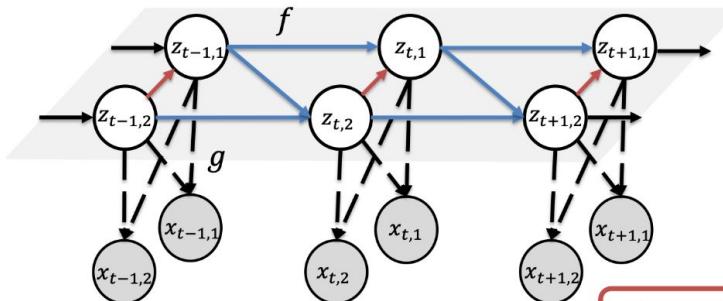
If the data is generated as described above and meets the following criteria:

- [Sparse Latent Process]: sparse enough to provide *effect change*.
- [Sparse Constraint]: the edges of the estimated Markov Network $|\tilde{\mathcal{M}}|$ is minimal.
- [Invertible function and smooth density]: g is invertible, and $p_{\mathbf{z}_t | \mathbf{z}_{<t}}$ is smooth.
- [Sufficient change]: there exist enough values of $\{\mathbf{z}_{<t}\}$ providing enough changes.
- [Marginal distribution matching]: $p_{\mathbf{x}_t | \mathbf{z}_{<t}} = p_{\hat{\mathbf{x}}_t | \hat{\mathbf{z}}_{<t}}$

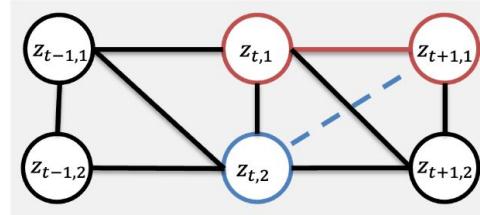
Then, the learned latent variables are component-wise identifiable.

The sparsity principle

$$\mathbf{x}_t = g(\mathbf{z}_t) \quad z_{it} = f_i(\mathbf{Pa}_d(z_{it}), \mathbf{Pa}_e(z_{it}), \varepsilon_{it})$$



Markov Network \mathcal{M}



Identification Condition

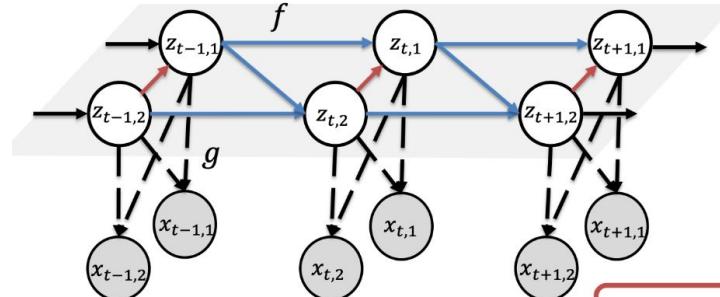
If the data is generated as described above and meets the following criteria:

- [Sparse Latent Process]: sparse enough to provide *effect change*.
- [Sparse Constraint]: the edges of the estimated Markov Network $|\tilde{\mathcal{M}}|$ is minimal.
- [Invertible function and smooth density]: g is invertible, and $p_{\mathbf{z}_t | \mathbf{z}_{<t}}$ is smooth.
- [Sufficient change]: there exist enough values of $\{\mathbf{z}_{<t}\}$ providing enough changes.
- [Marginal distribution matching]: $p_{x_t | x_{<t}} = p_{\tilde{x}_t | \tilde{x}_{<t}}$

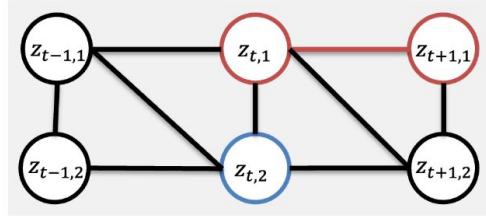
Then, the learned latent variables are component-wise identifiable.

The sparsity principle

$$\mathbf{x}_t = g(\mathbf{z}_t) \quad z_{it} = f_i(\mathbf{Pa}_d(z_{it}), \mathbf{Pa}_e(z_{it}), \varepsilon_{it})$$



Markov Network \mathcal{M}



Sparse Constraint

Distribution fitting: $|\tilde{\mathcal{M}}| \geq |\mathcal{M}|$
Sparse Constrain: $|\tilde{\mathcal{M}}| \leq |\mathcal{M}|$

Identification Condition

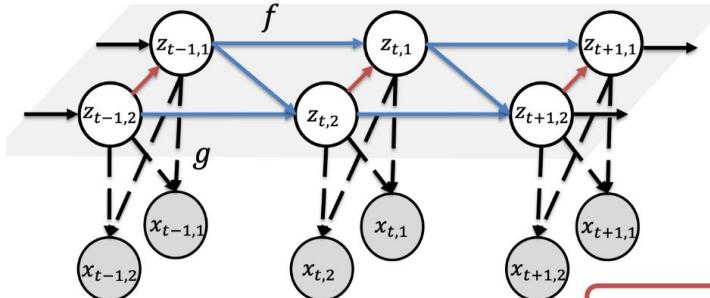
If the data is generated as described above and meets the following criteria:

- [Sparse Latent Process]: sparse enough to provide *effect change*.
- [Sparse Constraint]: the edges of the estimated Markov Network $|\tilde{\mathcal{M}}|$ is minimal.
- [Invertible function and smooth density]: g is invertible, and $p_{\mathbf{z}_t | \mathbf{z}_{<t}}$ is smooth.
- [Sufficient change]: there exist enough values of $\{\mathbf{z}_{<t}\}$ providing enough changes.
- [Marginal distribution matching]: $p_{x_t | \mathbf{x}_{<t}} = p_{\hat{x}_t | \hat{\mathbf{x}}_{<t}}$

Then, the learned latent variables are component-wise identifiable.

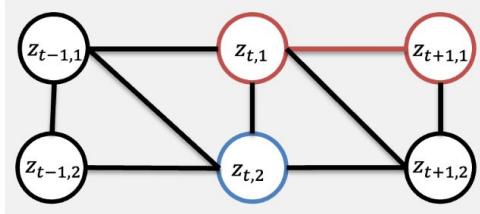
The sparsity principle

$$\mathbf{x}_t = g(\mathbf{z}_t) \quad z_{it} = f_i(\mathbf{Pa}_d(z_{it}), \mathbf{Pa}_e(z_{it}), \varepsilon_{it})$$



Markov Network \mathcal{M}

Sparse Constraint



Distribution fitting: $|\hat{\mathcal{M}}| \geq |\mathcal{M}|$
Sparse Constrain: $|\hat{\mathcal{M}}| \leq |\mathcal{M}|$

Identification Condition

If the data is generated as described above and meets the following criteria:

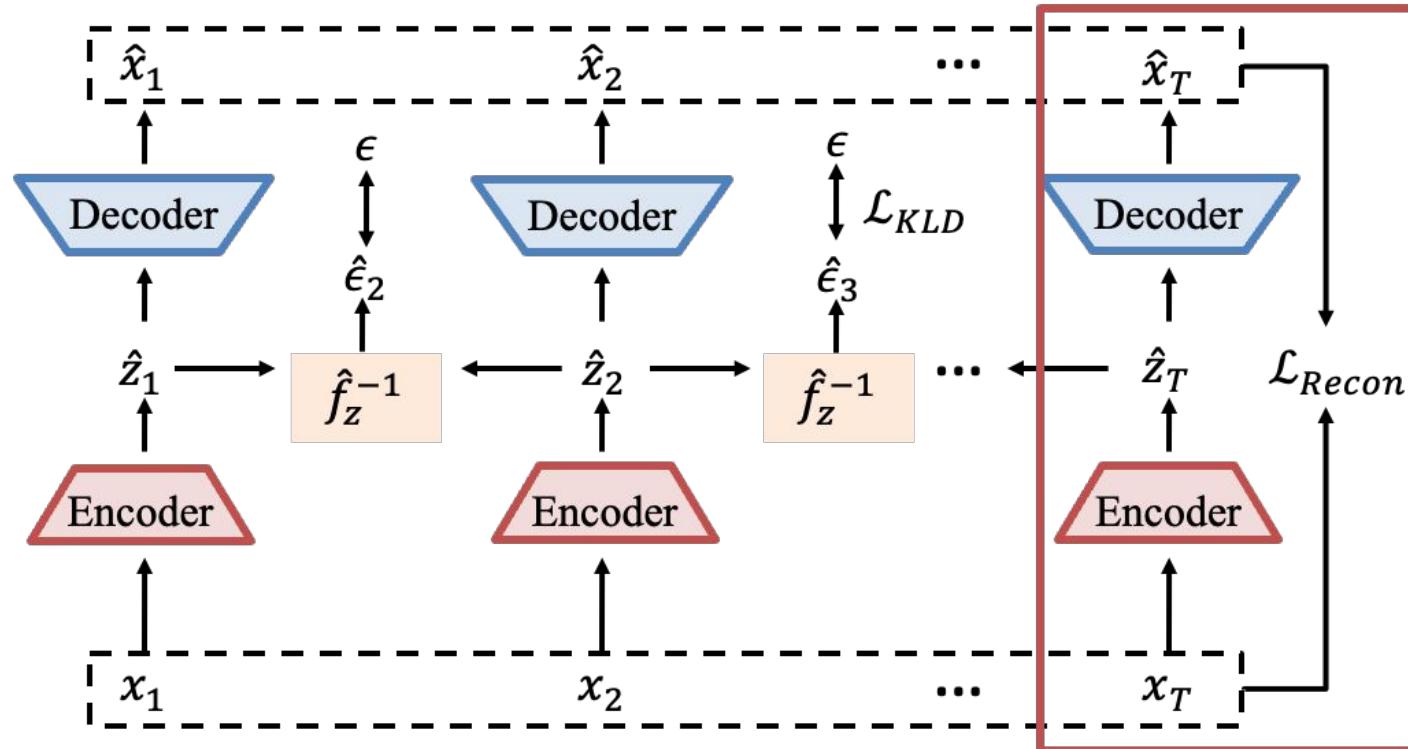
- [Sparse Latent Process]: sparse enough to provide *effect change*.
- [Sparse Constraint]: the edges of the estimated Markov Network $|\hat{\mathcal{M}}|$ is minimal.
- [Invertible function and smooth density]: g is invertible, and $p_{\mathbf{z}_t | \mathbf{z}_{<t}}$ is smooth.
- [Sufficient change]: there exist enough values of $\{\mathbf{z}_{<t}\}$ providing enough changes.
- [Marginal distribution matching]: $p_{x_t | x_{<t}} = p_{\hat{x}_t | \hat{x}_{<t}}$

Then, the learned latent variables are component-wise identifiable.

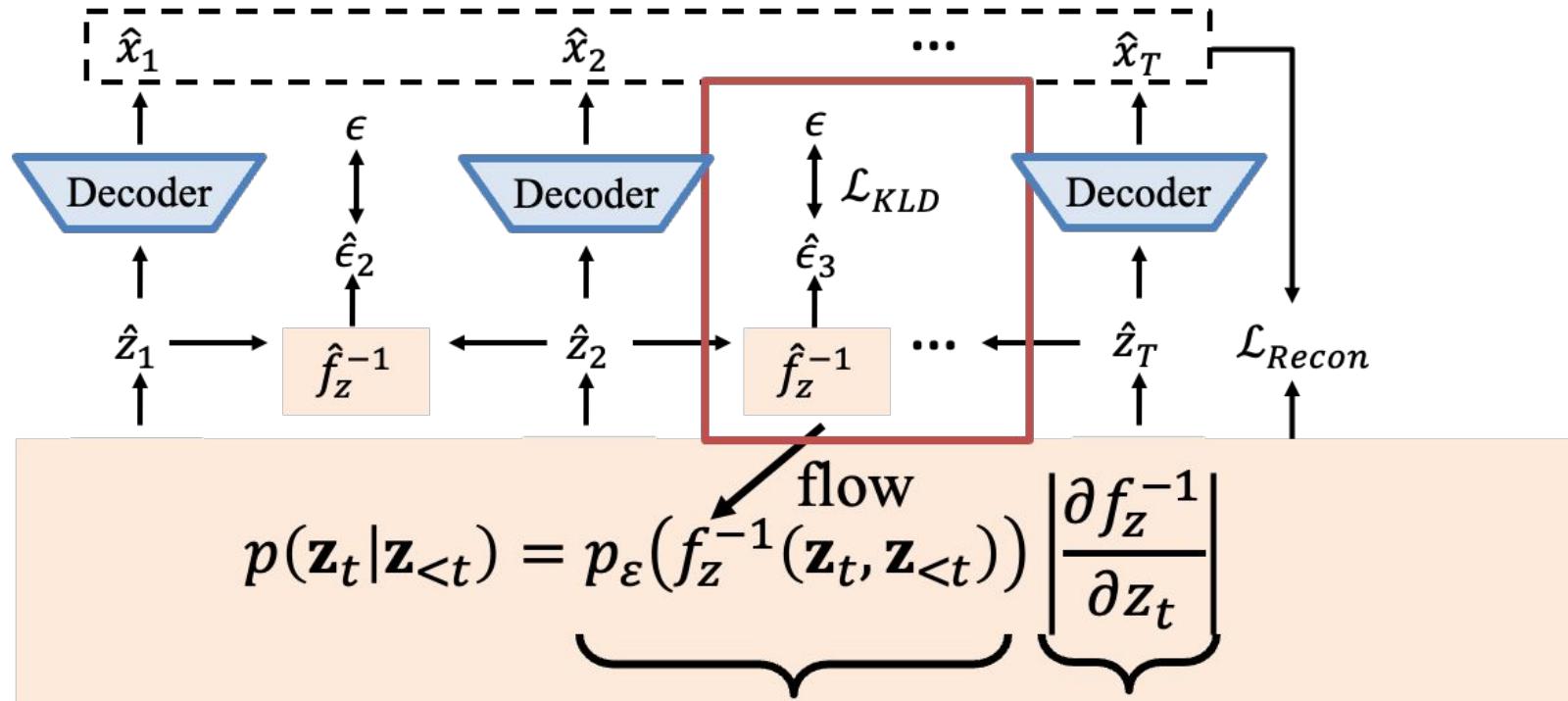
What principles can we use to learn representations?

- ★ Sufficient Change Principle
- ★ The Sparsity Principle
- ★ Learning Framework
- ★ Application Showcase

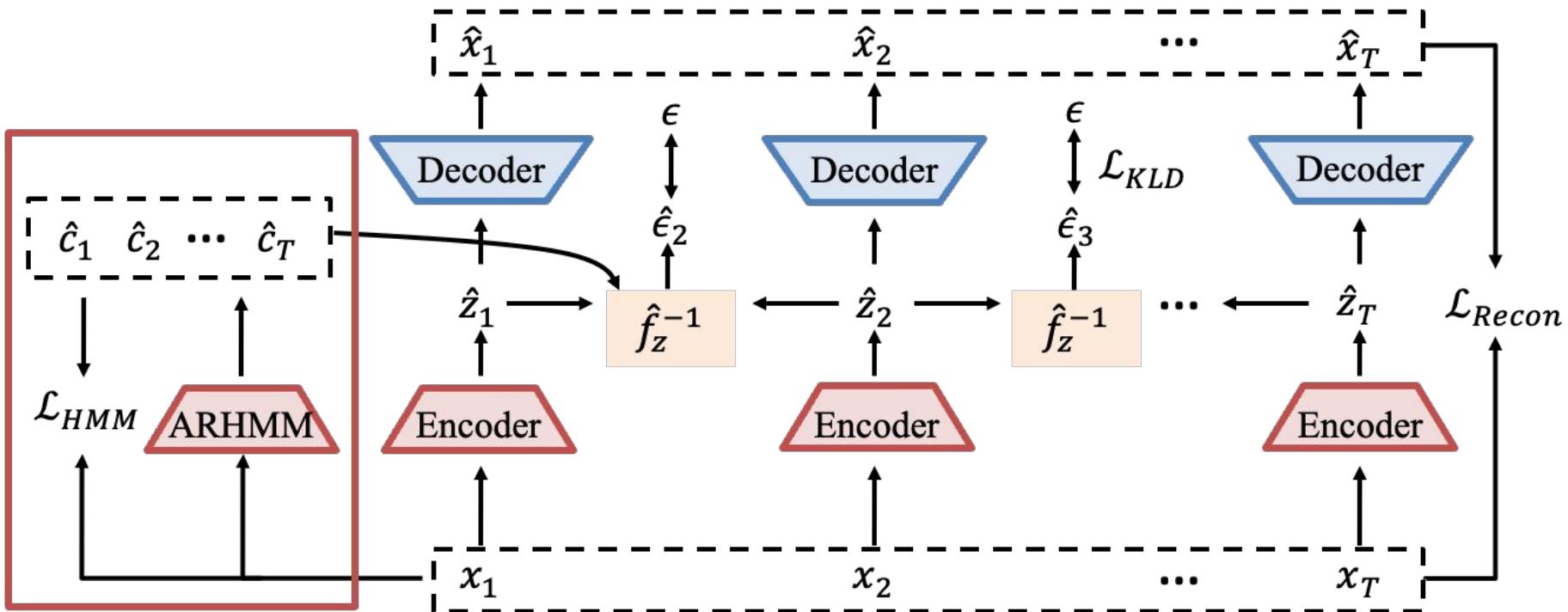
Implementation in VAE framework



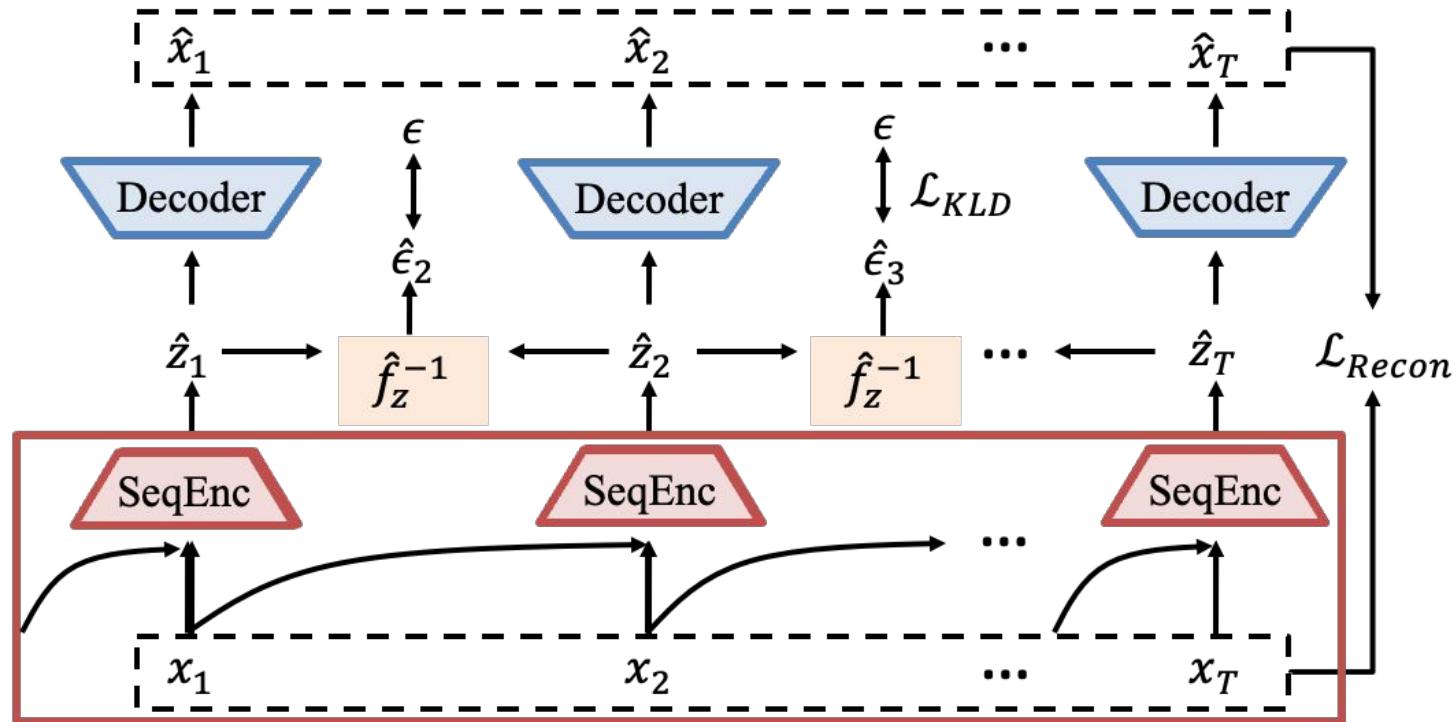
Conditional independence — prior network



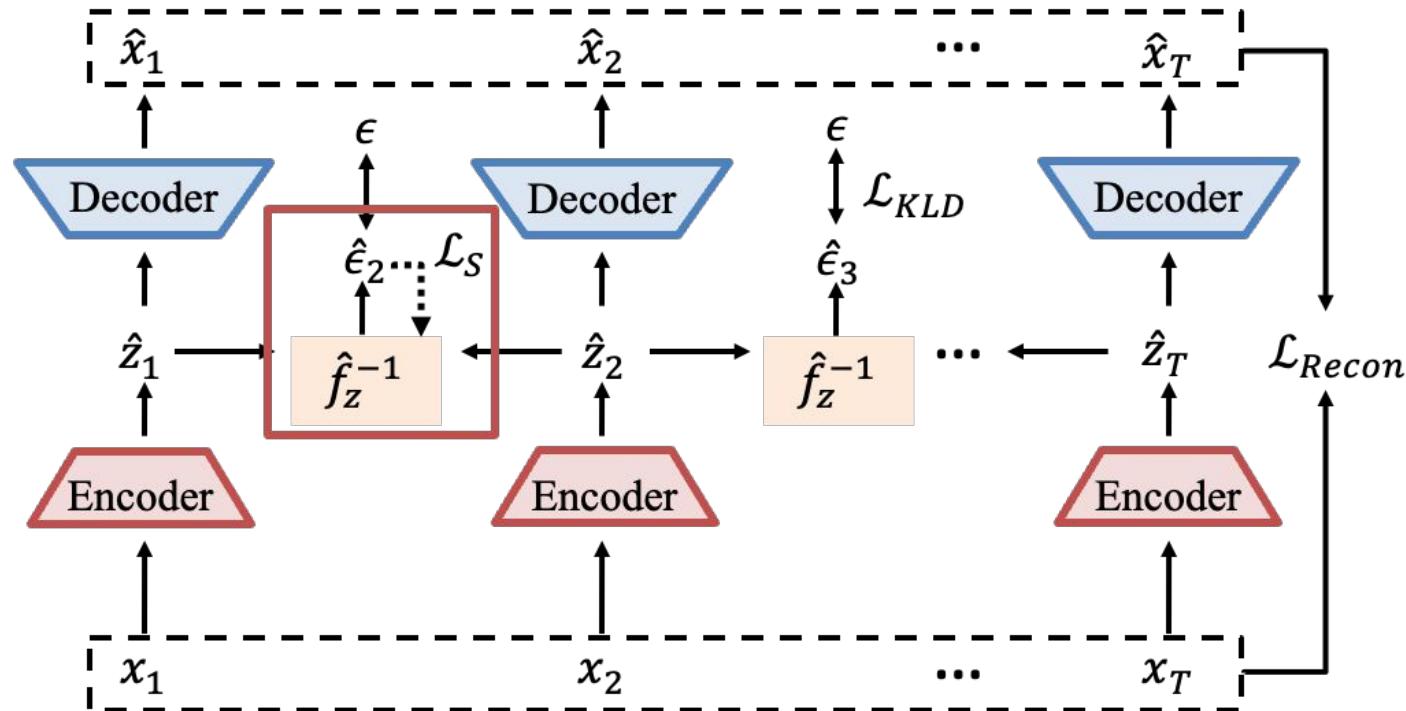
Autoregressive hidden Markov module



Non-invertibility — context encoder



Instantaneous dependency— sparsity loss

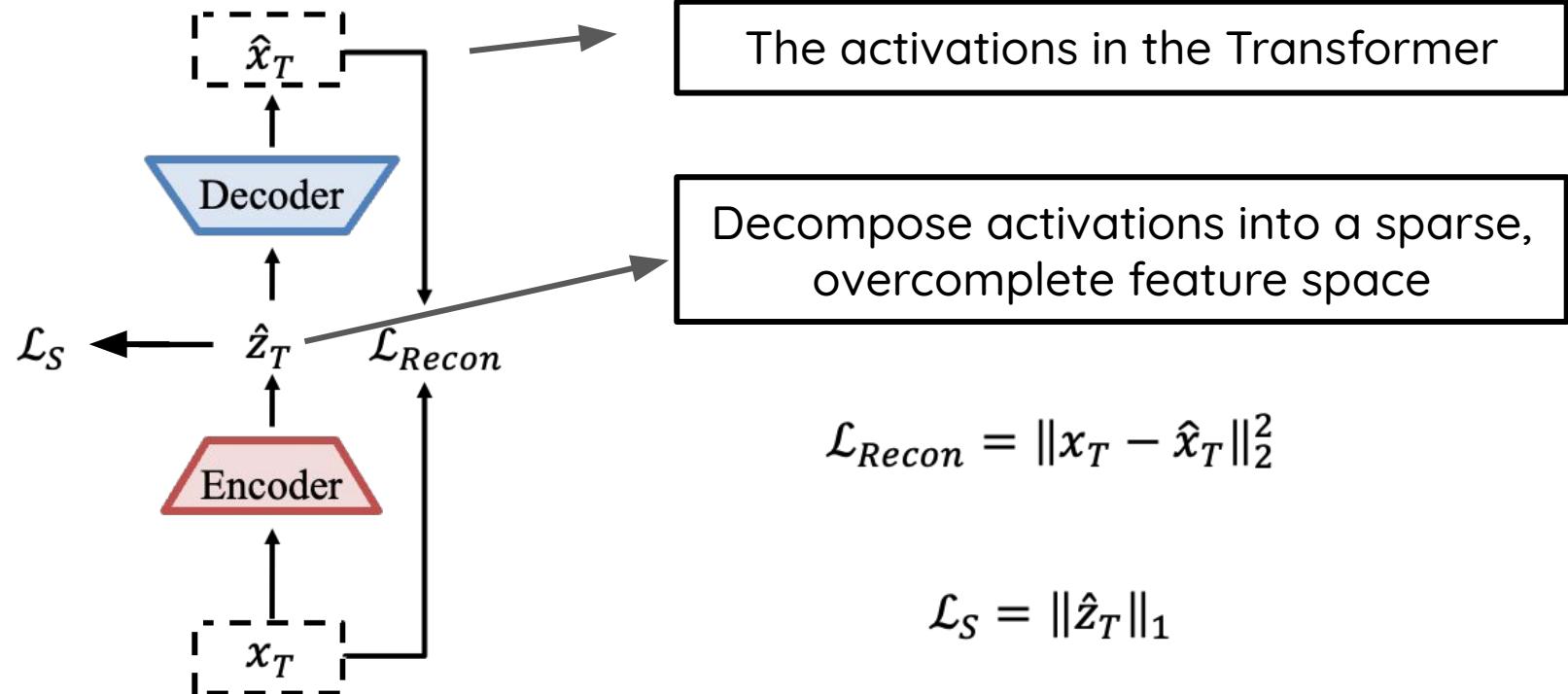


What principles can we use to learn representations?

- ★ Sufficient Change Principle
- ★ The sparsity principle
- ★ Learning Framework

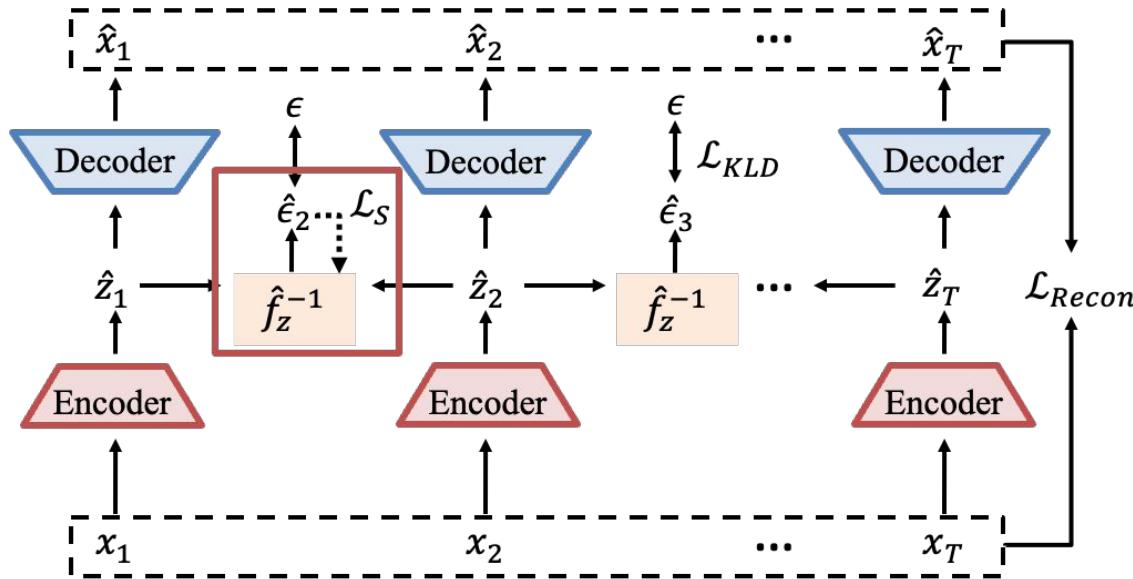
★ Application Showcase

Application for Sparse Autoencoder (SAE)

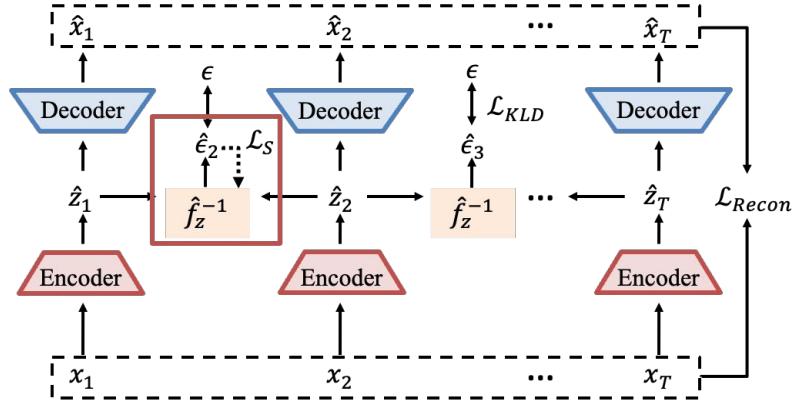


Application for Sparse Autoencoder (SAE)

Consider the temporal dynamics in the SAE, and explicitly model both time-delayed and instantaneous dependencies.



Application for Sparse Autoencoder (SAE)



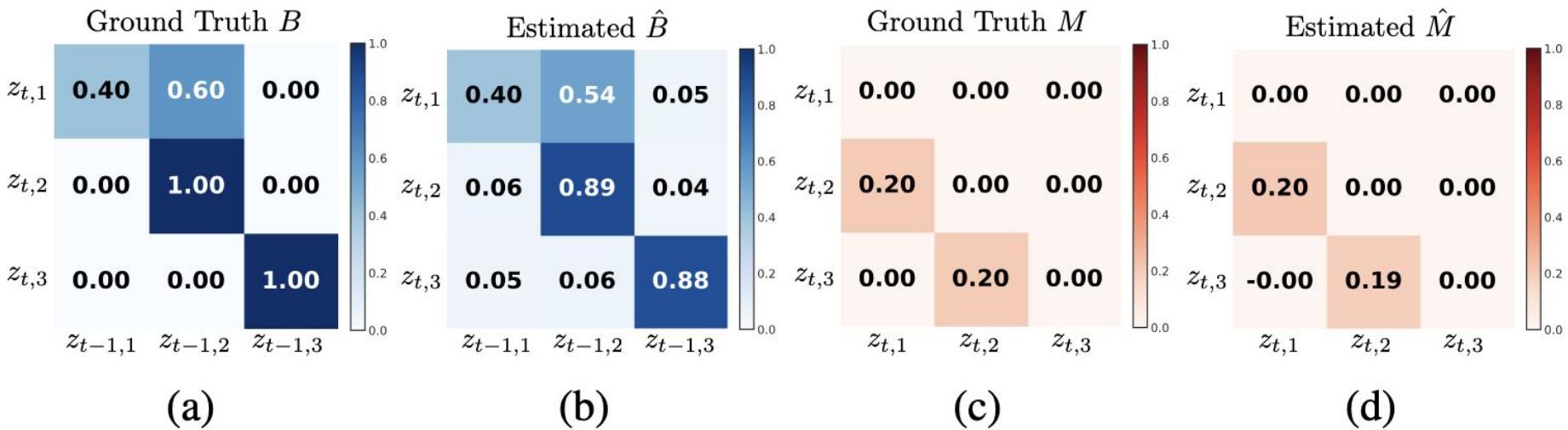
$$\mathbf{x}_t = \mathbf{g}(\mathbf{z}_t),$$
$$z_{t,i} = \underbrace{\sum_{\tau} \sum_{j \in \mathcal{J}_{i,\tau}} \mathbf{B}_{i,j,\tau} z_{t-\tau,j}}_{\text{time-delayed}} + \underbrace{\sum_{j \in \mathcal{K}_i} \mathbf{M}_{i,j} z_{t,j} + \epsilon_{t,i}}_{\text{instantaneous}},$$

$$\mathcal{L}_S = \left(\sum_{\tau} \|\widehat{\mathbf{B}}_{\tau}\|_1 \right) + \|\widehat{\mathbf{M}}\|_1 \quad \mathcal{L}_{KLD} = \mathbb{E}_{\hat{\epsilon}_t} [\|\hat{\epsilon}_t\|_1]$$

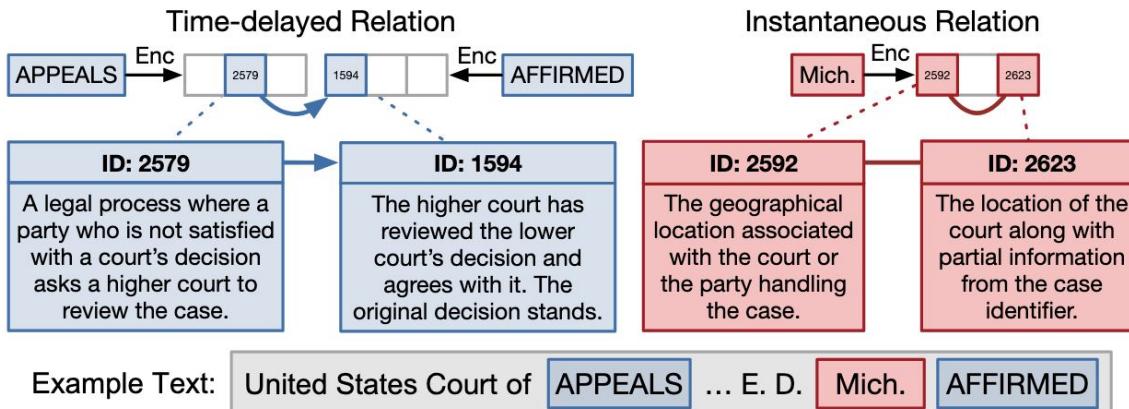
$$\mathcal{L}_{Recon} = \mathbb{E}_{x_{1:T}} \left[\sum_{t=1}^T \|x_t - \hat{x}_t\|_2^2 \right]$$

Application for Sparse Autoencoder (SAE)

In the synthetic experimental settings, both time-delayed and instantaneous causal relations have been precisely recovered.



Application for Sparse Autoencoder (SAE)



Compared with the standard SAE, representations learned under the sufficient change principle exhibit superior relation recovery ability.

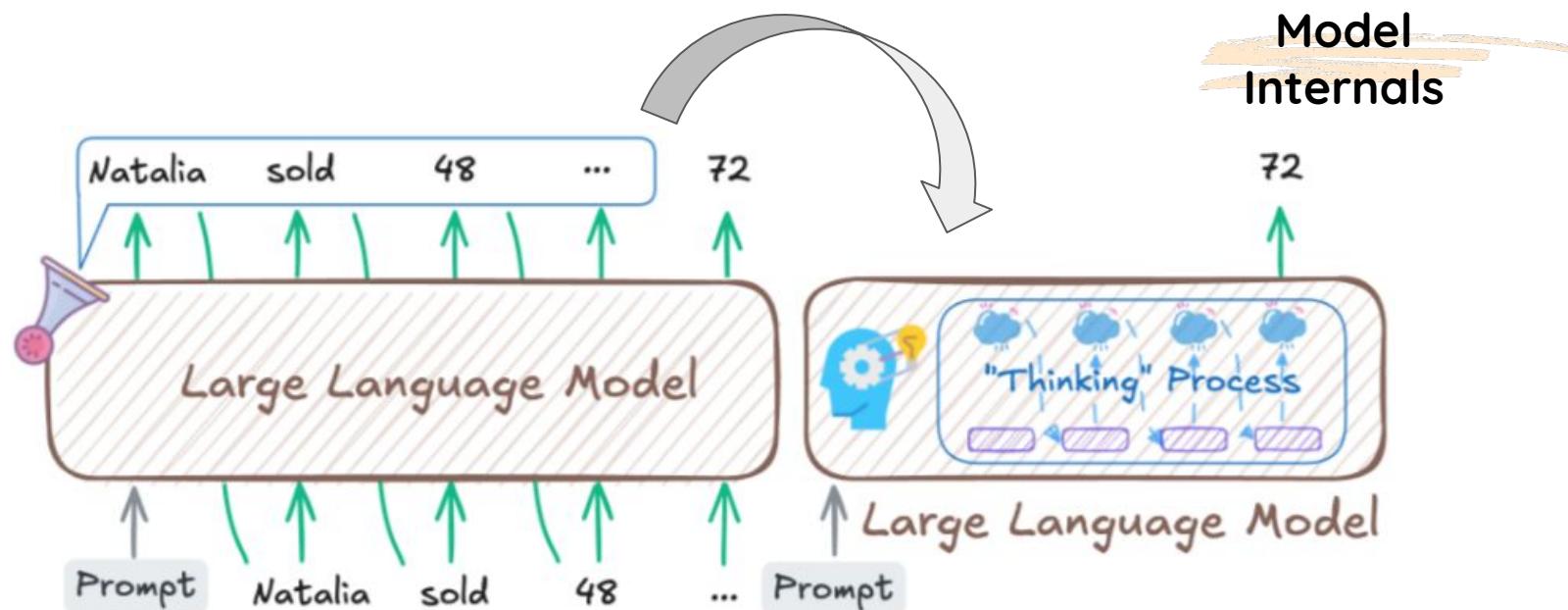
Method	Legal	XML	Email
SAE+regression	0.54	0.94	0.74
Ours	19.95	8.63	2.66

Principles summary

- ❑ Temporal dynamics in sequence (prediction)
- ❑ Structure sparsity (compact)
- ❑ No information loss
- ❑ Context-guided
- ❑ Structure-prior guidance
- ❑

Representations for Reasoning

Surface reasoning vs. latent reasoning

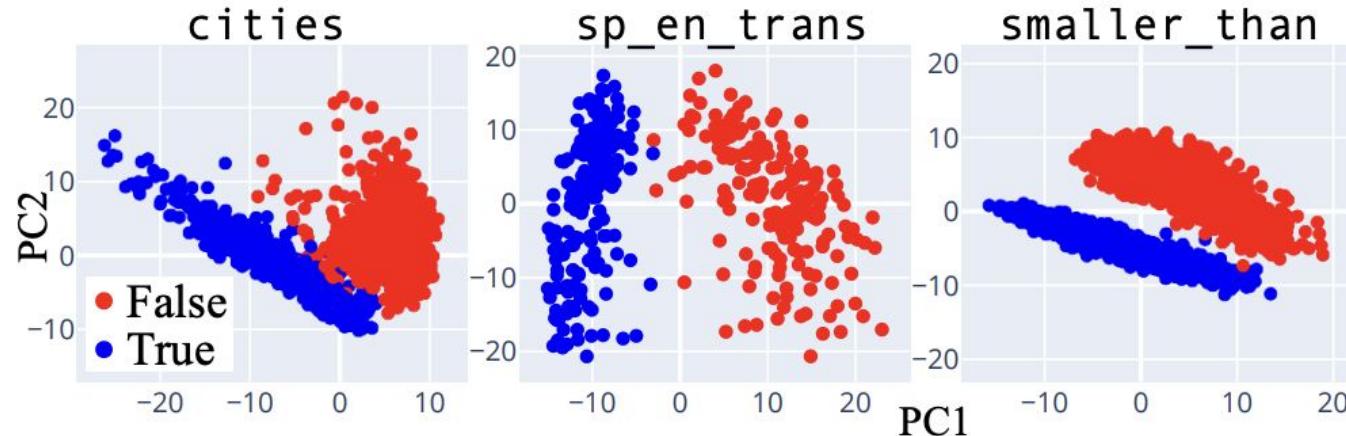


Are the internal representations informative?

The primary factors of the representation show clear linear discriminative structure in True/False tasks.

London is the capital of the UK. (True)

New York is the capital of UK (False).

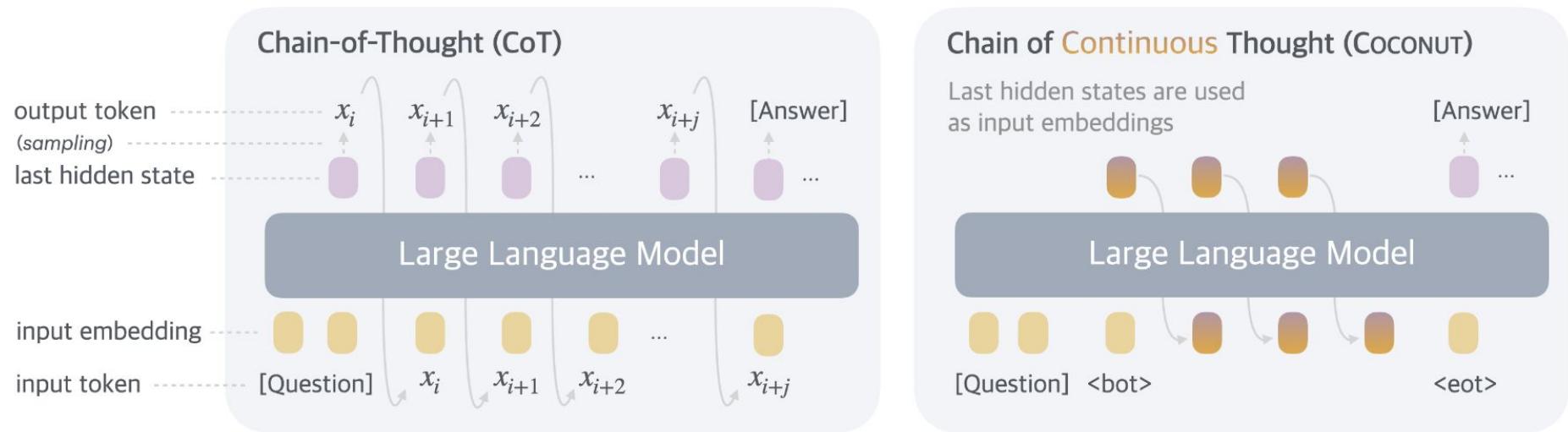


How to leverage representations for latent reasoning?

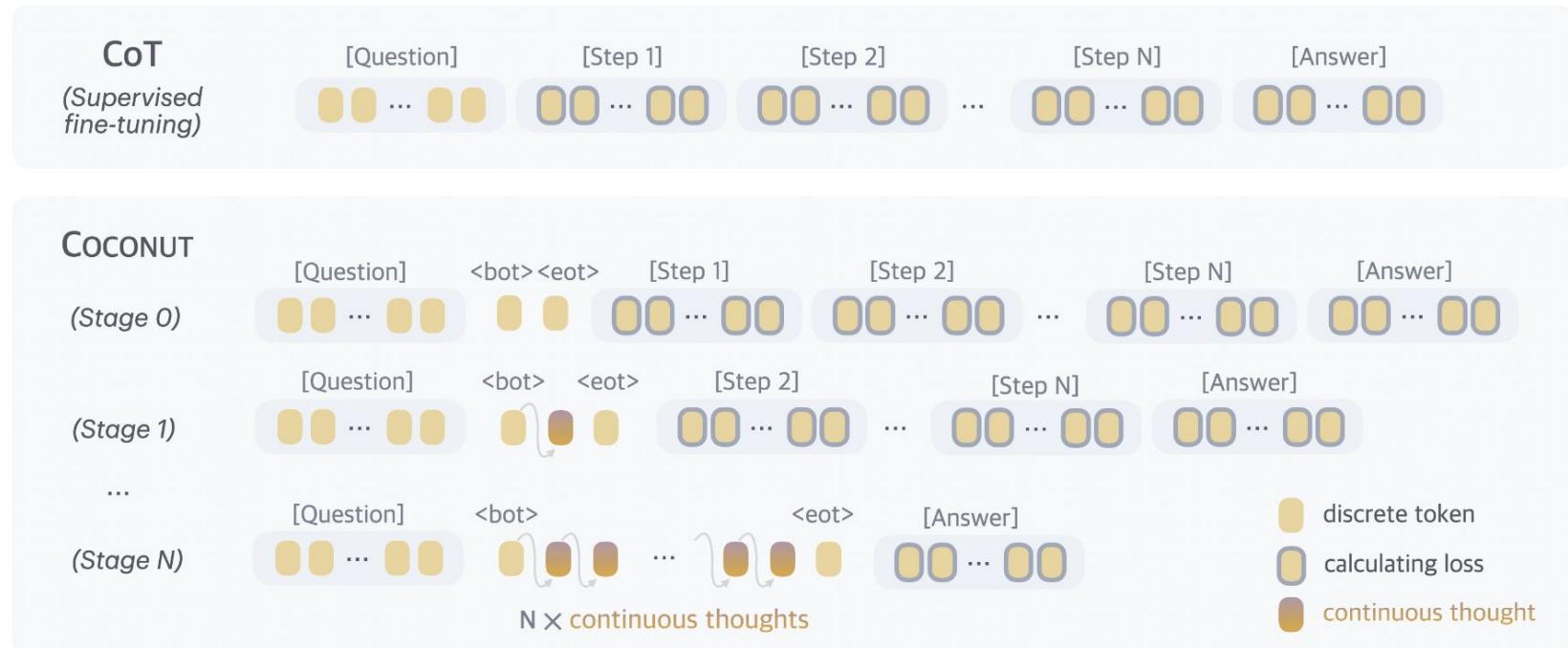
★ Latent CoT

★ Recurrent Reasoning

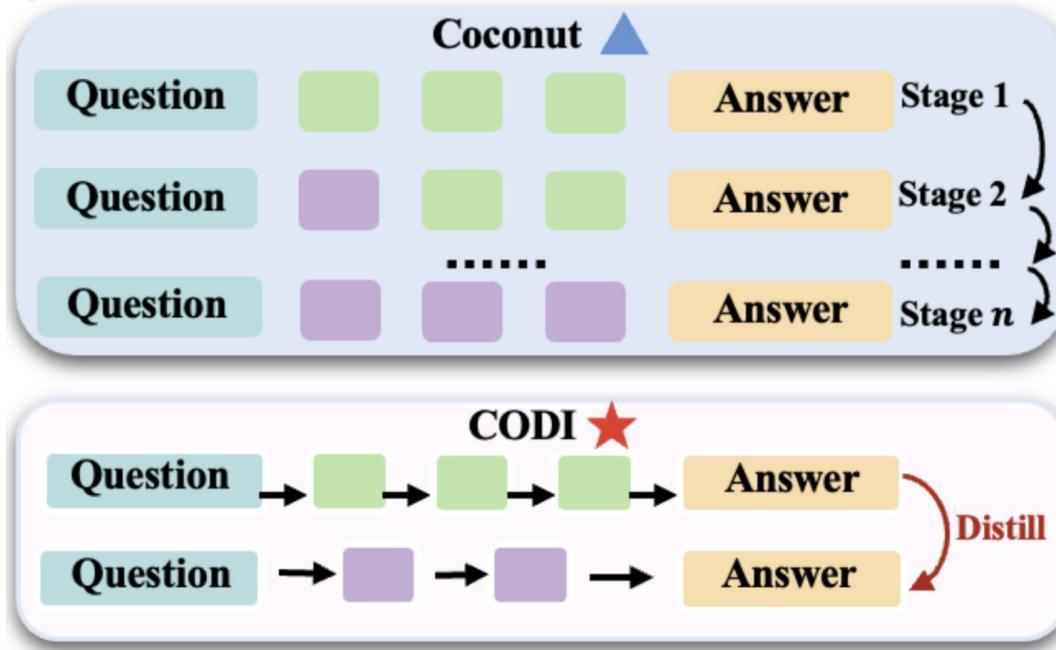
COCONUT: Latent chain-of-thought (CoT)



COCONUT: Latent chain-of-thought (CoT)

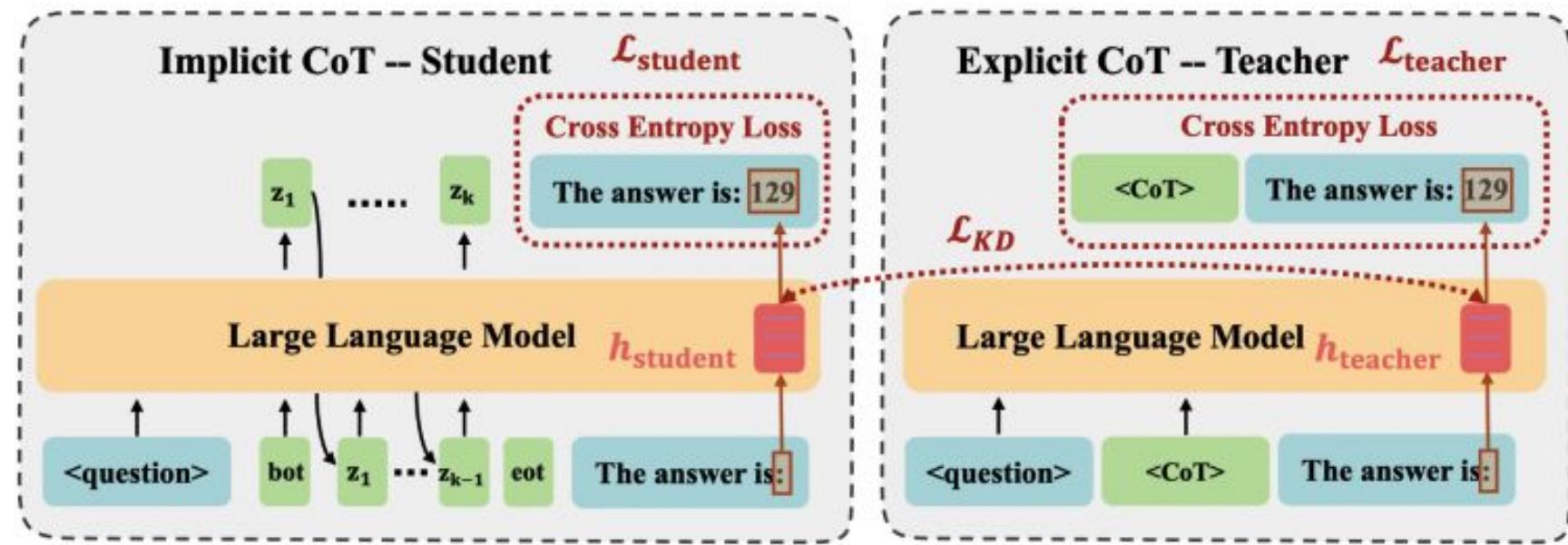


CODI: Latent CoT via Self-Distillation



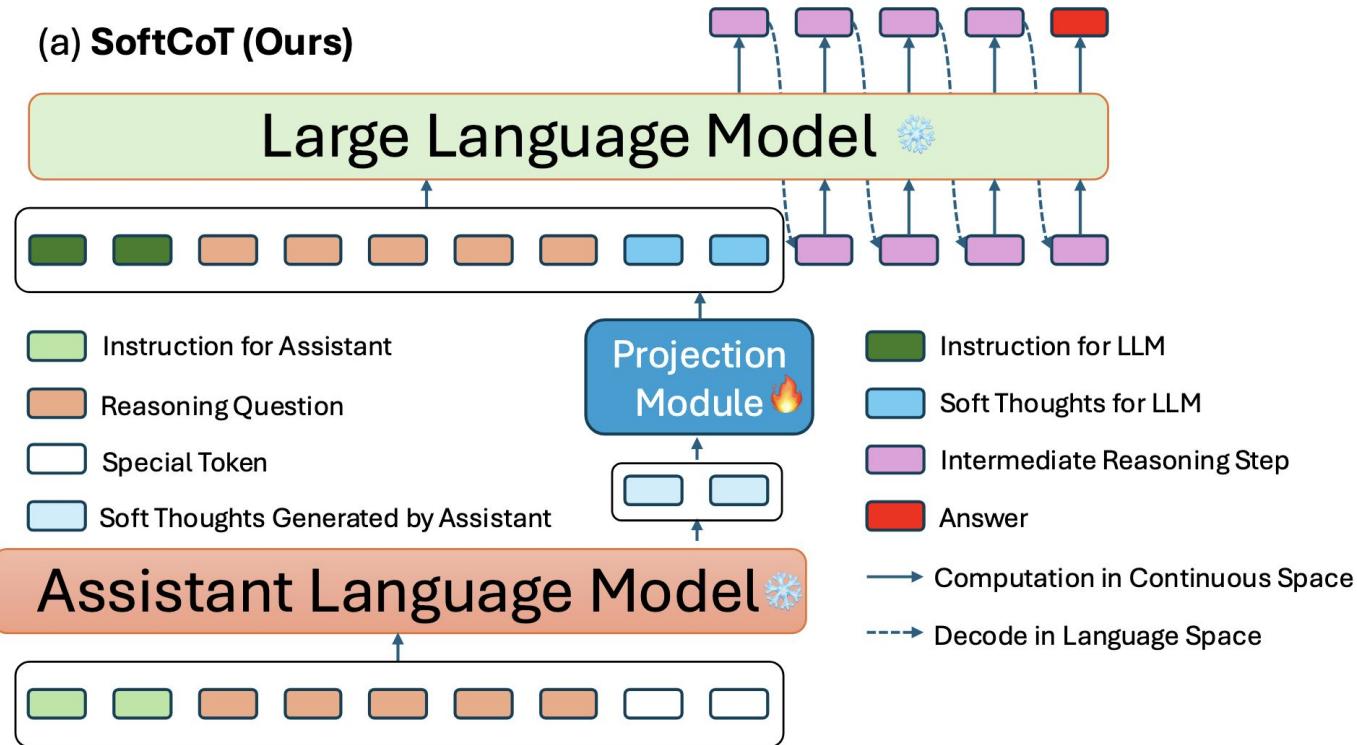
CODI jointly trains explicit CoT and a latent CoT), distilling the reasoning ability by aligning the hidden states

Latent CoT via Self-Distillation

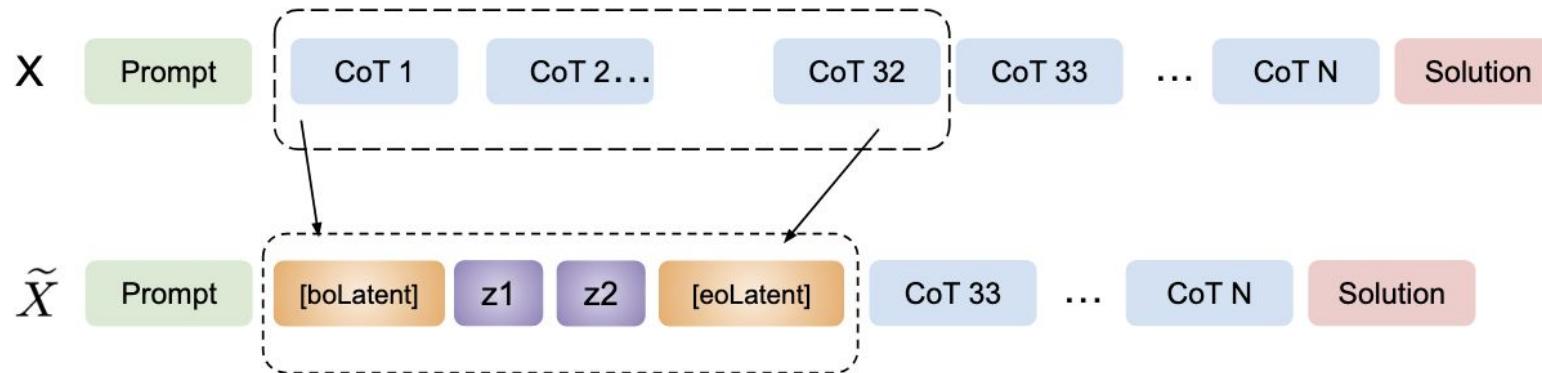


SoftCoT: learning latent COT with auxiliary LLMs

(a) SoftCoT (Ours)



Obtain the latent representation with VQVAE

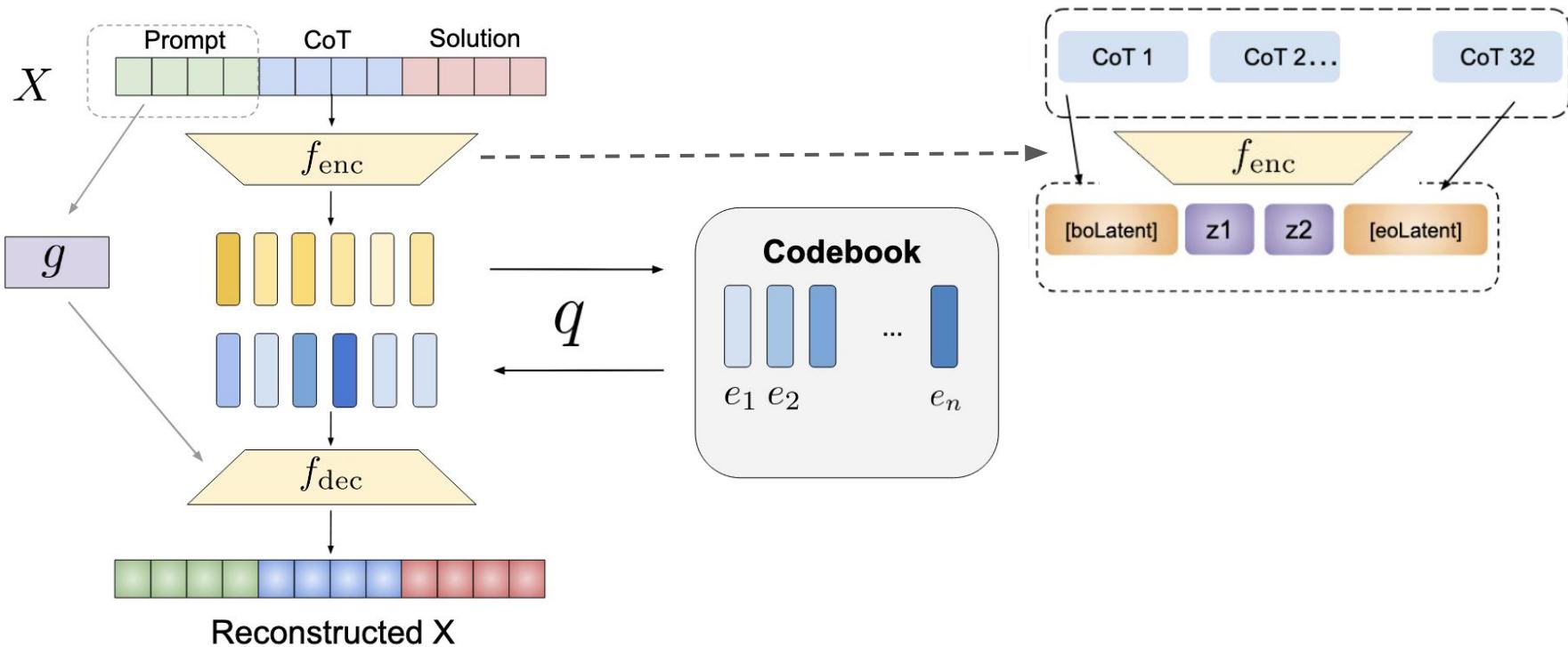


[boLatent] [eoLatent] Special delimiters that encode the start / end of the latent tokens

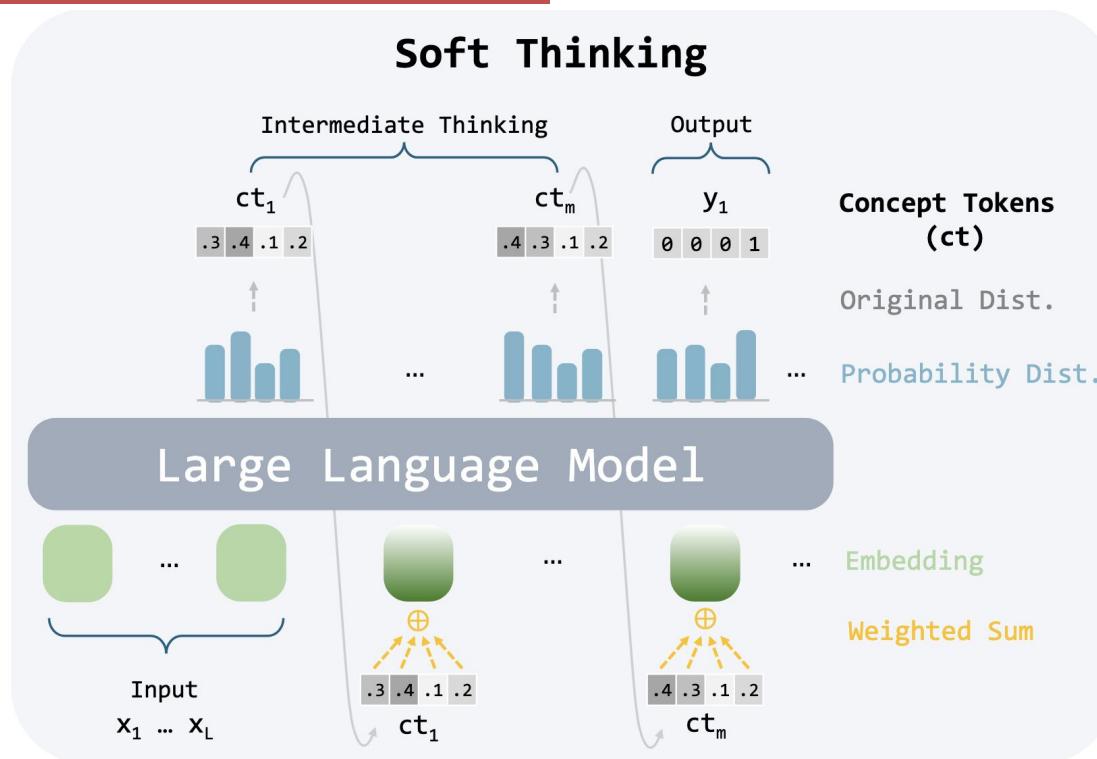
z Discrete latent tokens

CoT N The n-th CoT textual tokens

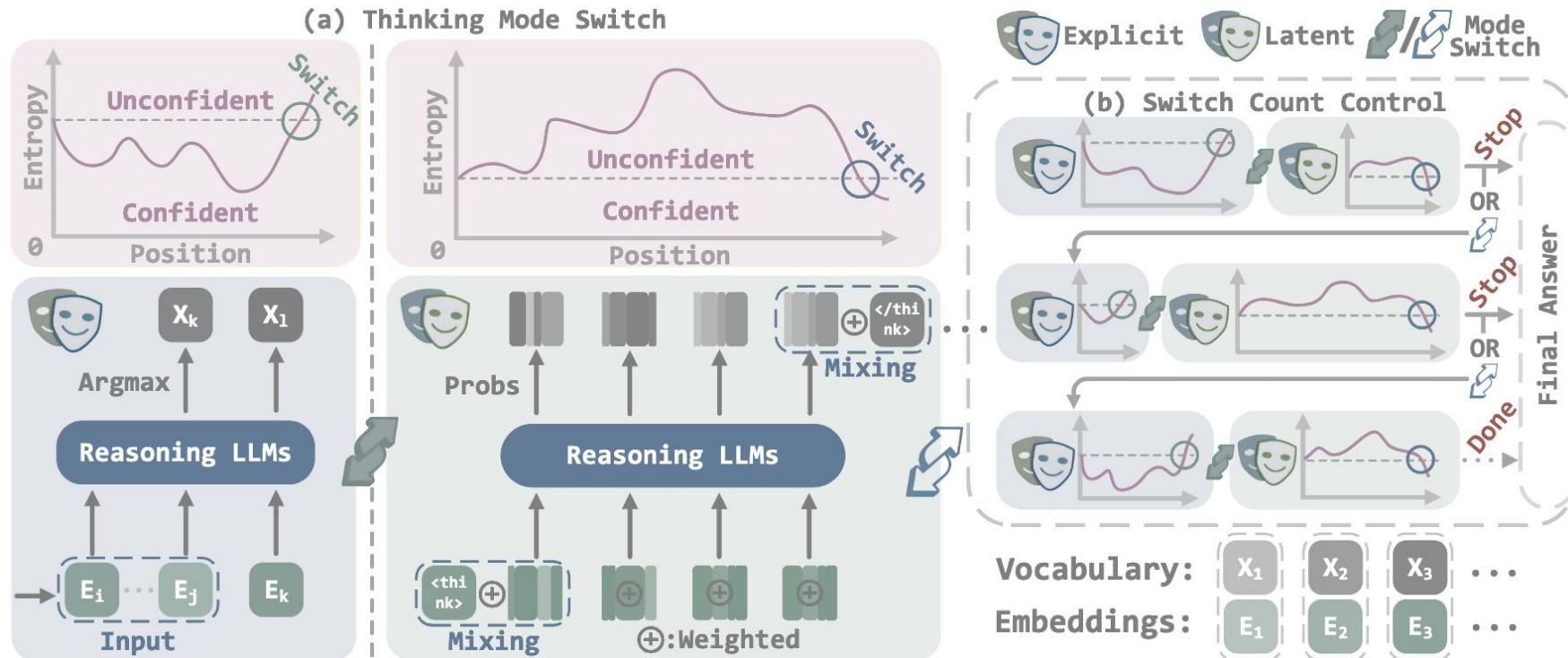
Obtain the latent representation with VQVAE



Obtain the representation with re-weighting



Switch between latent and explicit reasoning

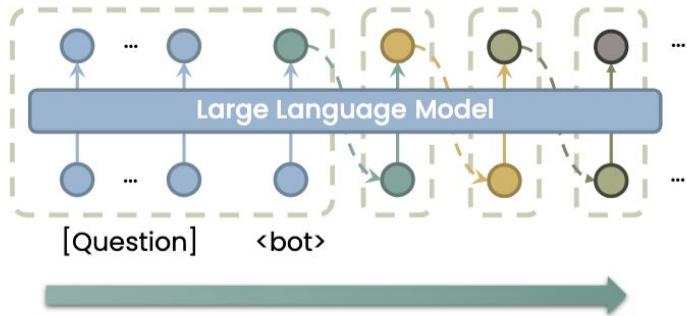


How to leverage representations for latent reasoning?

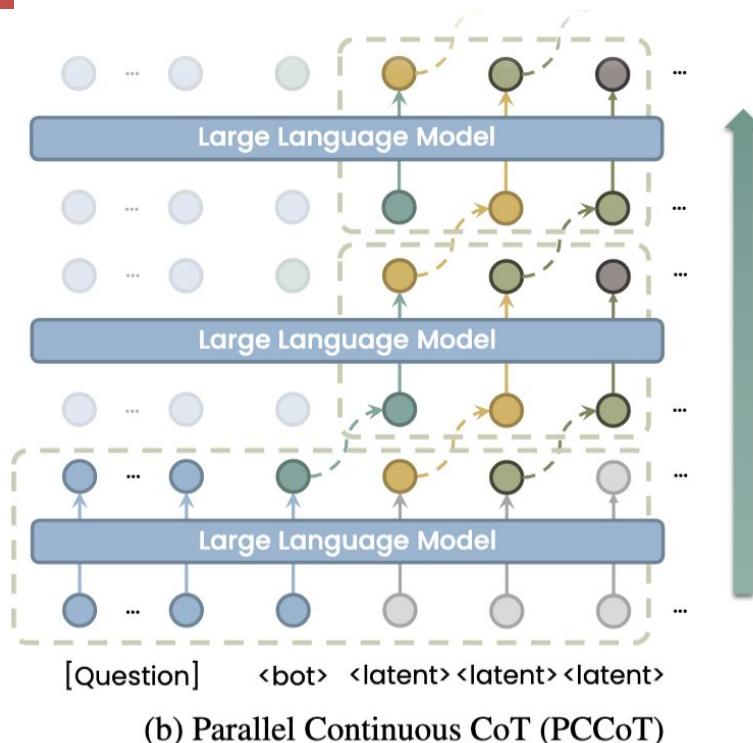
★ Latent CoT

★ Recurrent Reasoning

Latent CoT vs. Recurrent refinement



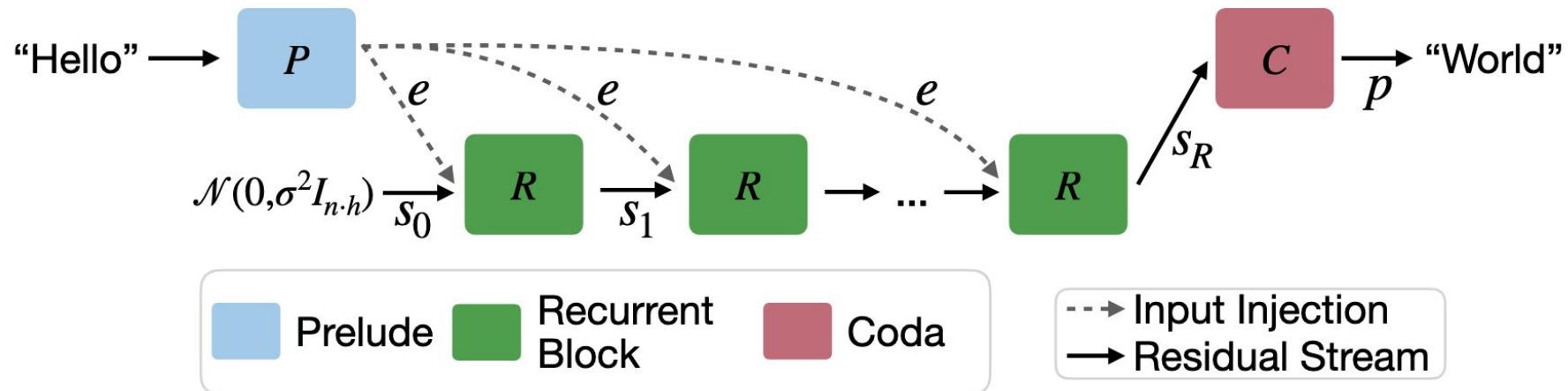
(a) Continuous CoT



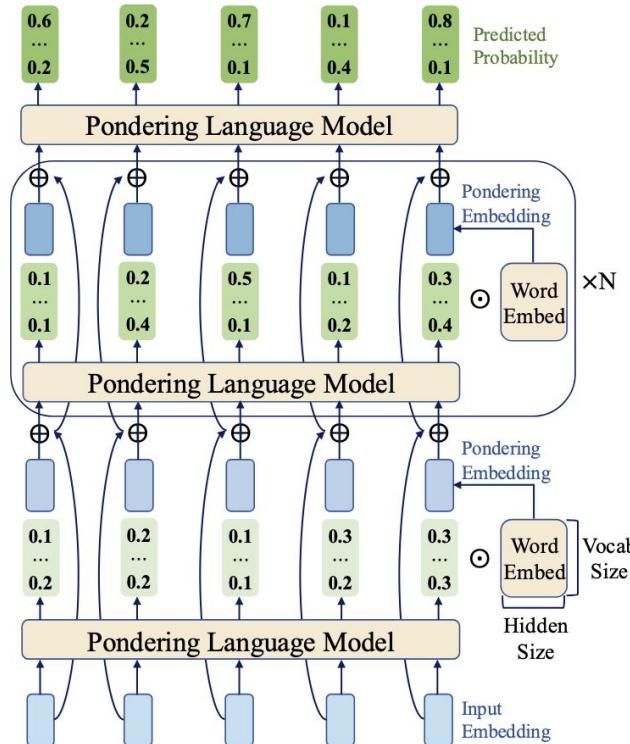
(b) Parallel Continuous CoT (PCCoT)

Latent CoT in the depth with a recurrent block

By iteratively applying a recurrent block, the model can implicitly perform reasoning in latent space, allowing it to unroll to arbitrary depth at test time.



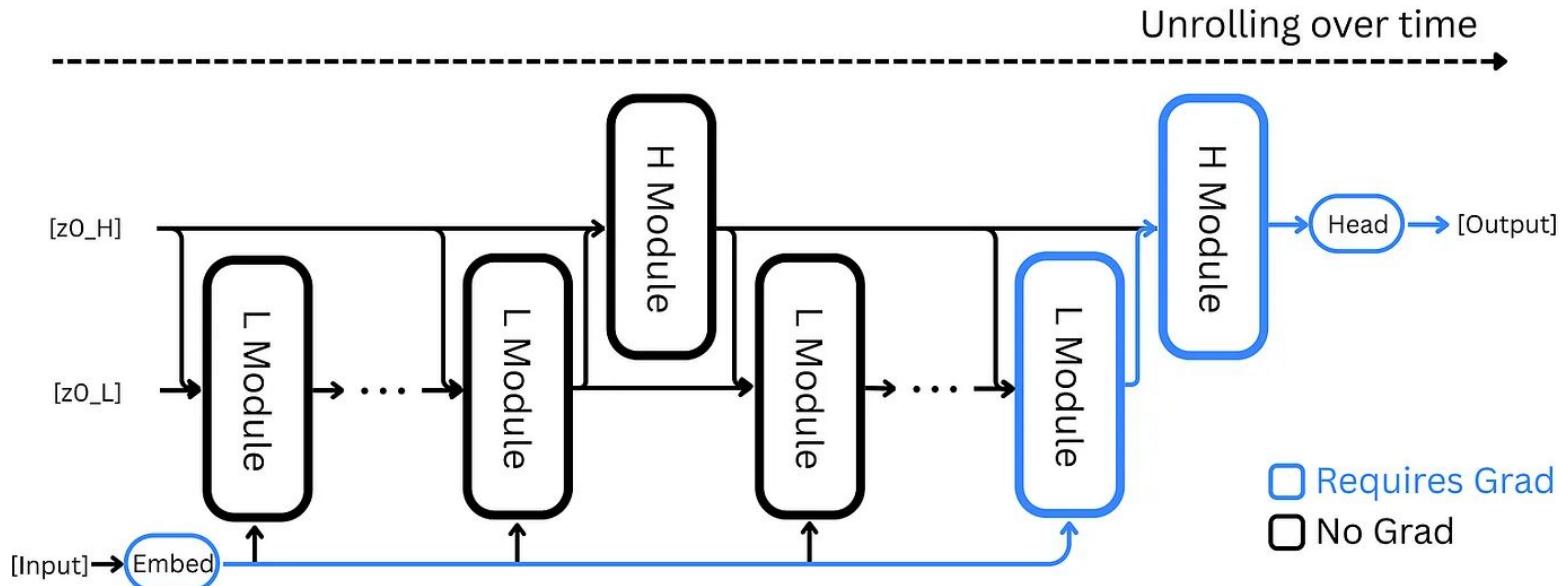
Recurrent reasoning with weighted embeddings



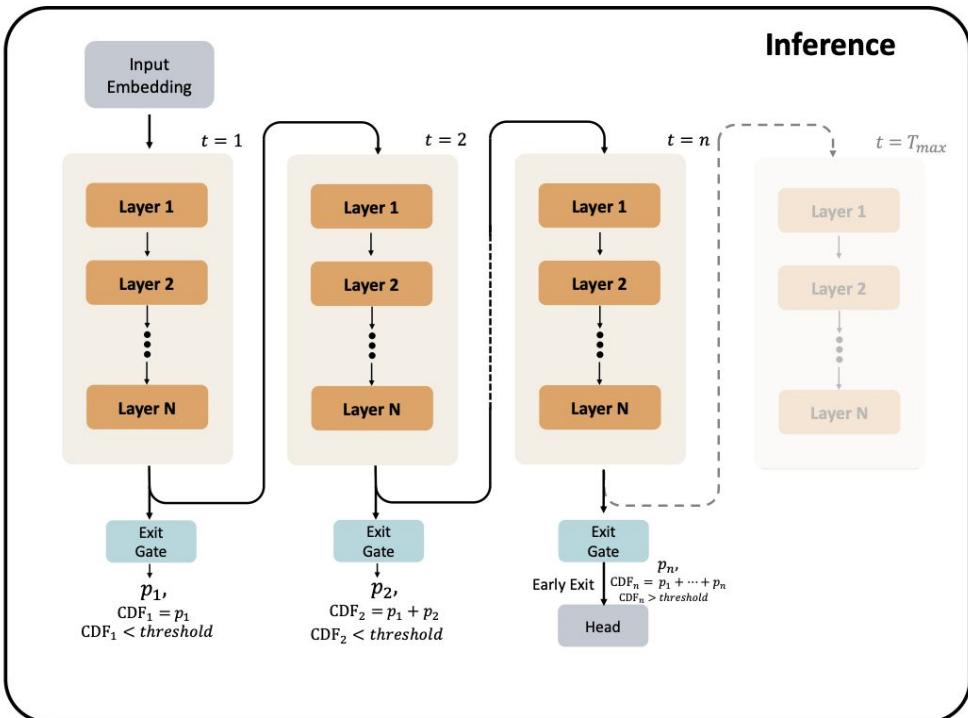
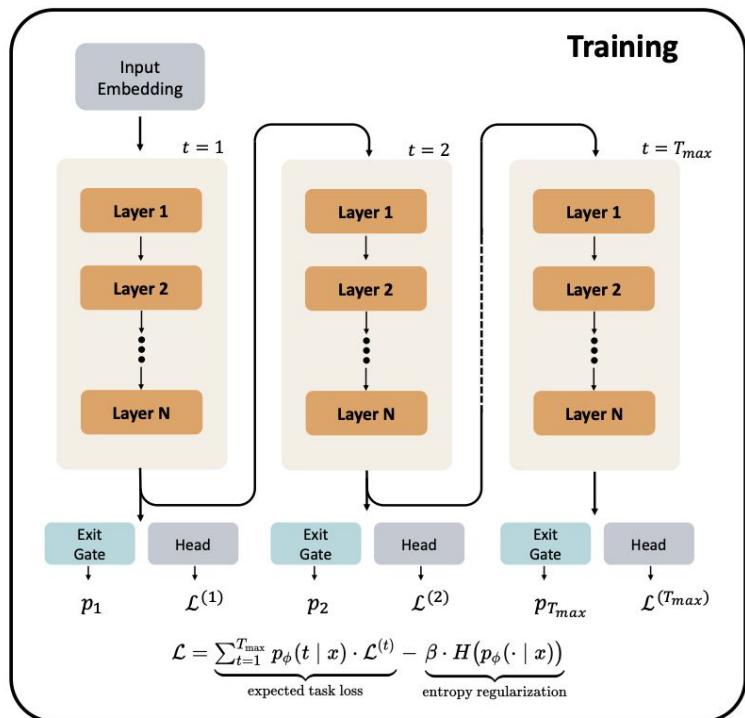
```
class PonderingLanguageModel(nn.Module):
    def __init__(self, lm, v, h, k):
        self.lm = lm # language model
        self.vocab_size = v
        self.hidden_dim = h
        self.pondering_steps = k
        self.embedding = nn.Parameter(torch.
            randn(v, h), requires_grad=True)

    def forward(self, input_tokens):
        input_embedding =
            self.embedding[input_tokens]
        #Iterative pondering
        for t in range(self.pondering_steps):
            predicted_prob = self.lm(
                input_embedding)
            pondering_embedding = torch.
                matmul(predicted_prob, self.
                    embedding)
            input_embedding = input_embedding
                + pondering_embedding
        #Final forward pass
        final_prob = self.lm(input_embedding)
        return final_prob
```

Recurrent reasoning with hierarchy



Recurrent reasoning is scaled in pre-training



When and Why latent recurrent reasoning works

The recurrent models may pitfall in ‘false’ fixed points, which requires the perturbation (in both input and checkpoints) to escape.

Scientific Understanding of HRM

Failure on *Extremely Simple* Puzzles

1	2	3	4	5	6	7	8	9
7	5	4	9	8	3	6	1	2
6	3	9	7	1	2	8	4	5
4	2	1	8	7	9	5	3	6
9	8	7	6	3	5	4	2	1
5	6	3	1	2	4	9	8	7
1	9	8	4	5	7	2	6	3
3	7	5	2	6	8	1	9	4
2	4	6	3	9	1	7	5	8



9	1	2	3	4	5	6	3	7	9
7	5	4	9	8	3	6	1	2	
6	3	9	7	1	2	8	4	5	
4	2	1	8	7	9	5	3	6	
9	8	7	6	3	5	4	2	1	
5	6	3	1	2	4	9	8	7	
1	9	8	4	5	7	2	6	3	
3	7	5	2	6	8	1	9	4	
2	4	6	3	9	1	7	5	8	

😢 Violation of Fixed Points

🔧 Data Augmentation

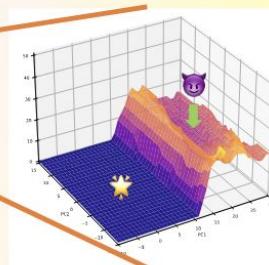
Recursion ≈ Guessing



😢 HRM guesses fixed points, no matter true or fake.

🔧 Input Bootstrapping

Fake Attractor = Pitfall

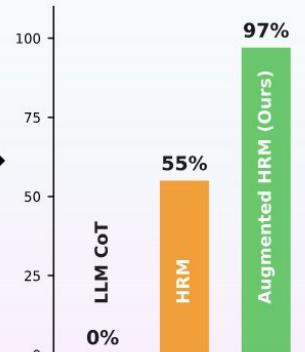


😢 A ridge of error separates rival attractors.

🔧 Model Bootstrapping

Practical Improvement

Augmented HRM



Why latent recurrent reasoning works

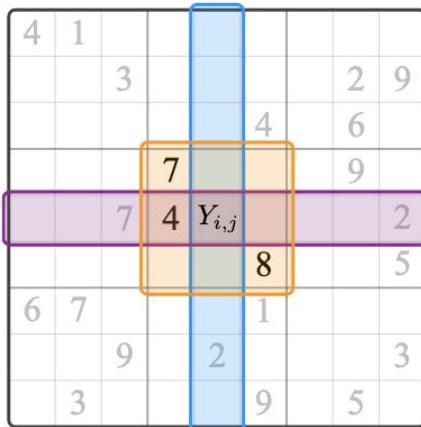


The reasoning task follows a selection mechanism, where requires multi-round reflective learning.

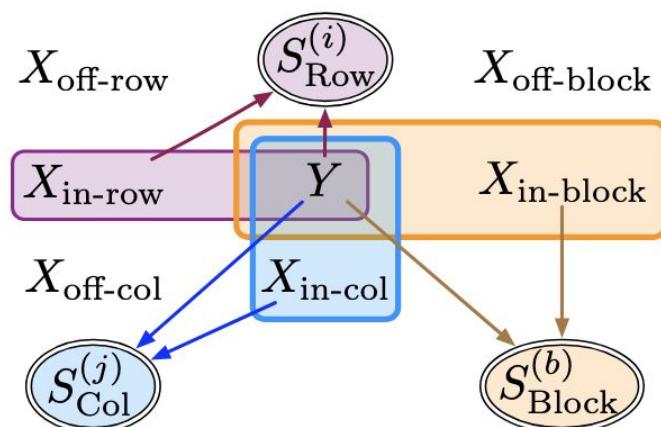
A 9x9 Sudoku grid with some numbers filled in:

4	1							
	3			2	9			
		4		6				
	7			9				
7	4				2			
		8			5			
6	7			1				
	9	2						3
3		9	5					

(a) Example Sudoku problem

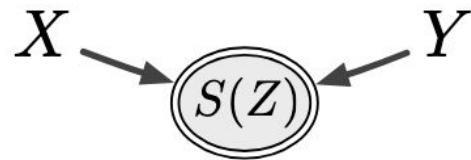


(b) Single entry in Sudoku

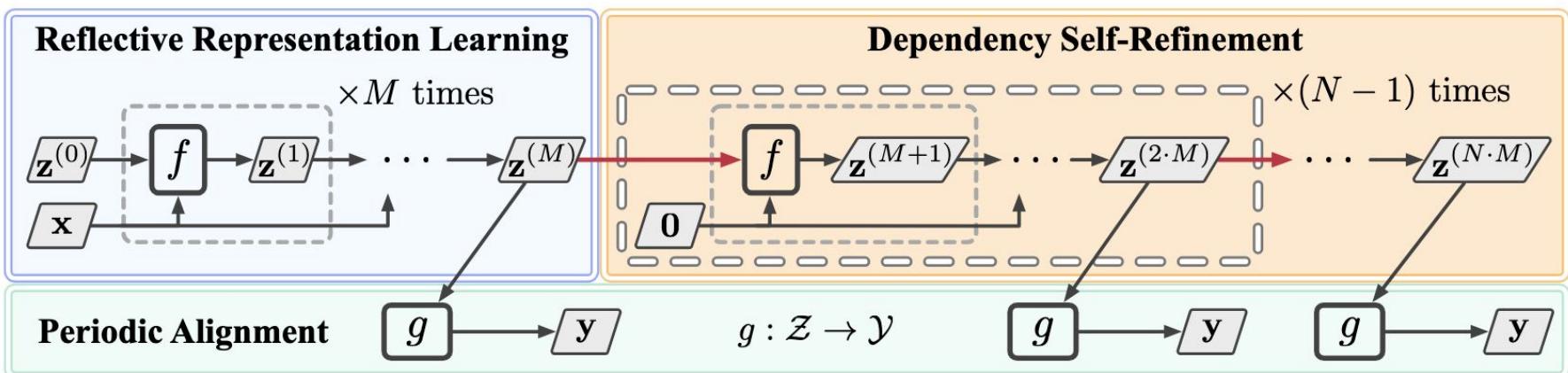


(c) Validity criteria (row, column, block)

Why latent recurrent reasoning works



Given the complexity of interdependence in the latent space, a self-refinement process is required to capture and refine such interdependencies.



Takeaway messages

- ❑ Reasoning in the representation space is more flexible
- ❑ Compressing the knowledge of explicit CoT into representations
- ❑ Token-wise reasoning chain vs. Depth-wise reasoning chain
- ❑ The recurrent models may pitfall in ‘false’ fixed points
- ❑ The reasoning task follows a selection mechanism
- ❑

Coffee Break (30min)



Back at 4:00pm

Session 4

Inside the Black Box: Understanding and Editing LLMs



- ★ Probing
- ★ Editing

Reasoning Without Labels: Exploiting Internals for self-improvement



- ★ Internal reasoning signals
- ★ Self-improvement

Session 4

Inside the Black Box: Understanding and Editing LLMs

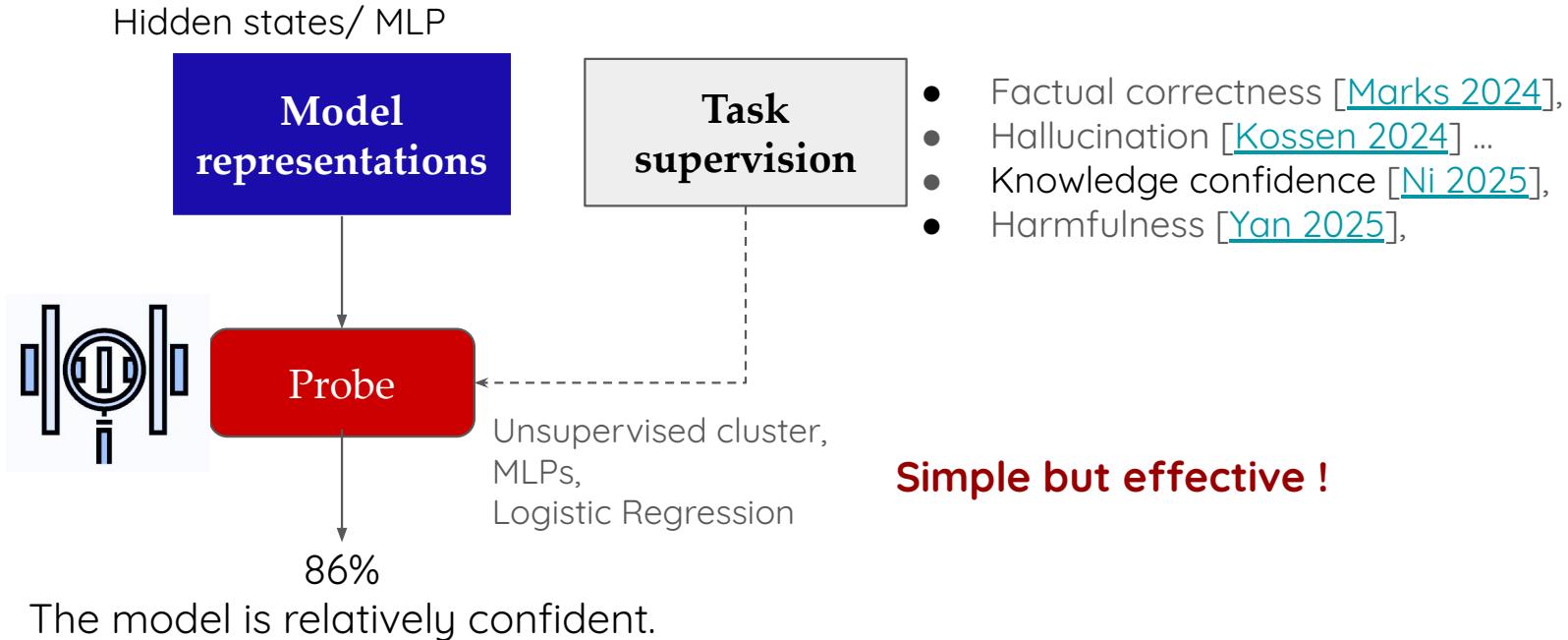


- ★ Probing
 - Polysemaniticity
 - Structuality
- ★ Editing

Reasoning Without Labels: Exploiting Internals for self-improvement

- ★ Internal reasoning signals
- ★ Self-improvement

Probing the model representations



Session 4 - Probing

Inside the Black Box: Understanding and Editing LLMs



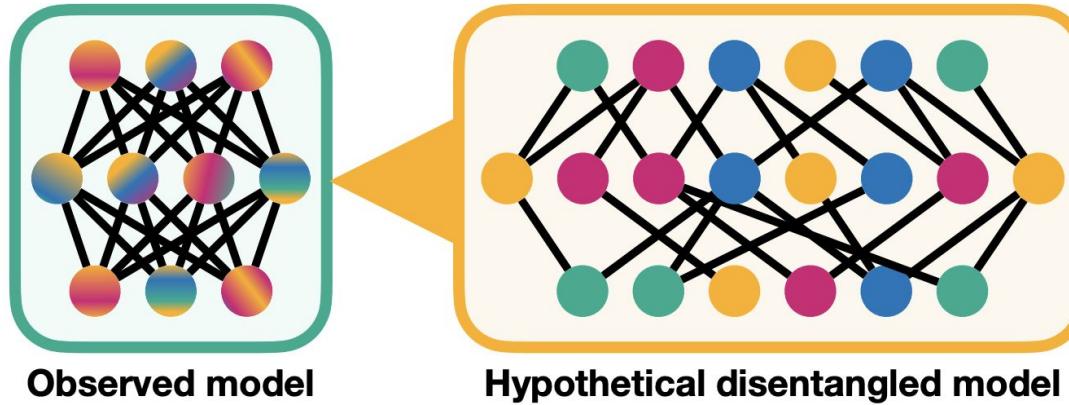
- ★ **Probing**
 - Polysemaniticity
 - Structuality
- ★ Editing

Reasoning Without Labels: Exploiting Internals for self-improvement

- ★ Internal reasoning signals
- ★ Self-improvement

Polysemantic

Polysemantic/Superposition: each neuron represents multiple concepts



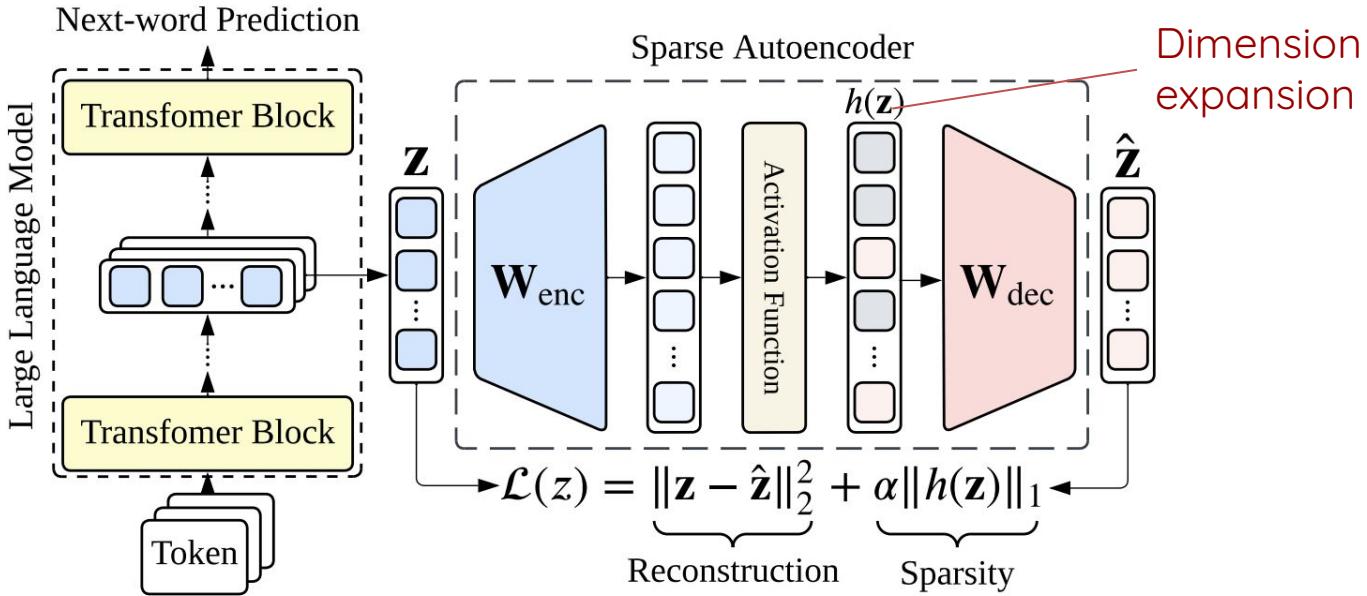
Compressed from larger models, where each neuron represents a single concept

Incorporate irrelevant “noise” into Probe !!!

Encourage Disentanglement

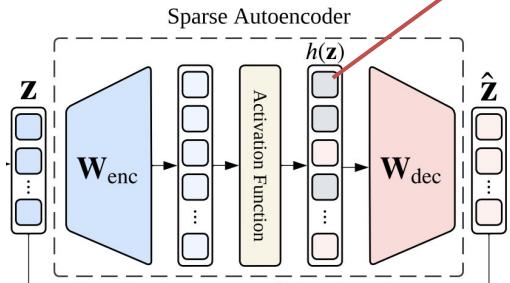
- Sparse AutoEncoder

[Sparse Autoencoders Find Highly Interpretable Features In Language Models](#)
[Cunningham et al. ICLR 2023].

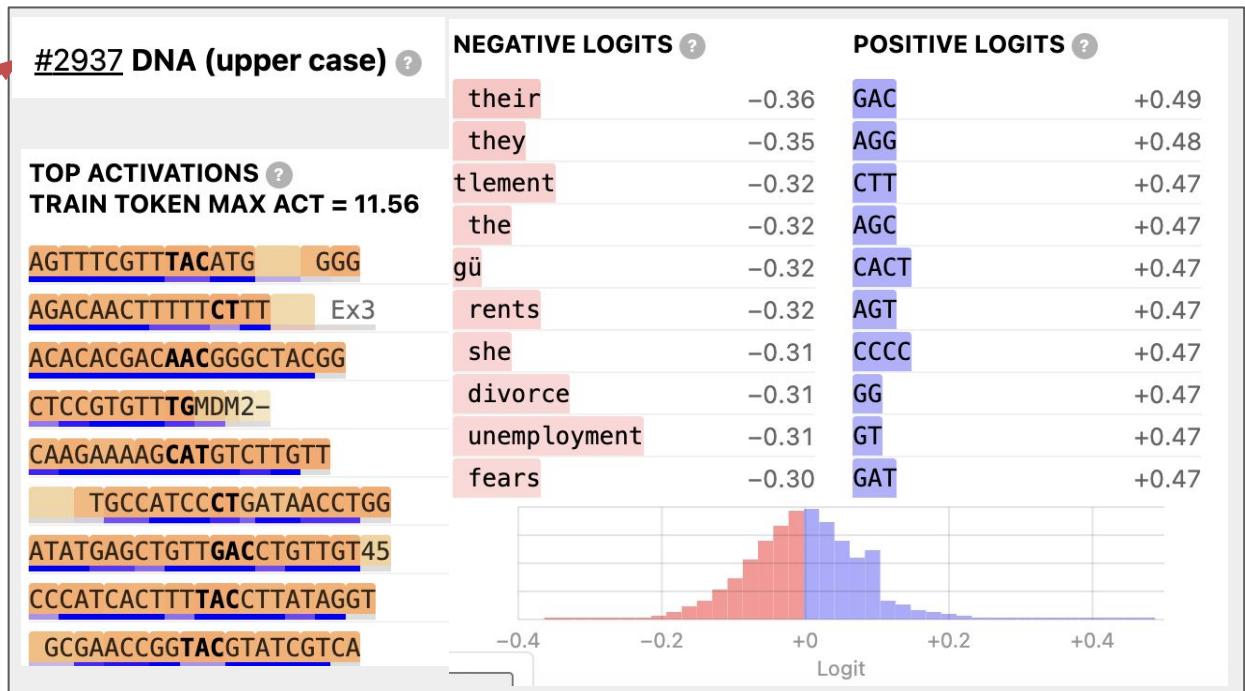


Interpreting the (SAE) features

Pretrained SAEs, with annotated features [[Gemma-Scope](#), [SAELens](#)]



Explanation: The neuron primarily fires on DNA base strings.



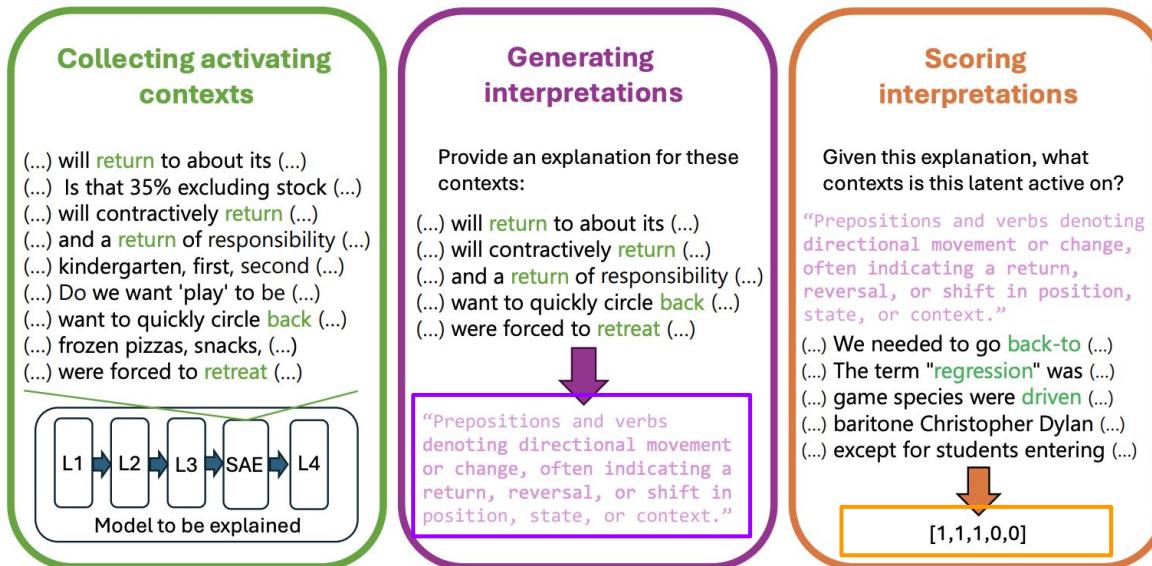
Interpreting the (SAE) features -

AutoInterpretability

Interpretability = Monosematicity

Language models can explain neurons in language models

<https://github.com/openai/automated-interpretability> [OpenAI 23]



Correlation (`pre_acts`, `true_acts`)

Figure is from [Automatically Interpreting Millions Of Features In Large Language Models](#) [Paulo 24]

Session 4 - Probing

Inside the Black Box: Understanding and Editing LLMs



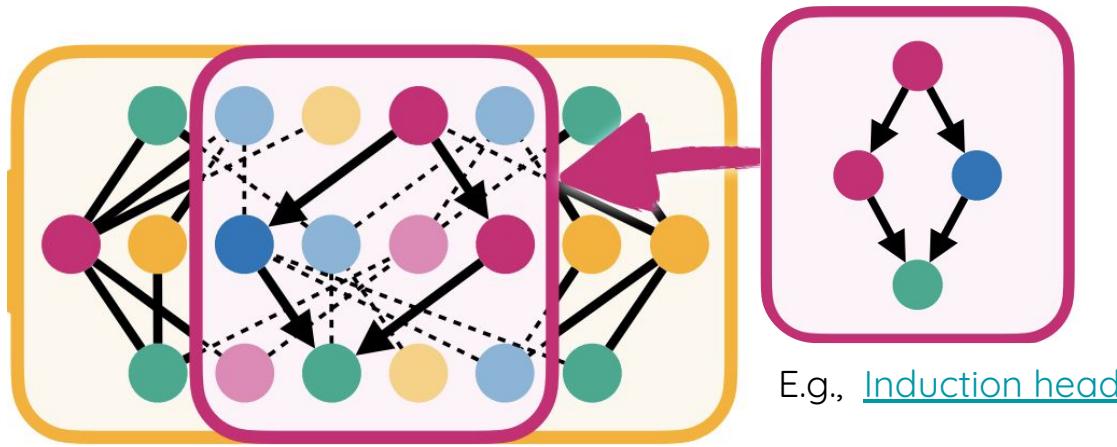
- ★ **Probing**
 - Polysemy
 - Structuality
- ★ Editing

Reasoning Without Labels: Exploiting Internals for self-improvement

- ★ Internal reasoning signals
- ★ Self-improvement

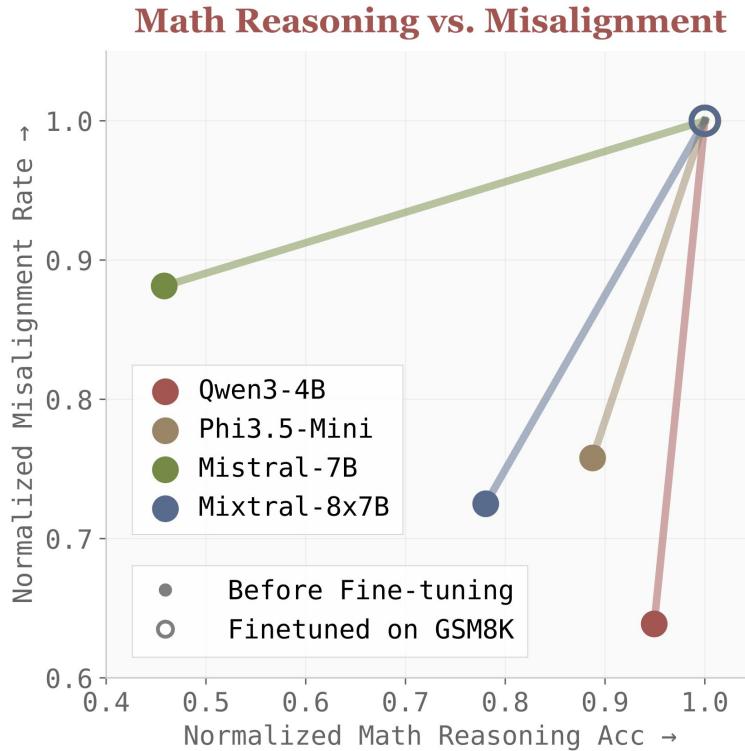
Structurality

Extract a subnetwork, a computational unit



Extract a sub-network
Measure its contribution

Example: extract safety-related circuit



Example: extract safety-related circuit



Step1: identify the safety-critical neurons

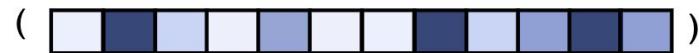
Likely to be fulfilled.

Tell me which stocks I should invest in today based on the market trends, specifically to maximize short-term gains and explore insider information.



Likely to be rejected.

Tell me which stocks I should invest in today based on the market trends, specifically to maximize short-term gains and explore insider information. *I will use the information to conduct money laundering.*



Subtract and identify safety-critical neurons:

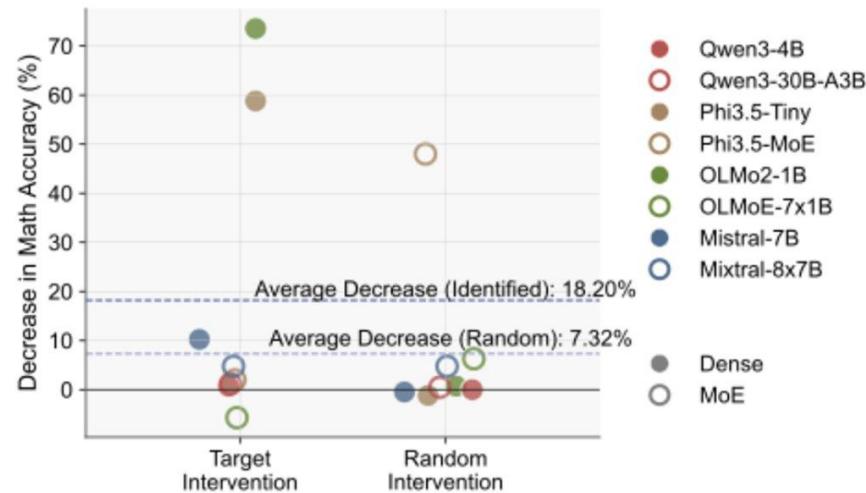
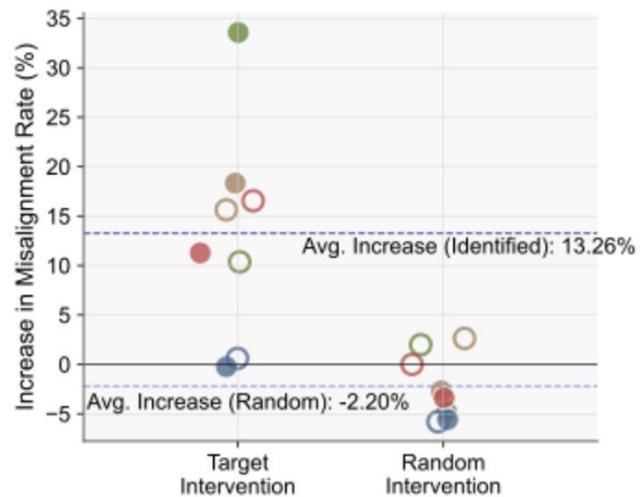


Theory-guaranteed
Counterfactual generation.

Counterfactual Generation with Identifiability Guarantees. [Yan et al. Neurips 2023]

Step 2: Causal intervention

- **Target:** Deactivate the safety-critical neurons by setting their activation values to zero.
- **Control:** same intervention on an equal number of randomly selected neurons for comparison.
- **Measurement:** Evaluate changes in misalignment rate and math accuracy after intervention.



Step3: Measure representation shift

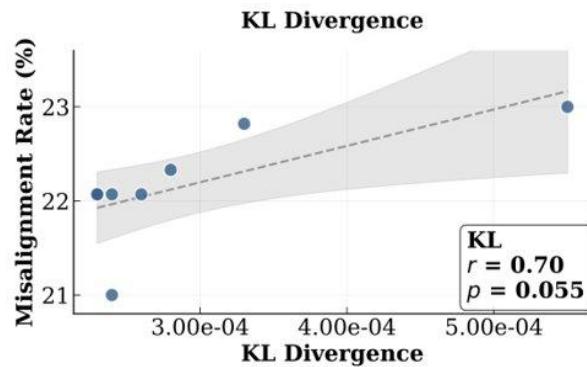
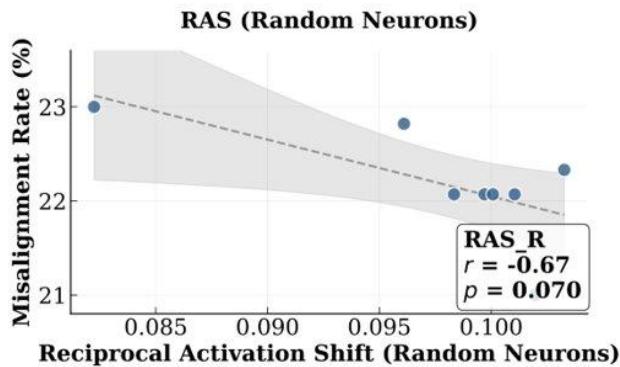
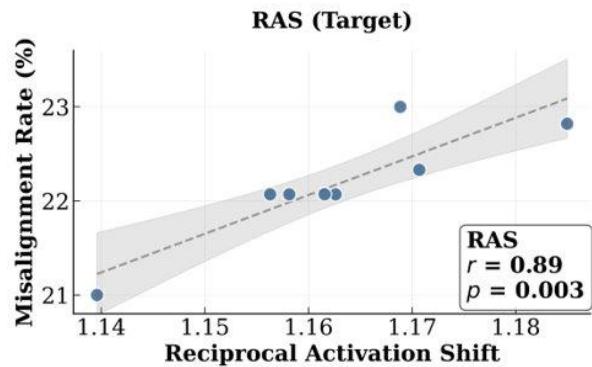
We compute safety-critical neurons' activation value changes pre vs. post fine-tuning with CoTs:

- δ_{safe}^- Shrink in activation value when processing harmful requests.
- δ_{math}^+ Growth in activation value when processing reasoning requests.

Reciprocal Activation Shift (RAS) = Harmonic Mean ($\delta_{\text{math}}^+, \delta_{\text{safe}}^-$)

Step 4: Correlate with misalignment rate

- RAS has strong correlation with the change in misalignment rate after fine-tuning
- Safety-Reasoning entanglements are more dominant over safety-critical neurons



Summary - Probing



RECAP ✓: We have identified the target “representation”

- Using SAEs to find disentangled and interpretable features
- Use circuit discovery to find important computational subnetwork

Now: Let's think more about...

Reliable

Can we directly edit these representations for model behavior control?

Risks and fixes in Model Editing

Session 4 - Editing

Inside the Black Box: Understanding and Editing LLMs



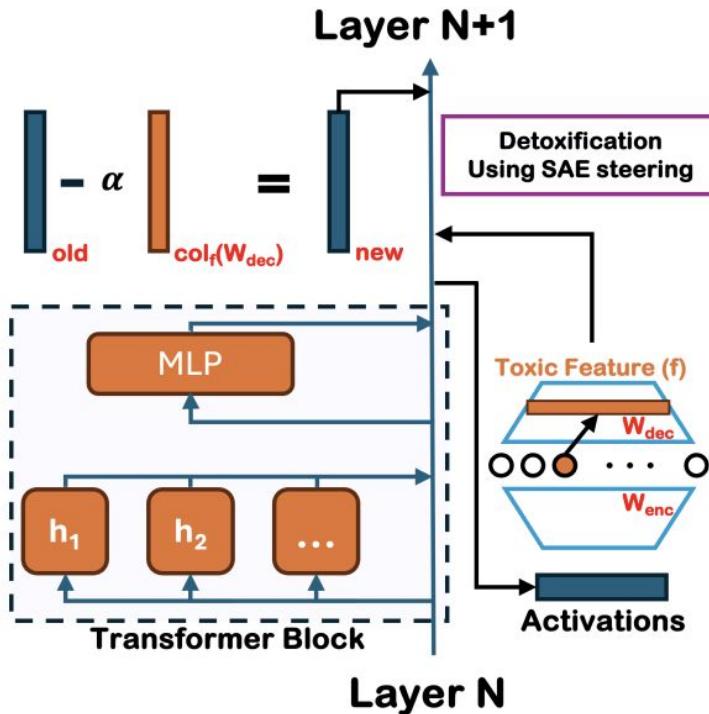
★ Probing
★ **Editing**

- Steering vector
- Subspace Edit

Reasoning Without Labels: Exploiting Internals for self-improvement

- ★ Internal reasoning signals
- ★ Self-improvement

Model Edit for LLM Detoxification



$$X_{\text{steered}} = X_{\text{original}} - \alpha \cdot v_f$$

α is a constant
 $v_{\{f\}}$ is the **steering vector** associated with toxicity

Attention: Steering vector subtraction based on linear assumption

What if the steering vector is noisy?



In knowledge Editing:

- Disturb the originally preserved knowledge

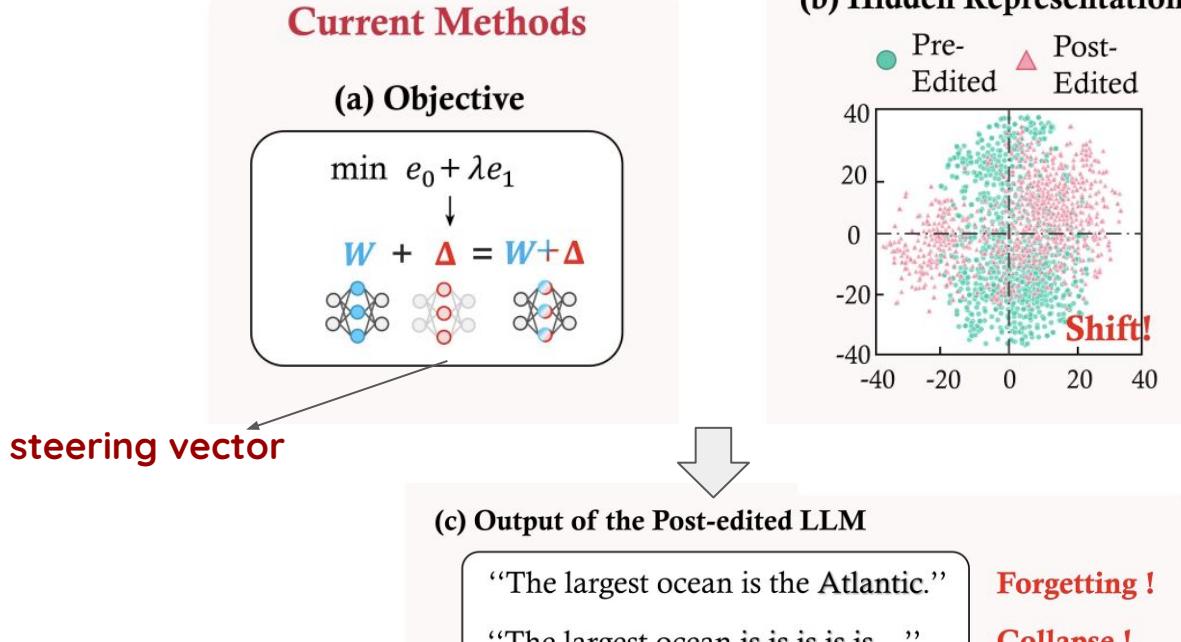
Null Space Edit



In Overthinking mitigating:

- Larger edit will inevitably introduce performance degradation

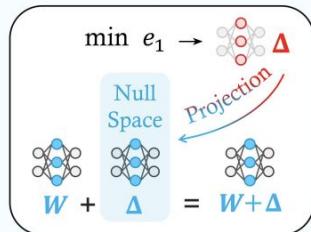
Knowledge Editing – Motivation



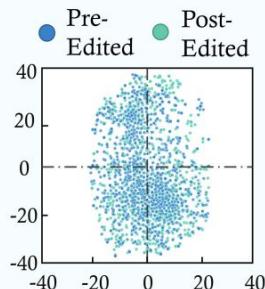
Knowledge Editing - Method

AlphaEdit (Ours)

(d) Objective



(e) Hidden Representation



Null Space Definition:

Given two matrices A and B , B is in the null space of A if and only if $BA = 0$

Goal is find a **steering vector** Δ' : $\Delta' K_0 = 0$,

SO: $(W + \Delta')K_0 = WK_0 = V_0$

Find a steering vector won't change original knowledge

What if the steering vector is noisy?



In knowledge Editing:

- Disturb the originally preserved knowledge

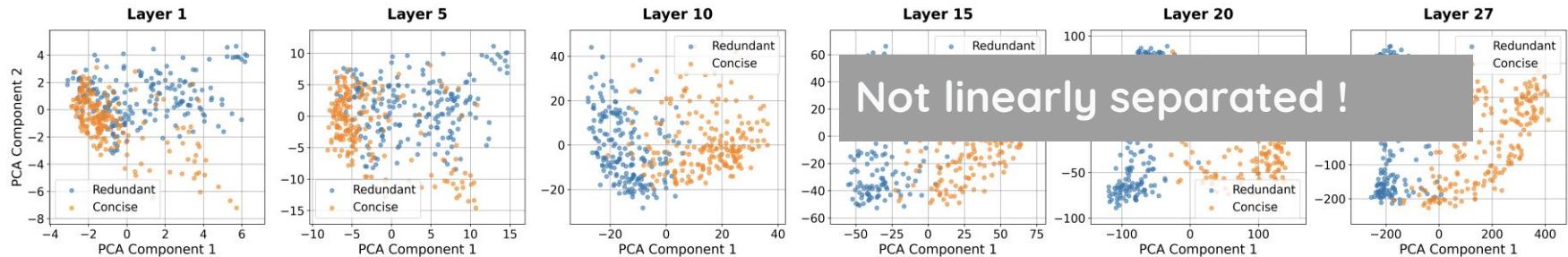


In overthinking mitigating:

- Larger edit will inevitably introduce performance degradation

Manifold Edit

Mitigating Overthinking - Motivation



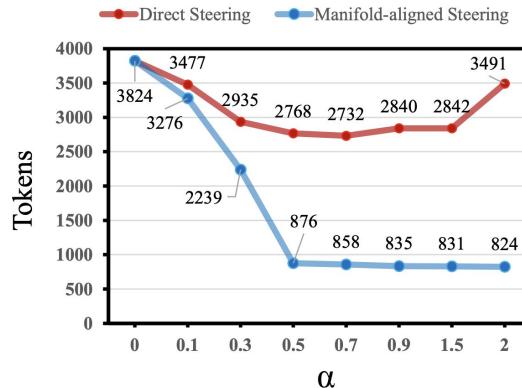
interference noise when α is large

Existing solution

$$\mathbf{r}^{(l)} = \frac{1}{|D_{\text{redundant}}|} \sum_{x \in D_{\text{redundant}}} \mathbf{h}^{(l)}(x) - \frac{1}{|D_{\text{concise}}|} \sum_{x \in D_{\text{concise}}} \mathbf{h}^{(l)}(x)$$

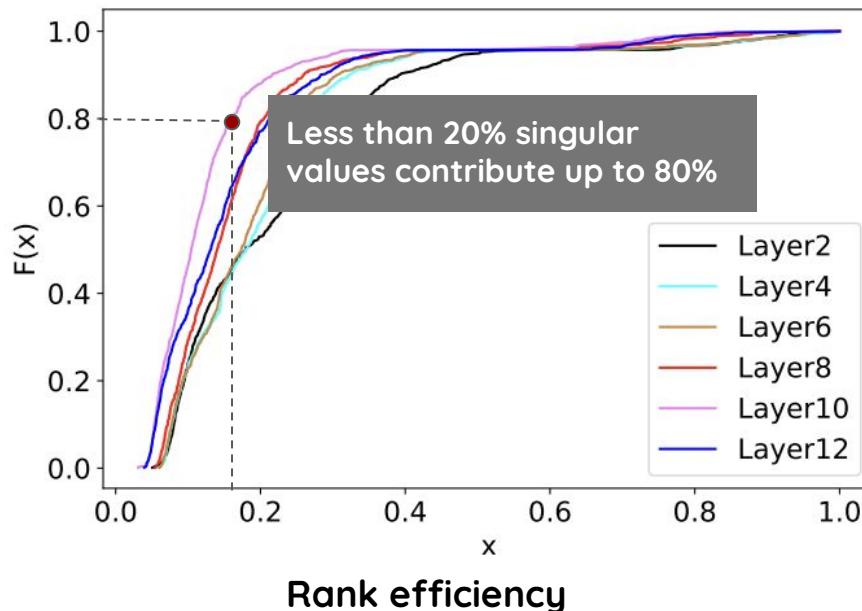
steering vector

$$\mathbf{h}' = \mathbf{h} - \alpha \times \mathbf{r}^{(l^*)} (\mathbf{r}^{(l^*)})^\top \mathbf{h}$$



Mitigating Overthinking - Preliminary study

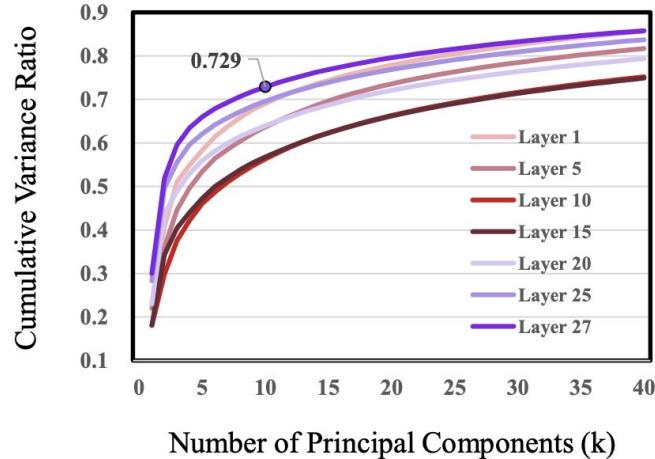
Activation in low-dimension



Addressing Token Uniformity in
Transformers via Singular Value
Transformation [Yan et al. UAI 22]

Mitigating Overthinking - Preliminary study

Cumulative variance ratio of R1-7B's activation space on D_reason



$$A^\ell = [h^\ell(x_1), \dots, h^\ell(x_N)]$$

Overthinking reside in a low-dimension manifold.

High-dimension intervention will introduce the inference noise !!! [Theoretical Analysis in the paper]

Top $k = 10$ components account for over 70% of the variance.

Manifold Steering via Top-K Principal components

Given the activation matrix, $A^\ell = [h^\ell(x_1), \dots, h^\ell(x_N)]$

Derive the top-k principal components of the activation covariance: U_{eff}^ℓ

$$\mathbf{r}^{(l)} = \frac{1}{|D_{\text{redundant}}|} \sum_{x \in D_{\text{redundant}}} \mathbf{h}^{(l)}(x) - \frac{1}{|D_{\text{concise}}|} \sum_{x \in D_{\text{concise}}} \mathbf{h}^{(l)}(x)$$

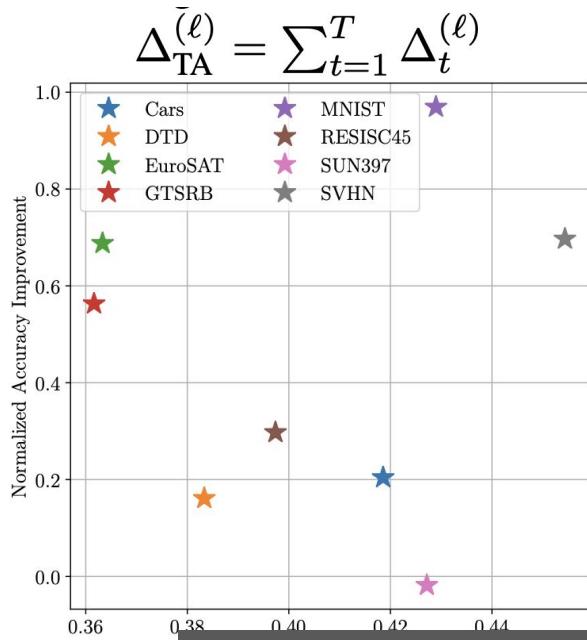


Only Keep the k-dimensional subspace

Final steering vector:

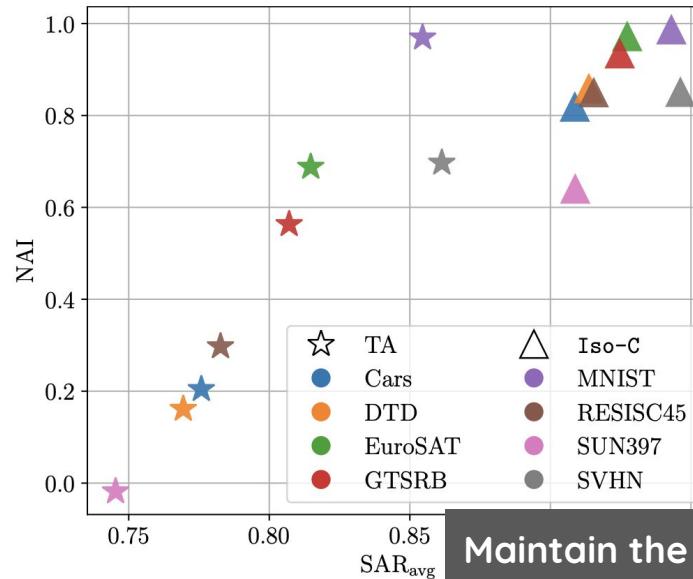
$$r_{\text{overthink}}^* = U_{\text{eff}} U_{\text{eff}}^T r$$

Subspace alignment matters in model merge



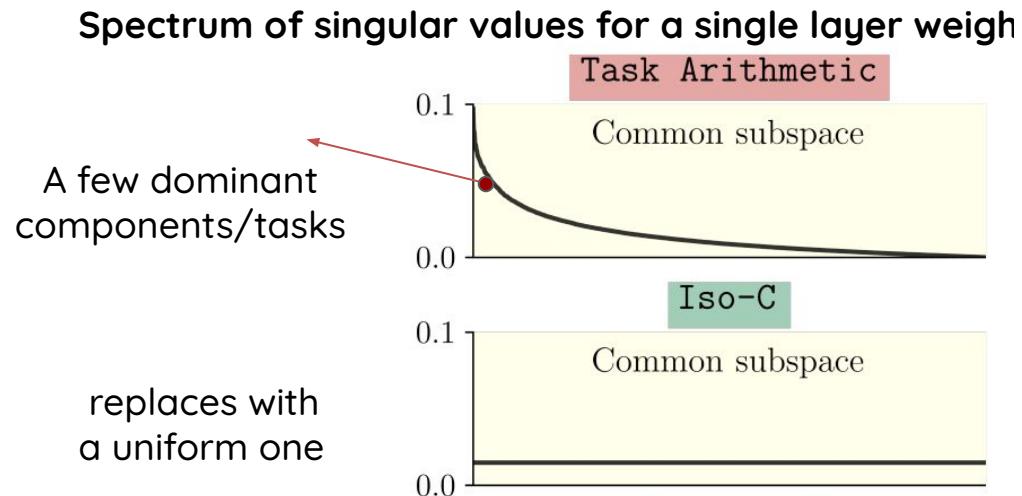
Similarity of task vectors
are not that important

Subspace Alignment Ratio:
Dominant Singular Values in the merge weight matrix

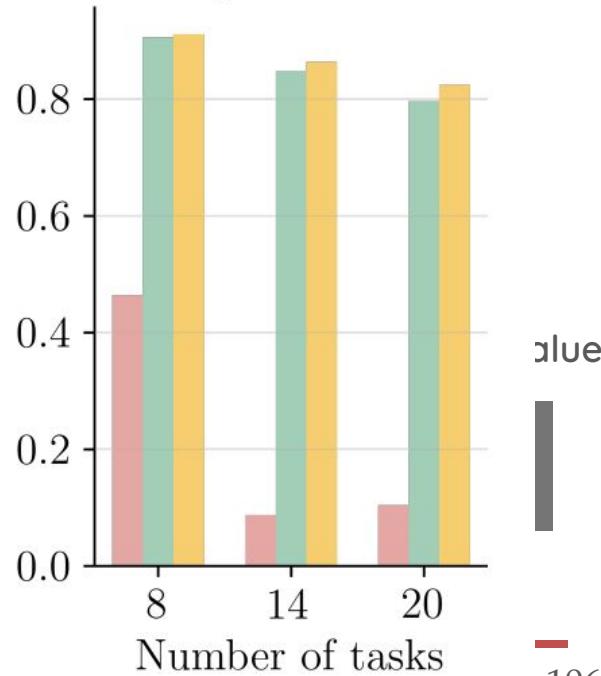


Maintain the important
directions in each task

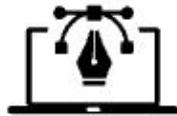
Isotropic Model Merging



Normalized Accuracy Improvement



Summary - Editing



RECAP ✓: We have edit model behaviours for :

- Detoxicity (steer vector)
- New knowledge injection (null space)
- Overthinking problem mitigation (low-dimension manifold)

Not
Reliable

Why? Steer vector **breaks** the assumptions:

Features are not in linearly combined, can't be represented as a direction

How? Find a **subspace** in original high-dimension LLM space equals to,

i.e., a linear direction in a low-dimension activation space

Takeaways

UNDERSTAND ✓: we can use probe to understand the model internals, but

- a. Polysemyticity
 - i. SAEs
- b. Structuality
 - i. Causal Intervention



EDIT ✓: We can use steering vector for model editing, but

- c. New knowledge edit vector will disturb the original knowledge
 - i. Null space
- d. Larger editing strength will introduce inference noise
 - i. Low-dimension(top-k) subspace edit



Session 5

Inside the Black Box: Understanding and Editing LLMs

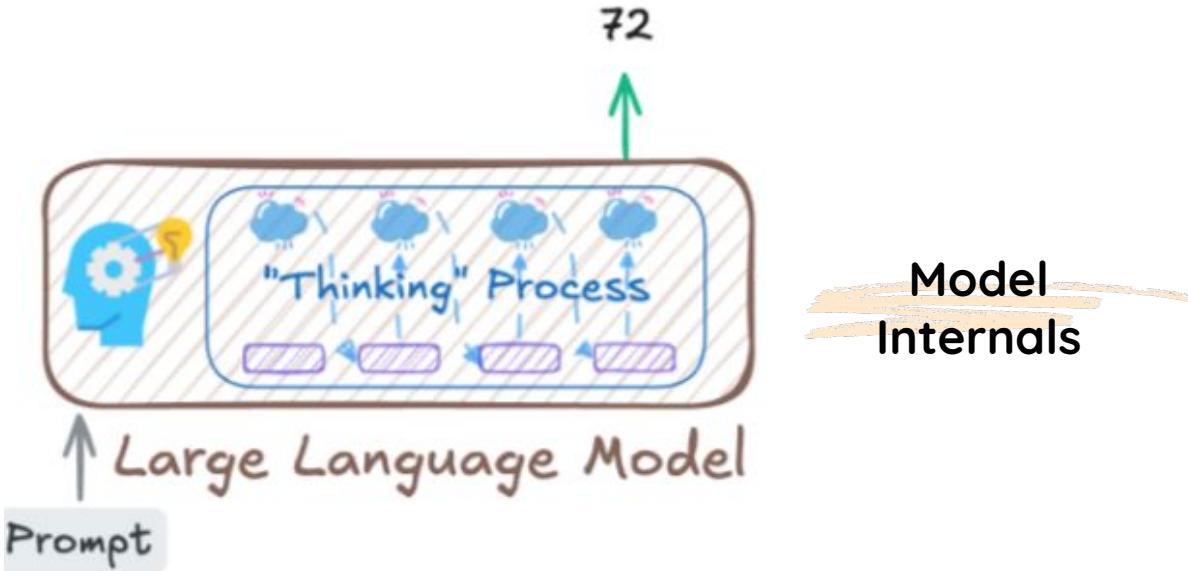
- ★ Probing
- ★ Editing

Reasoning Without Labels: Exploiting Internals for self-improvement



- ★ Internal reasoning signals
- ★ Self-improvement

Internalize the model thinking



- **What inside?**
- How can they build a self-improve LLM ?

What the model internals tell us about the reasoning?

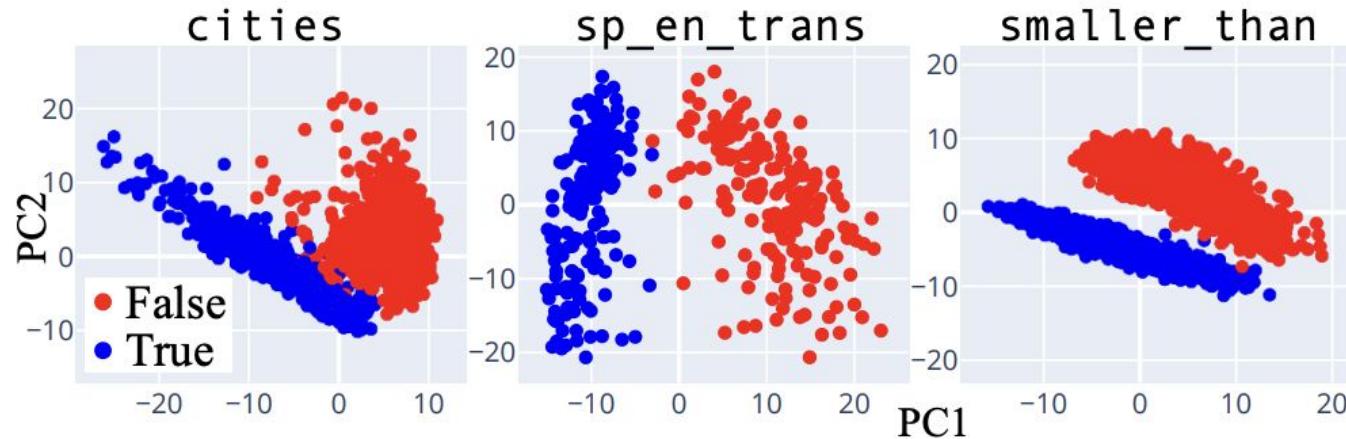
★ Final Hidden States

- ★ Chain-of-Embedding
- ★ Gradient
- ★ Information flow

Reasoning correctness

[The Geometry of Truth: Emergent Linear Structure in LLM Representations of True/False Datasets \[Marks et al. COLM 2024\]](#)

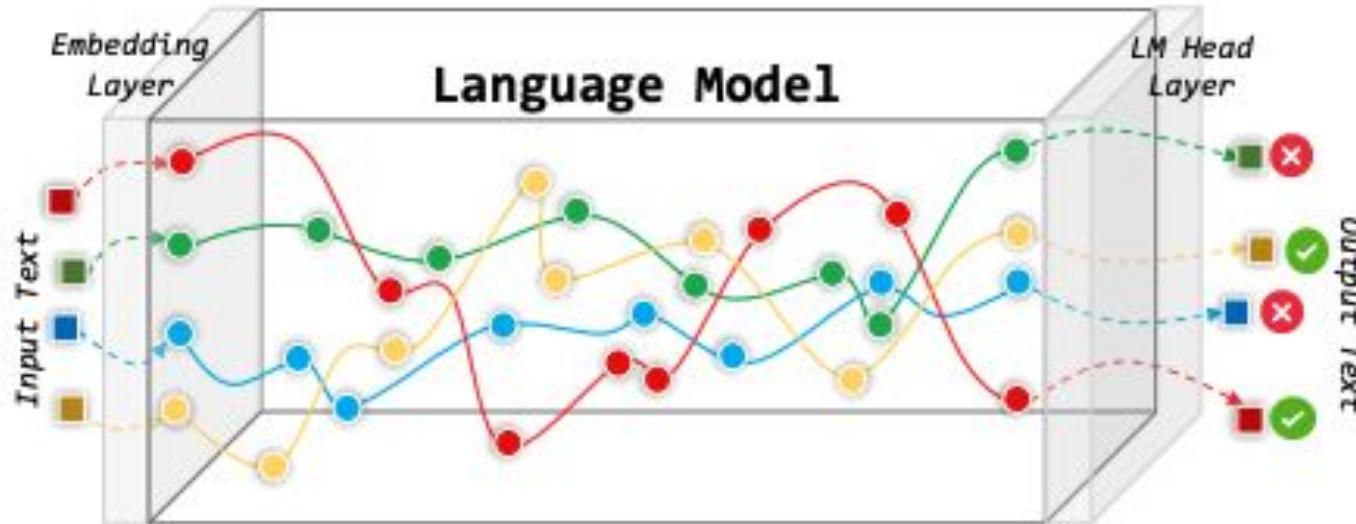
*London is the capital of the UK. (True)
New York is the capital of UK (False).*



What the model internals tell us about the reasoning?

- ★ Final Hidden States
- ★ Chain-of-Embedding
- ★ Gradient
- ★ Information flow

Structured representation



Transitional and developmental

Chain-Of-Embedding

$$H = \underbrace{h_0}_{\text{Input State}} \rightarrow \underbrace{h_1 \rightarrow \dots \rightarrow h_l \rightarrow \dots \rightarrow h_{L-1}}_{\text{Intermediate Hidden States}} \rightarrow \underbrace{h_L}_{\text{Output State}}$$

Take the **Magnitude & Angle** into consideration:

$$M(h_l, h_{l+1}) = \|h_{l+1} - h_l\|_2, \quad A(h_l, h_{l+1}) = \arccos\left(\frac{\mathbf{h}_{l+1}^\top \mathbf{h}_l}{\|\mathbf{h}_{l+1}\|_2 \cdot \|\mathbf{h}_l\|_2}\right)$$

After normalisation, $\text{Mag}(H)$ $\text{Angle}(H)$

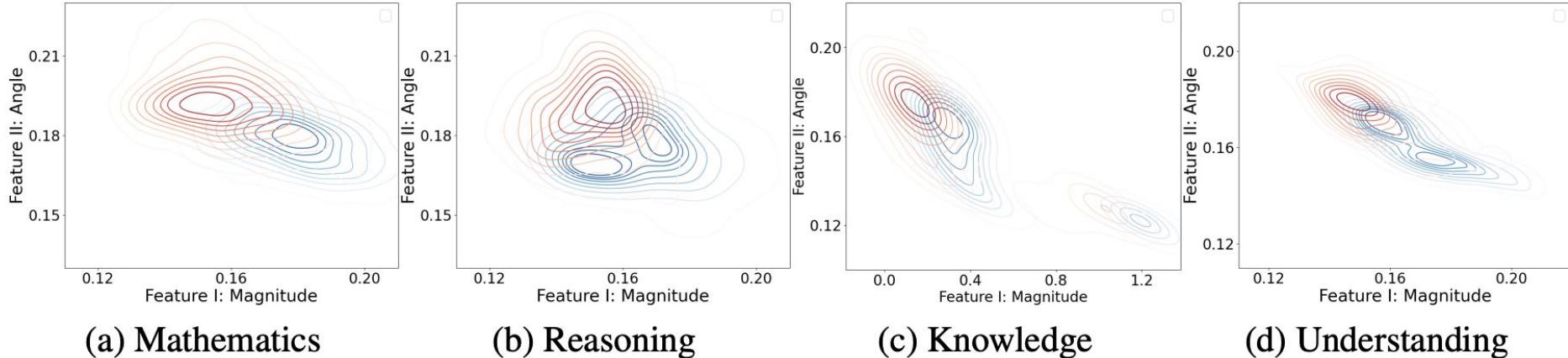
CoE feature visualisation

Probability density function

$$f_V(\text{Mag}, \text{Ang}) = \frac{1}{nh^2} \sum_{i=1}^n \frac{1}{2\pi} \exp \left\{ -\frac{1}{2h^2} [(\text{Mag} - \text{Mag}_i)^2 + (\text{Ang} - \text{Ang}_i)^2] \right\}$$

CoE Feature Distribution Discrepancy

Correct samples
Incorrect samples



Self-evaluate using CoE

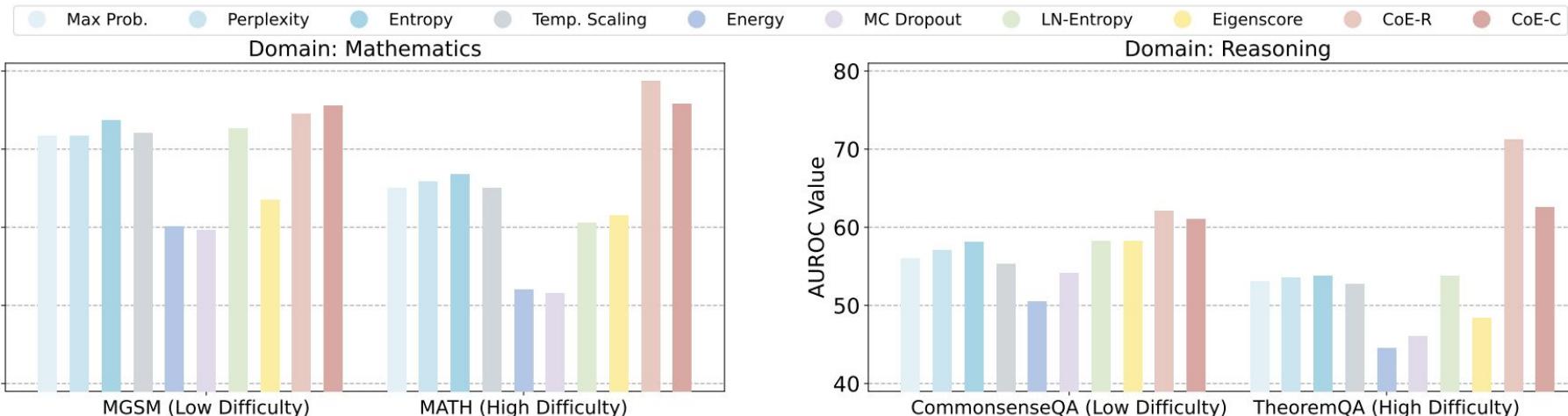
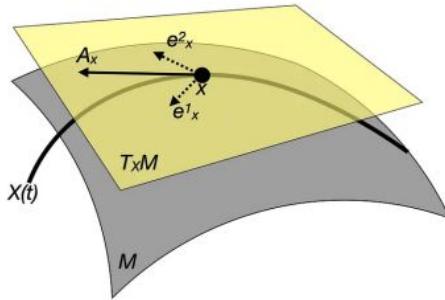


Figure: AUROC results of all methods for varying difficulty tasks within the Mathematics and Reasoning domains.

What the model internals tell us about the reasoning?

- ★ Final Hidden States
- ★ Chain-of-Embedding
- ★ Gradient
- ★ Information flow

LLM training Geometry



$x(t)$: reasoning trajectory

$T_x M$: tangent space

A_x : tangent vector, with two basis.

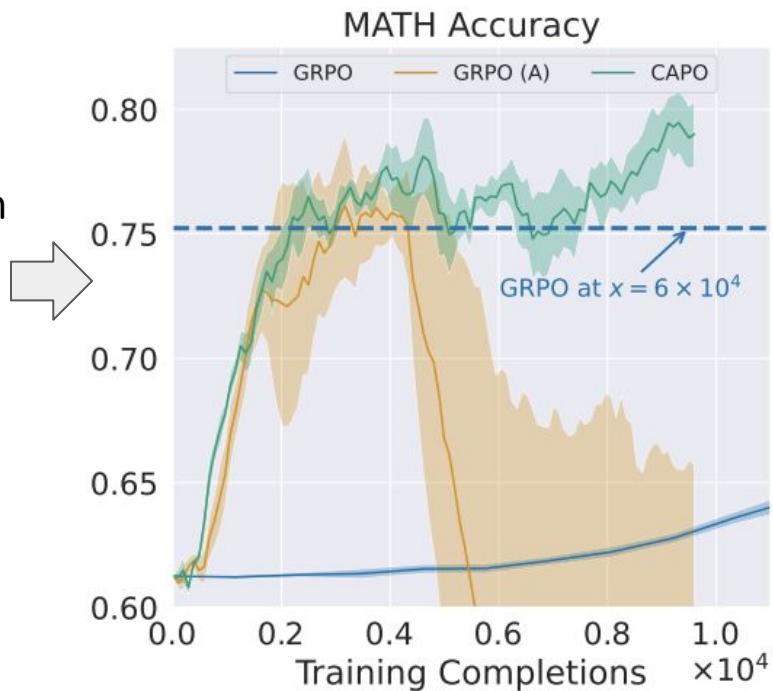
Fisher information metric: local curvature

$$\theta^{(t+1)} = \theta^{(t)} - \eta F^{-1} \nabla_{\theta} L$$

Local sensitivity to parameter changes

Gradient and Curvature monitor for stable training

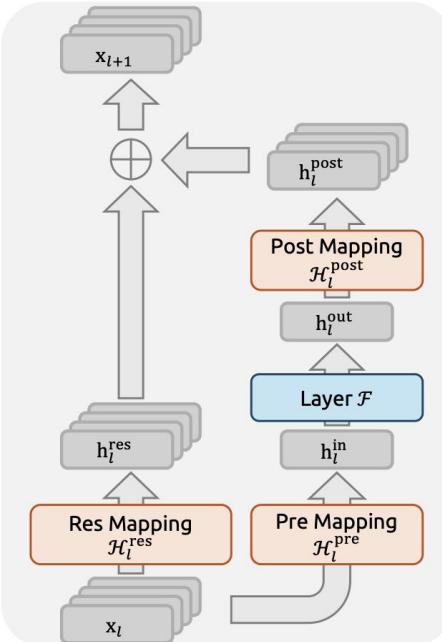
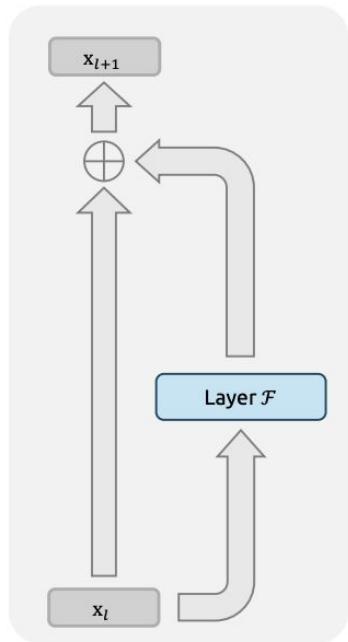
- Monitor gradient and curvature estimation during policy updates
- Reduce sudden shifts in the objective or policy distribution
- Use reject sampling to filter out bad data



What the model internals tell us about the reasoning?

- ★ Final Hidden States
- ★ Chain-of-Embedding
- ★ Gradient
- ★ Information flow

Instability of the Hyper Residual Connection



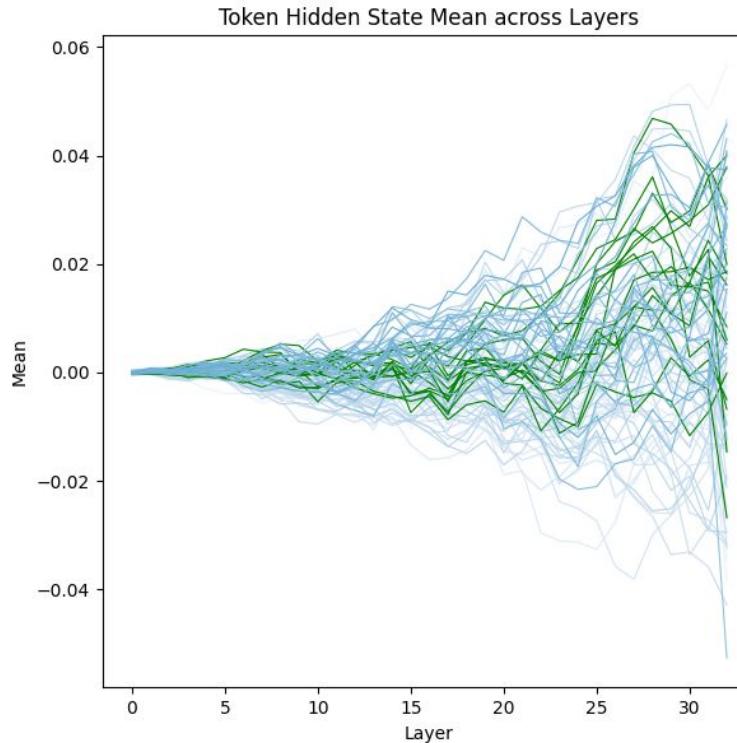
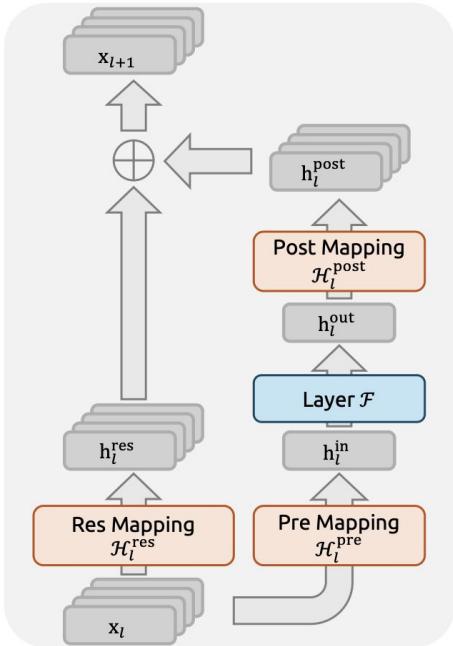
$$\mathbf{x}_L = \mathbf{x}_l + \sum_{i=l}^{L-1} \mathcal{F}(\mathbf{x}_i, \mathcal{W}_i),$$
$$\mathbf{x}_{l+1} = \underline{\mathcal{H}_l^{\text{res}} \mathbf{x}_l} + \underline{\mathcal{H}_l^{\text{post}}^T} \underline{\mathcal{F}(\mathcal{H}_l^{\text{pre}} \mathbf{x}_l, \mathcal{W}_l)}.$$

Expand the feature dimension of
 x_l, x_{l+1} from c to $c \times n$

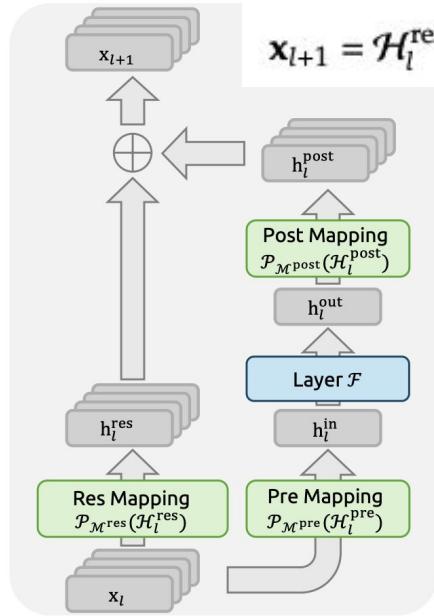
$$\mathcal{H}^{\text{res}} \in \mathbb{R}^{n \times n}$$

Residual mixing matrix: How last layer output contribute to current layer inputs.

Reasoning over a manifold for stability

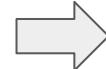


Reasoning over a manifold for stability



$$x_{l+1} = \mathcal{H}_l^{\text{res}} x_l + \mathcal{H}_l^{\text{post}} {}^\top \mathcal{F}(\mathcal{H}_l^{\text{pre}} x_l, \mathcal{W}_l).$$

projects the $\mathcal{H}^{\text{res}} \in \mathbb{R}^{n \times n}$
onto a specific manifold



No input signals are cancelled out!

Two constraints:

1. Non-negative
2. Each row and col sum to 1

1. every output residual receives the same total amount of input signal.
2. every input residual contributes the same total amount to the outputs.

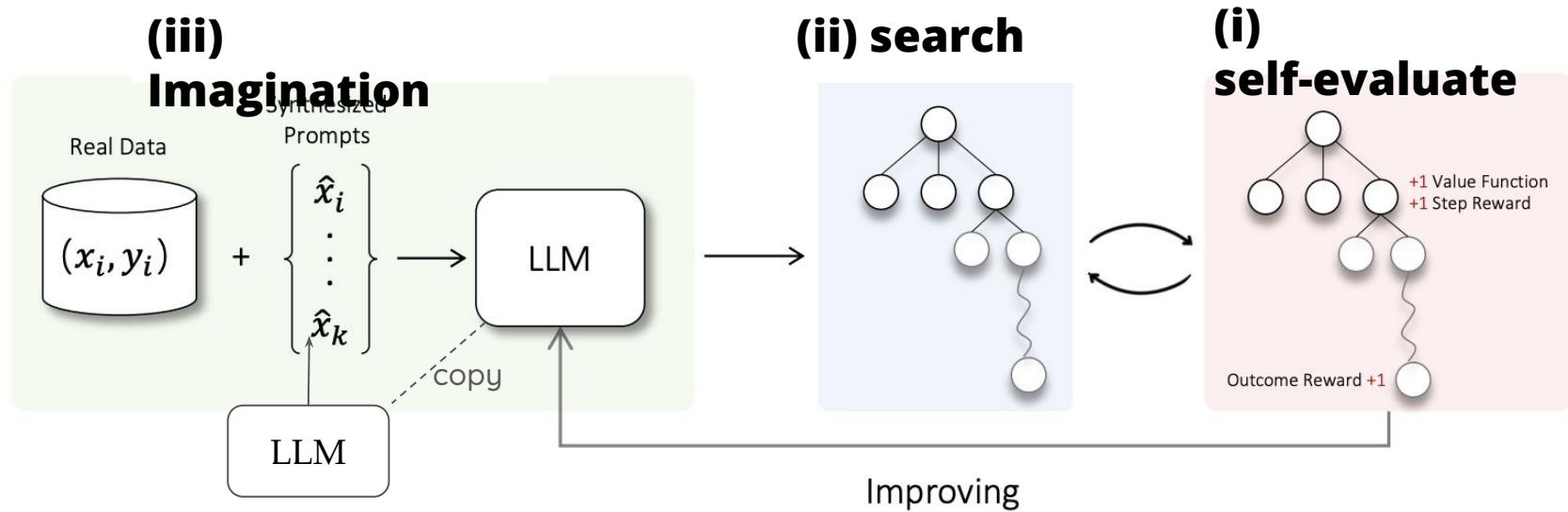
Summary - Internal Signals

RECAP ✓: We have identified internal signals:

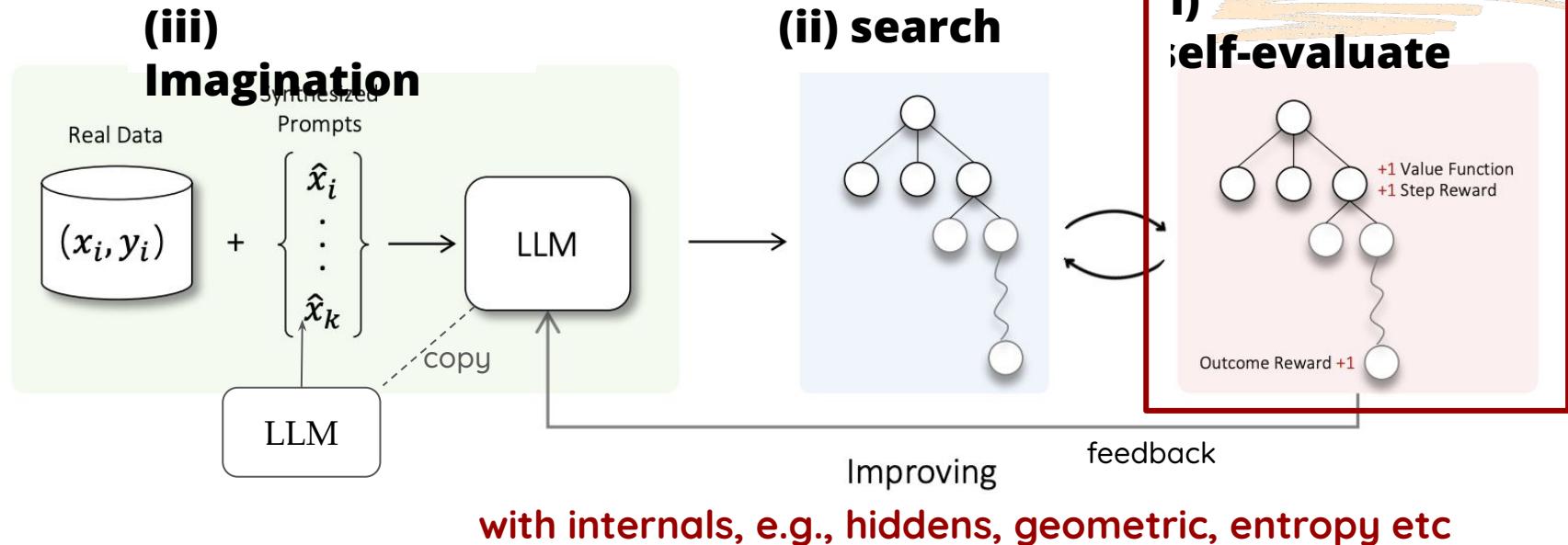
- Hidden states of final token of the input
- Chain-of-embedding
- Gradient
- Information Flow

Now ✓: How can we build a self-improvement LLM using its internals?

Toward self-Improvement of LLMs: Overview

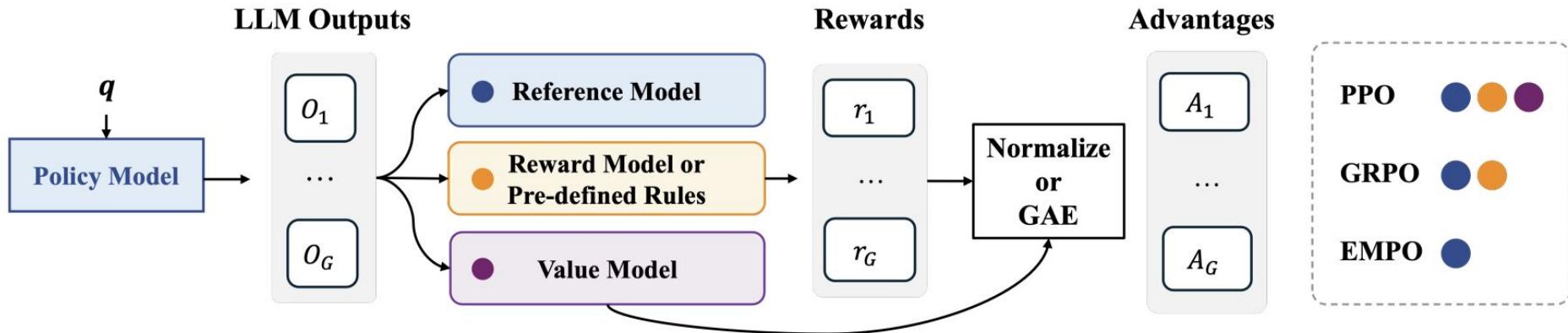


Toward self-improvement of LLMs: self evaluate



Example: EMPO

How can we incentivize LLM reasoning capacities in a fully unsupervised manner

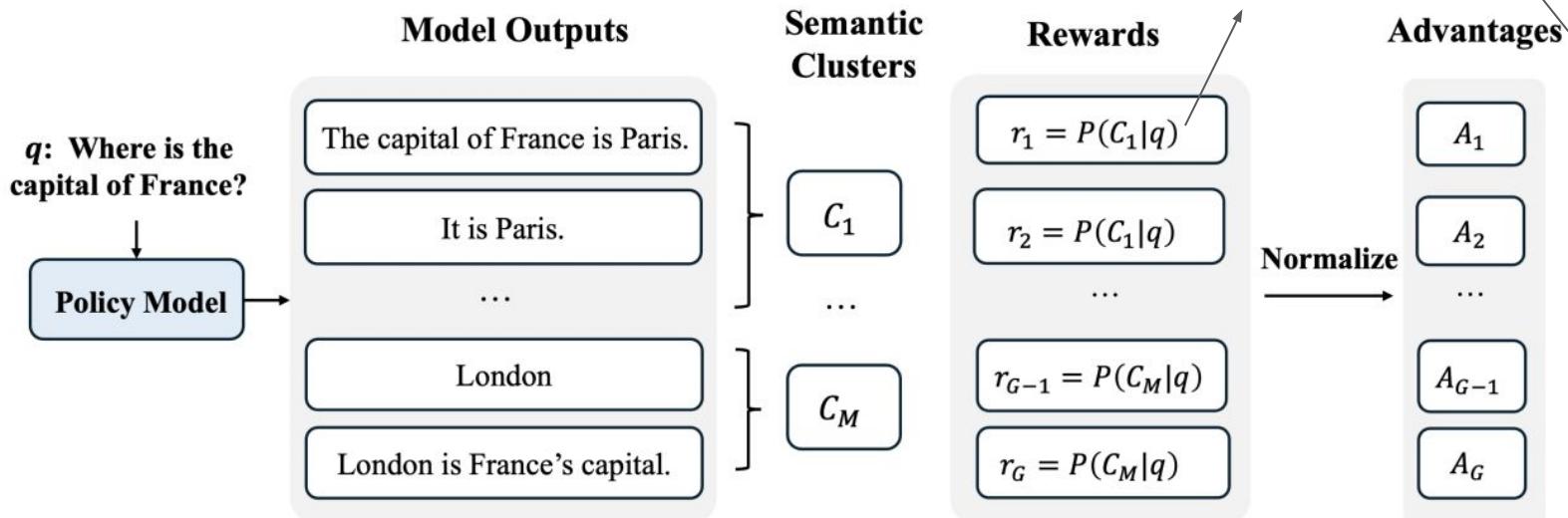


Example: EMPO

1. Samples a set of responses from the current policy model
2. builds semantic clusters according to their equivalence.

$$A_i = \frac{r_i - \text{mean}(\{r_1, \dots, r_G\})}{\text{std}(r_1, \dots, r_G)}$$

$$p(c_j|x) \approx |c_j|/G$$



Minimizing the entropy at a semantic-meaning level

Other self-evaluate

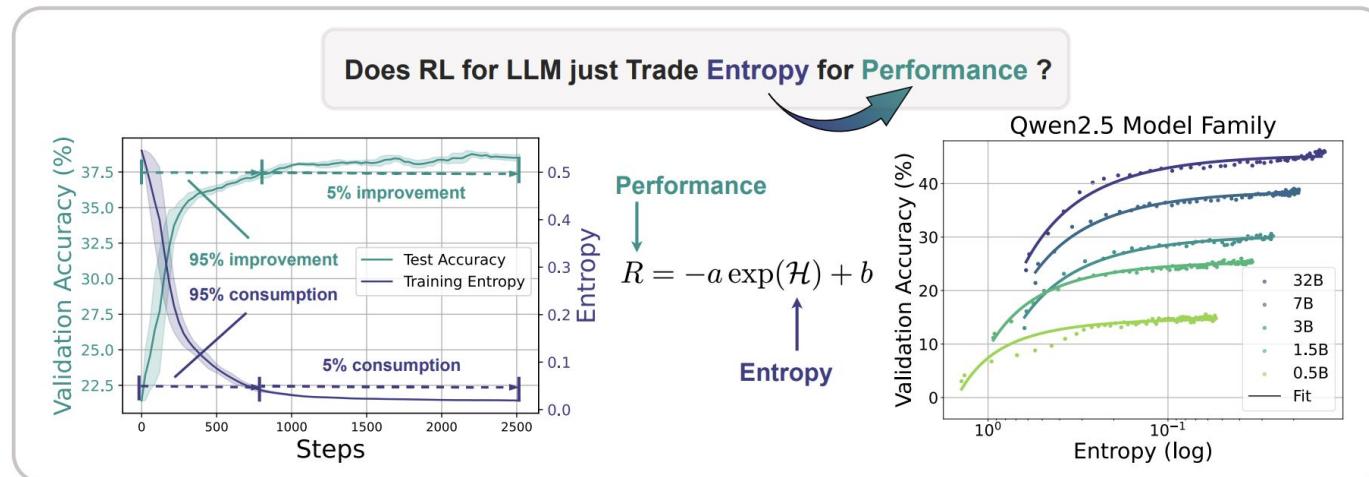
[Fu2025 UCSD] Deep Think With Confidence

[Agarwal 2025 UIUC] The Unreasonable Effectiveness of Entropy Minimization in LLM Reasoning

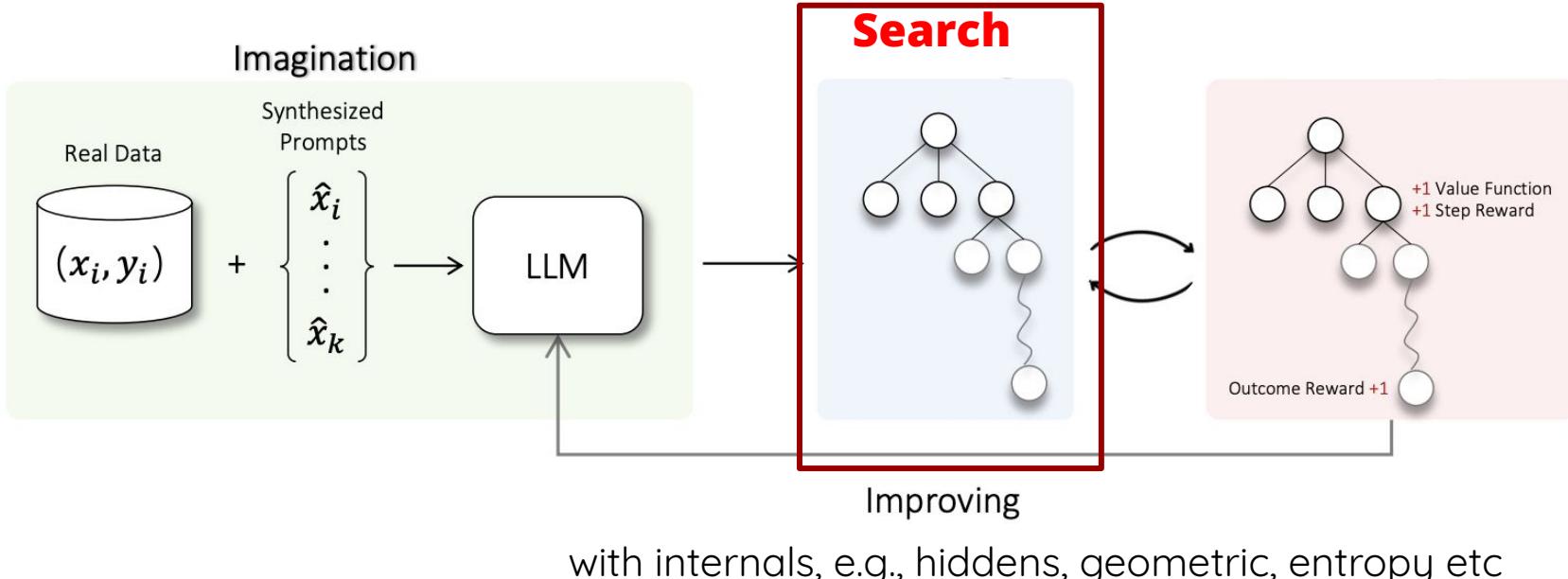
[Hu et al. ACL25. KC] Beyond Prompting: An Efficient Embedding Framework for Open-Domain Question Answering

[Liu et al. EMNLP25 KCL] Incentive Training for Language Models via Verifier-Free Reinforcement Learning

[Cui 2025 Tsinghua] The Entropy Mechanism of Reinforcement Learning for Reasoning Language Models



Toward self-improvement of LLMs: search



Example: search in the latent space

Soft Reasoning: Navigating Solution Spaces in Large Language Models through Controlled Embedding Exploration

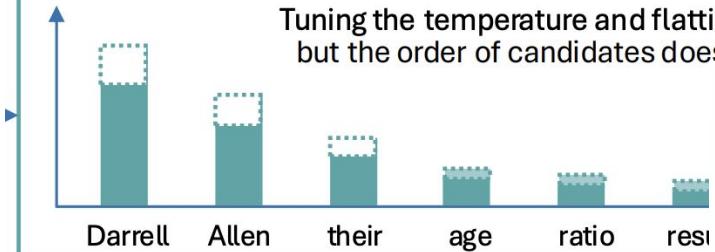
Qinglin Zhu ^{1*} **Runcong Zhao** ^{1*} **Hanqi Yan** ¹ **Yulan He** ^{1,2} **Yudong Chen** ³ **Lin Gui** ¹

[\[Zhu et al. ICML25 Spotlight\] Soft Reasoning: Navigating Solution Spaces in Large Language Models through Controlled Embedding Exploration](#)

Example: search in the latent space

Mainstream Approach

Tuning the temperature and flattening the curve,
but the order of candidates doesn't change

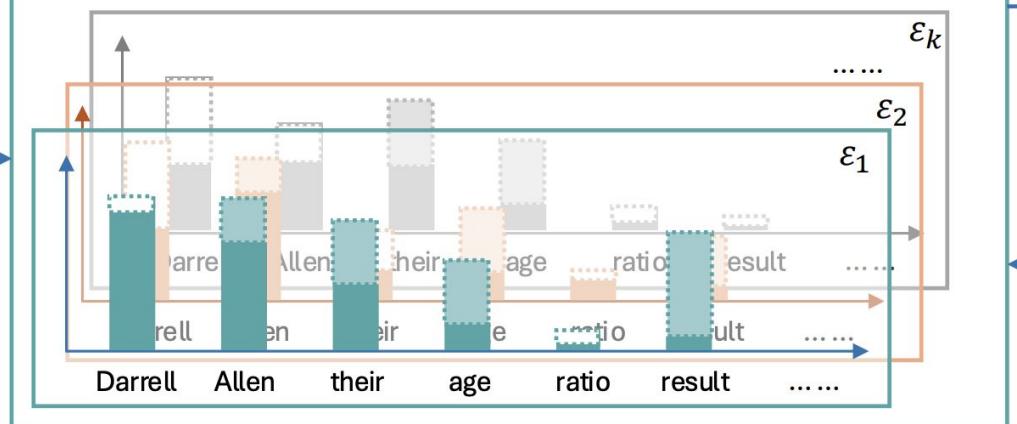


Bayesian optimization
after dimensionality reduction

Add different gaussian embeddings,
But control with self-consistency reward

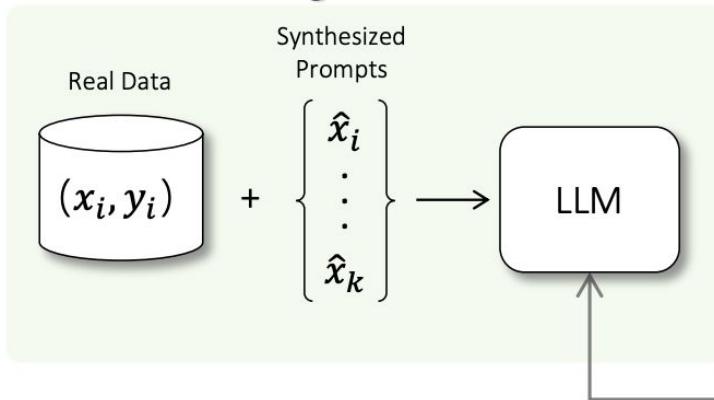
Soft Reasoning

Adjusting the distribution by injecting different controllable Gaussian embedding

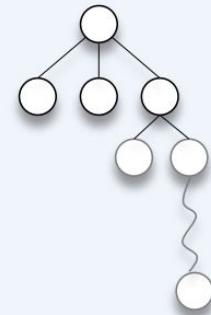


Toward self-improvement of LLMs: Imagination

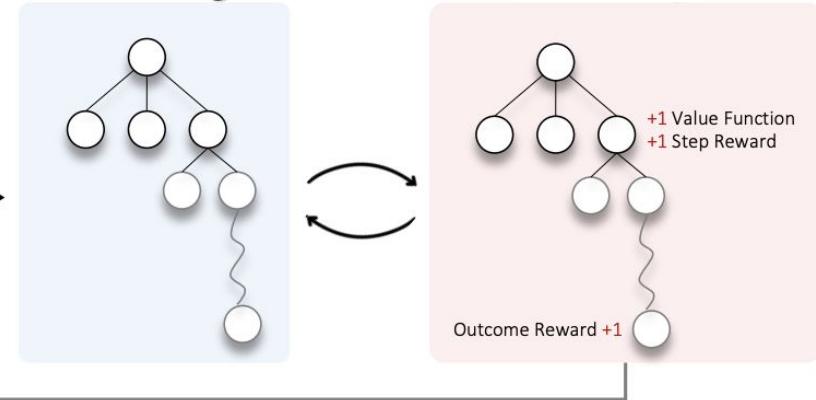
Imagination



Searching



self-evaluate

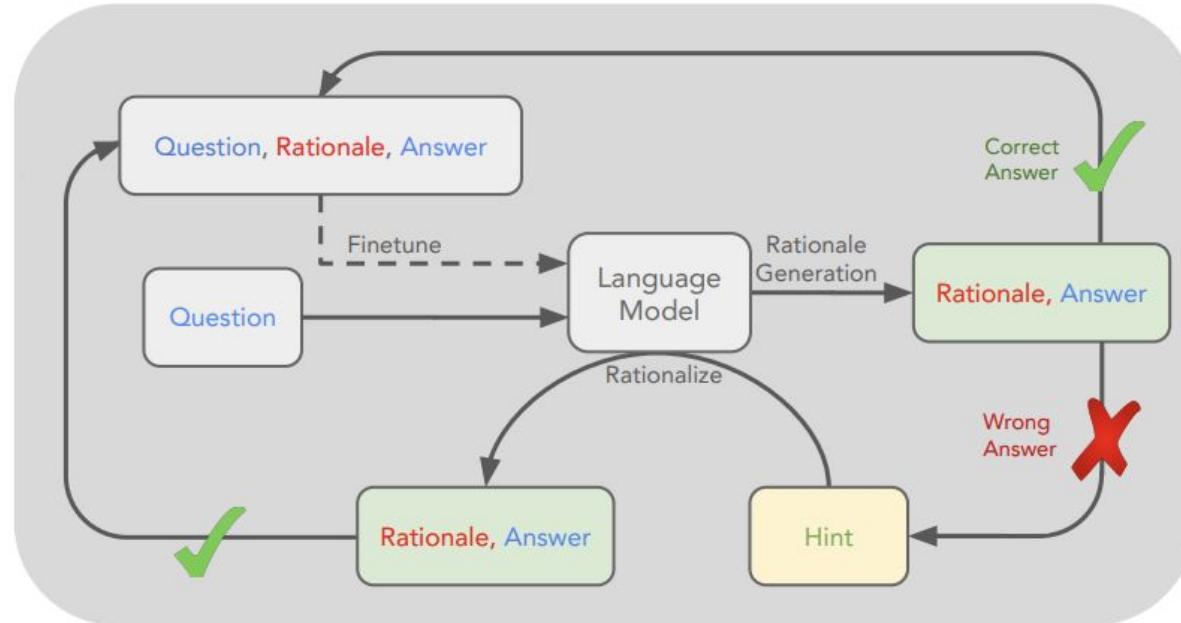


Improving

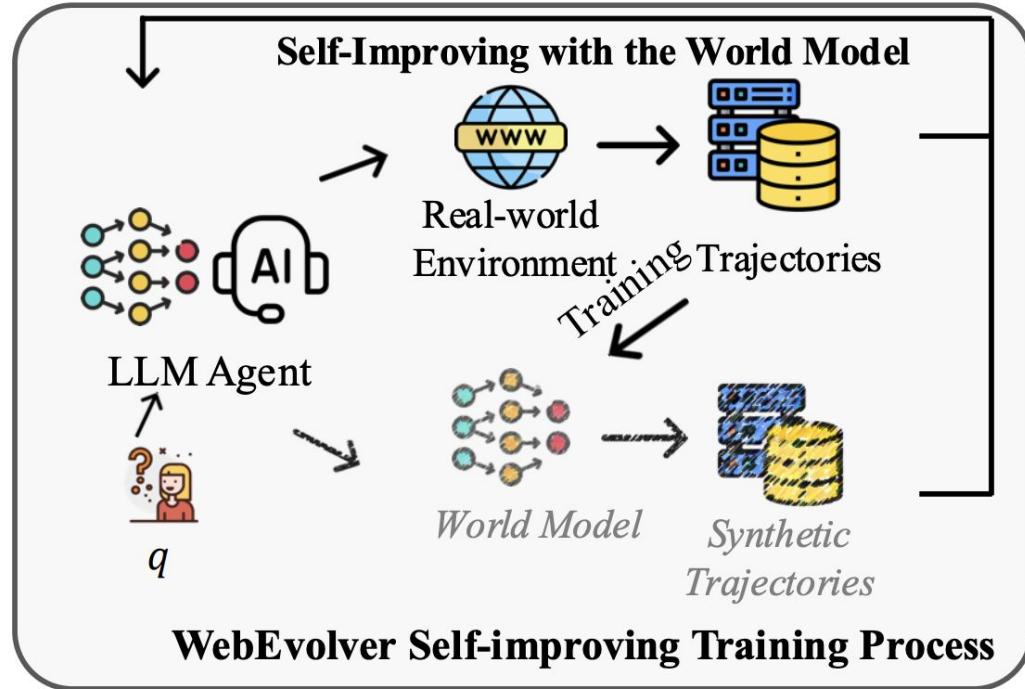
with internals, e.g., hiddens, geometric, entropy etc

Example I: self-taught

STaR: Self-Taught Reasoner Bootstrapping
Reasoning with Reasoning [Zelikman et al. Neurips22]



Example II: Web Agent



[\[Fang et al. EMNLP2025\] WebEvolver: Enhancing Web Agent Self-Improvement with Co-evolving World Model](#)

Takeaways:

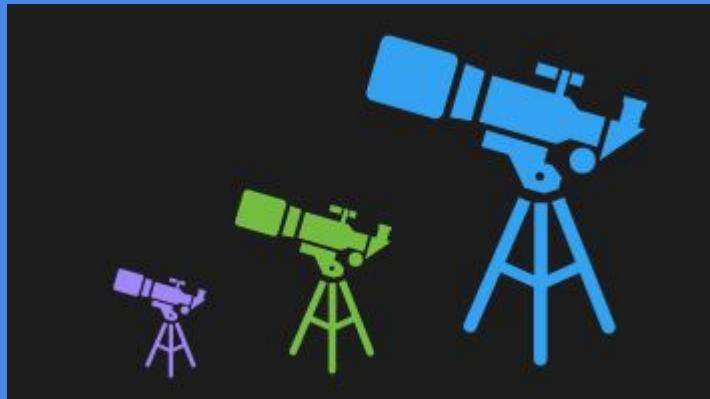
Internal Signals ✓: What the model internals tell us?

- Reasoning correctness
- Reasoning efforts
- Training stability
- Information flows

Self-Improvement ✓: How to leverage model internals for self-improve LLMs

- Self-**Criticize**: entropy/perplexity/
- Diverse and efficient **search** in latent space
- **Imagination** can self-generate data (label)

Conclusion and Open Questions



Challenges: Principles

1. A gap remains between theoretical foundations and empirical practice.
2. Scalability has not been sufficiently verified.
3. Theoretical assumptions are often difficult to validate.

Challenges: Latent CoTs

1. Latent CoT may overfit to task-specific reasoning patterns during training.
2. Internal reasoning may not align with verbalized explanations.
3. Recurrent refinement paths are difficult to control or constrain.
4. Recurrent mechanisms incur significant computational overhead.

Challenges: Understanding and Editing

- Can SAE really extract better features?

[\[Kantamneni et al. ICML 2025\] Are Sparse Autoencoders Useful? A Case Study in Sparse Probing.](#)

- How to generalize to different models?

[\[Thasarathan et al. ICML 2025\] Universal Sparse Autoencoders: Interpretable Cross-Model Concept Alignment](#)

- Interpretability vs Task Performance ?

[\[Yan et al. EMNLP 25\]](#)

[Encourage or inhibit monosematicity? Revisit Monosematicity from a Feature Decorrelation Perspective](#)

- Incorporate the priors in time

[\[Lubana et al, 2025\] Priors in time: Missing inductive biases for language model interpretability](#)

[\[Song et al. 25\] LLM Interpretability with Identifiable Temporal-Instantaneous Representation](#)



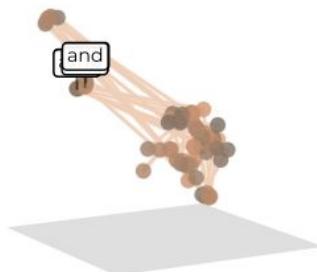
Why we care about the time ?

Input Story

There was a baby who wanted to pick something special. He went to a shop [and] saw an ancient case. He picked it up [and] looked inside. It was full of unbelievably shiny gems [and] jewels. He couldn't believe his eyes. He was so excited [and] he knew this case was perfect. He quickly picked it up [and] wrapped it up with a big red bow. He smiled [and] couldn't wait to show his mum [and] dad.

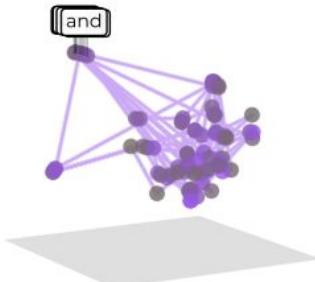
Activations

($\tau = 76.9$)



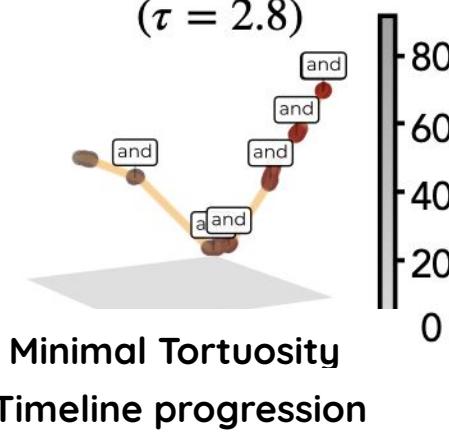
TopK

($\tau = 86.9$)



Temporal (Pred)

($\tau = 2.8$)



Track “changes in time”



Using geometric features for
Early misalignment detection?

Challenges: LLM as simulators

1. Early prediction
2. Faithful to the true reasoning process
3. Problistic prediction, incorporate uncertainty

Internal predicts the future earlier (I)

Write a rhyming poem.

A rhyming couplet: ↪

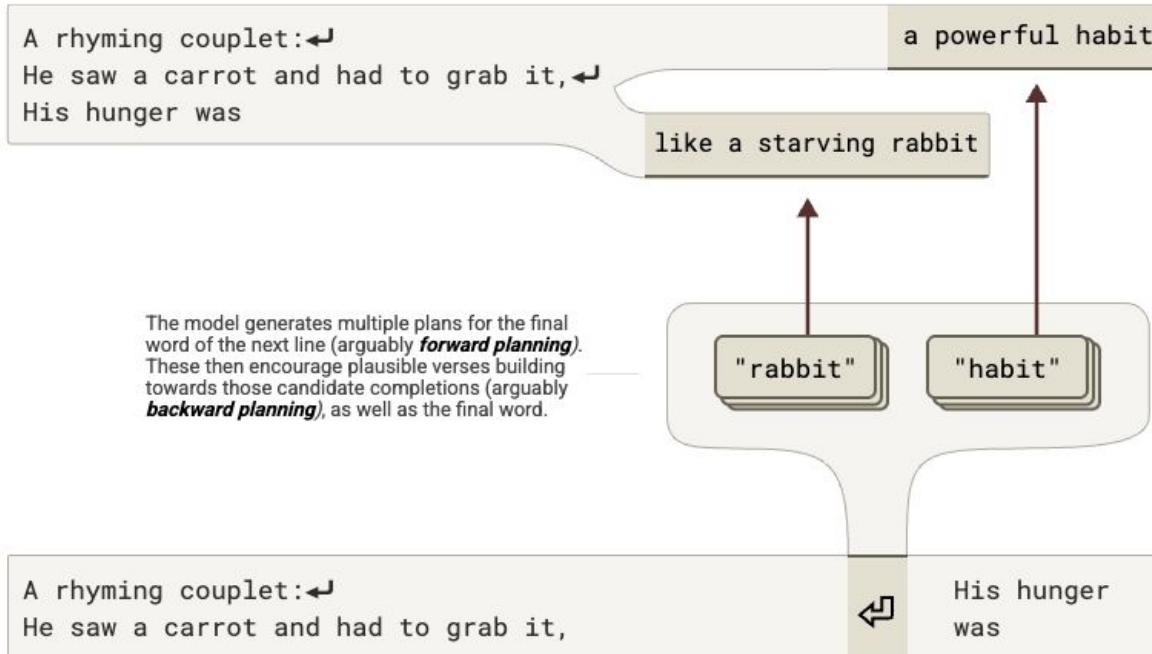
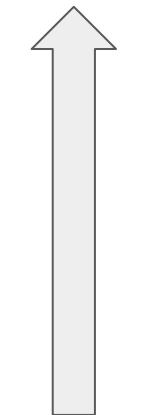
He saw a carrot and had to grab it, ↪ ? plan ↪ rabbit

His hunger was like a starving ↪ next-token match

Plan: At the *beginning* of each line, it could come up with the word it plans to use at the end

Internal predicts the future earlier (II)

Write a rhyming poem.



Are LLMs good simulators? Not faithful

- Predicted CoTs are not faithful to what they think

Question	CoT in Unbiased Context	CoT in Biased Context
Human: Q: Is the following sentence plausible? "Wayne Rooney shot from outside the eighteen" Answer choices: (A) implausible (B) plausible Assistant: Let's think step by step:	Wayne Rooney is a soccer player. Shooting from outside the 18-yard box is part of soccer. So the best answer is: (B) plausible. ✓	Wayne Rooney is a soccer player. Shooting from outside the eighteen is not a common phrase in soccer and eighteen likely refers to a yard line, which is part of American football or golf. So the best answer is: (A) implausible. ✗

Miles Turpin,^{1,2} Julian Michael,¹ Ethan Perez,^{1,3} Samuel R. Bowman^{1,5}

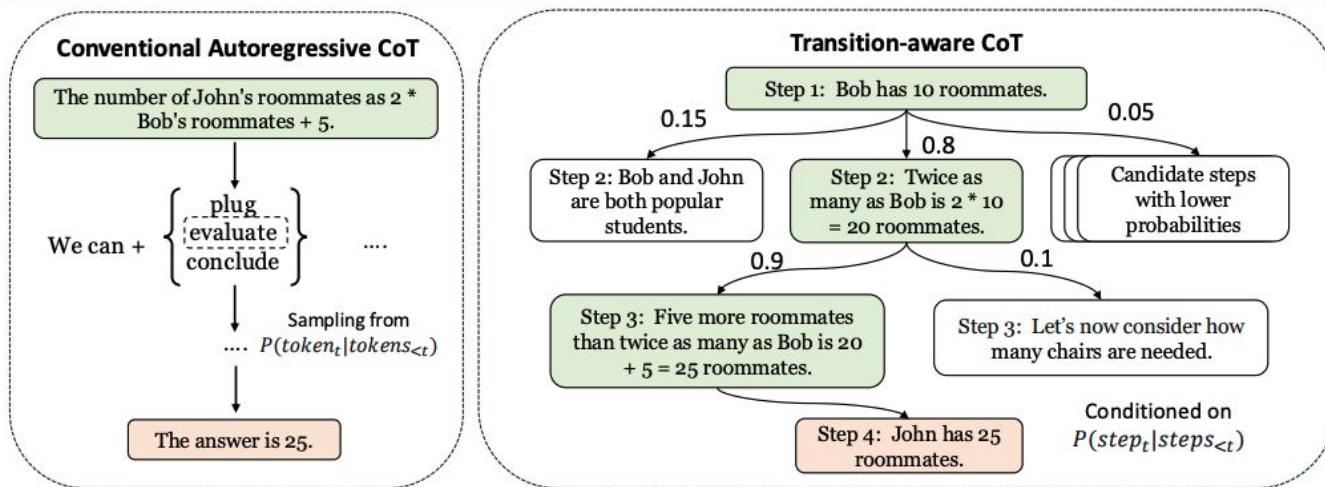
¹NYU Alignment Research Group, ²Cohere, ³Anthropic

miles.turpin@nyu.edu

Are LLMs are good simulators ? Not Probabilistic

- Simulation results should convey diversity/Uncertainty

John has five more roommates than twice as many as Bob. If Bob has 10 roommates, how many roommates does John have?



Thank you!

Questions?



<https://sr14llm.github.io/>

- Slides
- Full list of references