



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



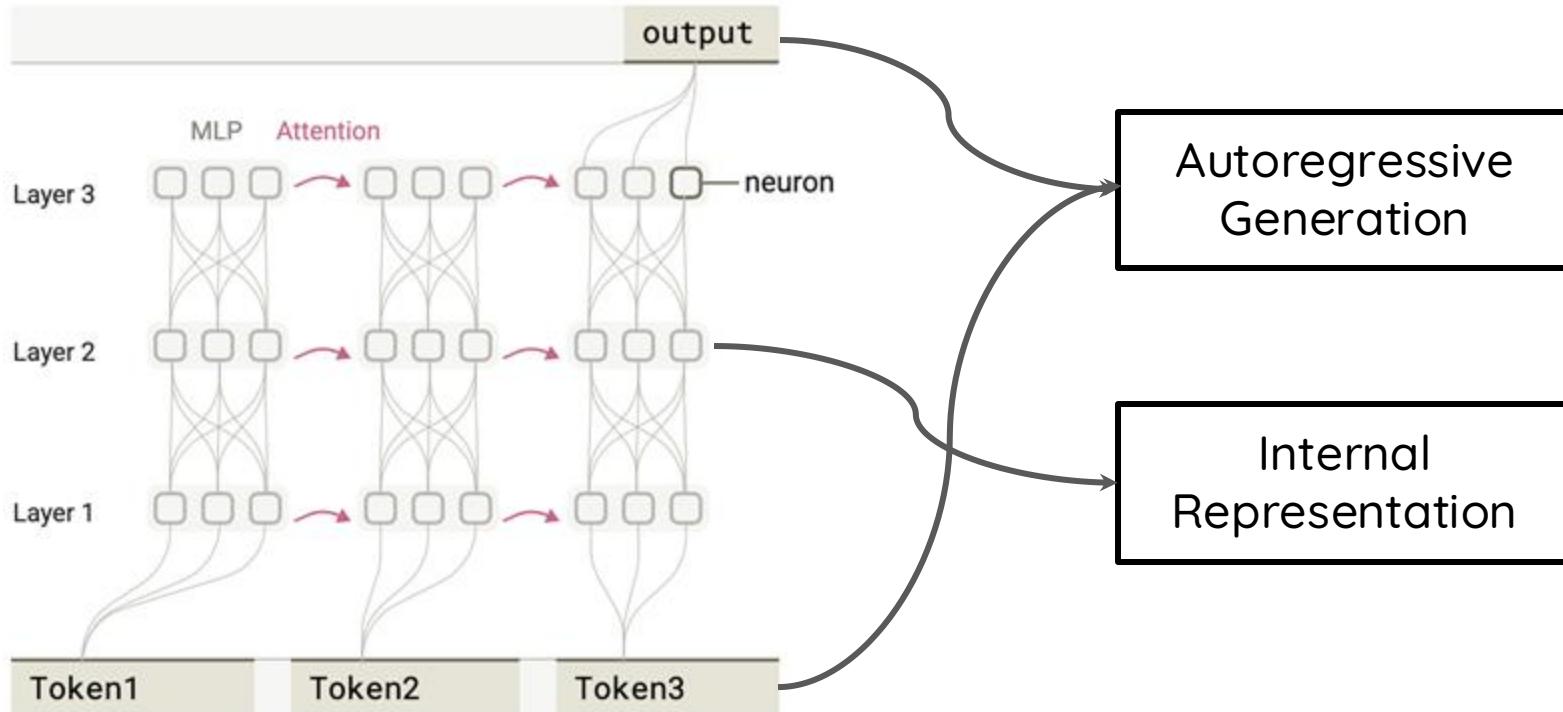
Guangyi Chen
Carnegie Mellon University
MBZUAI



Jonathan Richard Schwarz
Imperial College London
Thomson Reuters

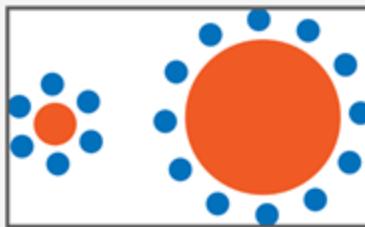
Why do we need to care about Representation Learning?

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

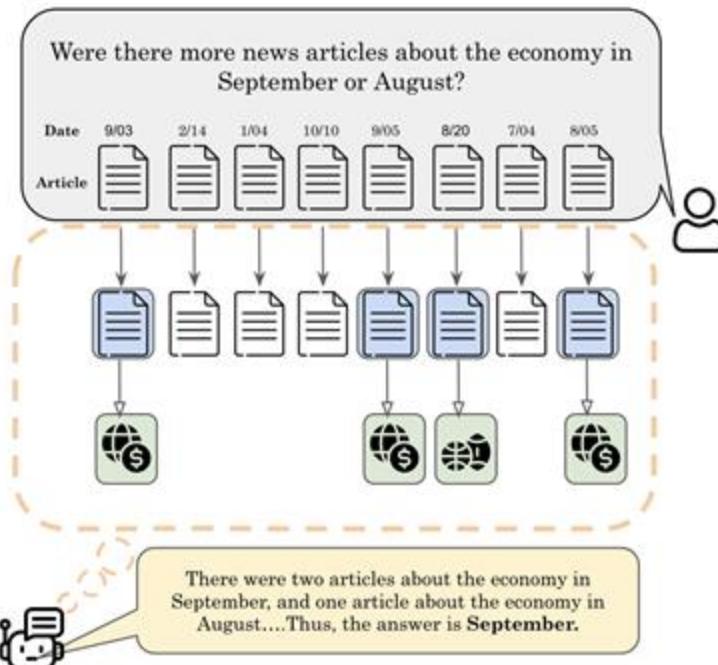
This image is a variant of the Ebbinghaus illusion. Although the circle on the right appears larger due to being surrounded by smaller blue circles, **both orange circles are actually the same size**. The arrangement of the surrounding circles creates a visual context that tricks our perception, making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



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

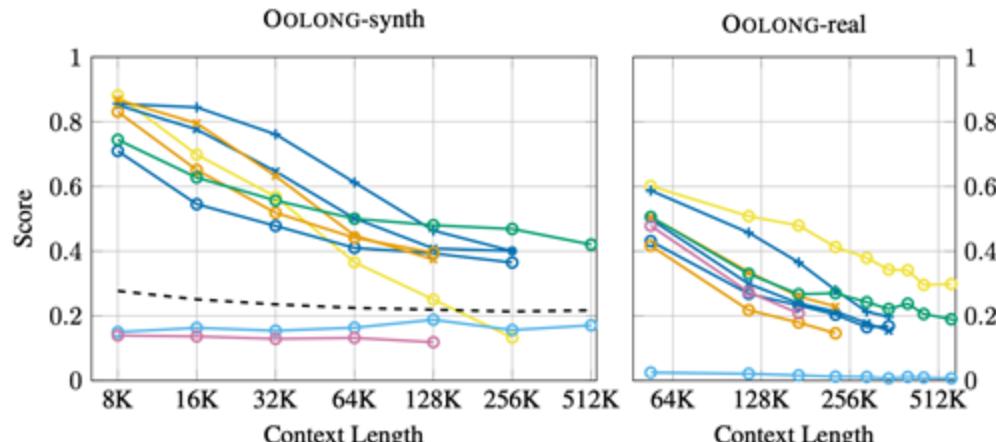
Common LLM failures are internal state failures

Context rot



Context didn't disappear – its influence did.

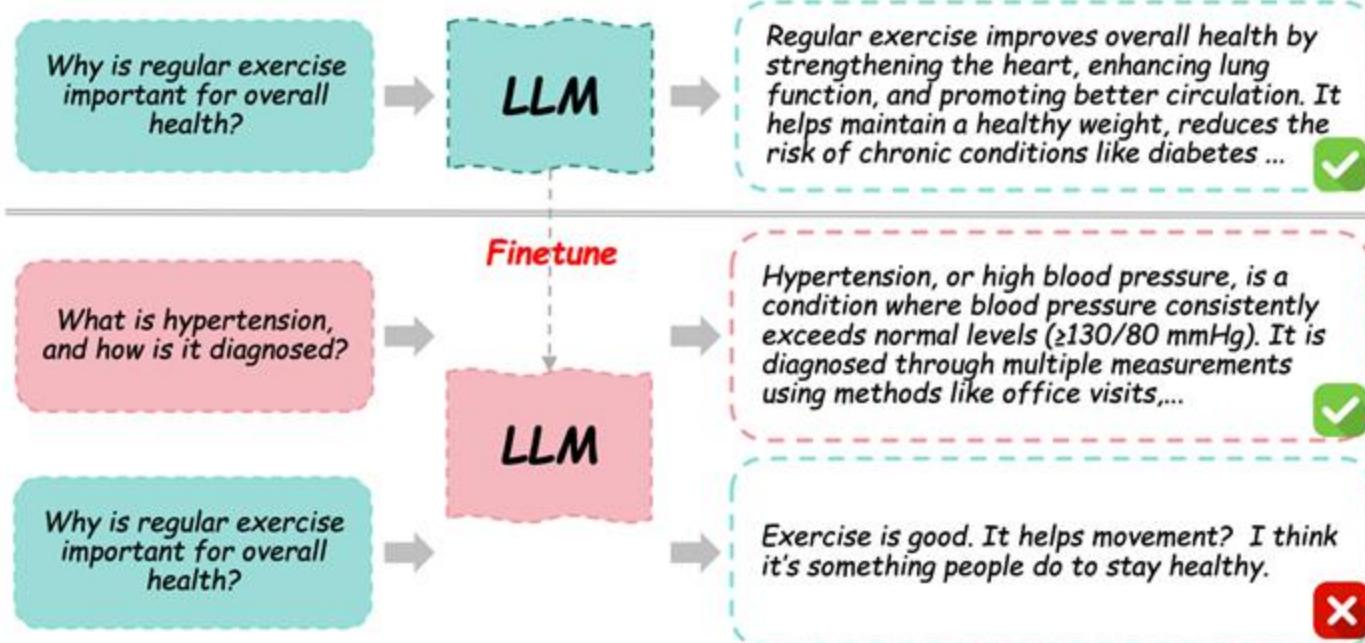
Legend:
gemini-2.5-pro, gpt-5-nano, gpt-5-mini, gpt-5, o4-mini
o3, claude-sonnet-4, llama-4-maverick, deepseek-r1-0528, random baseline



OOLONG (Bertsch *et al.* 2025)

Common LLM failures are internal state failures

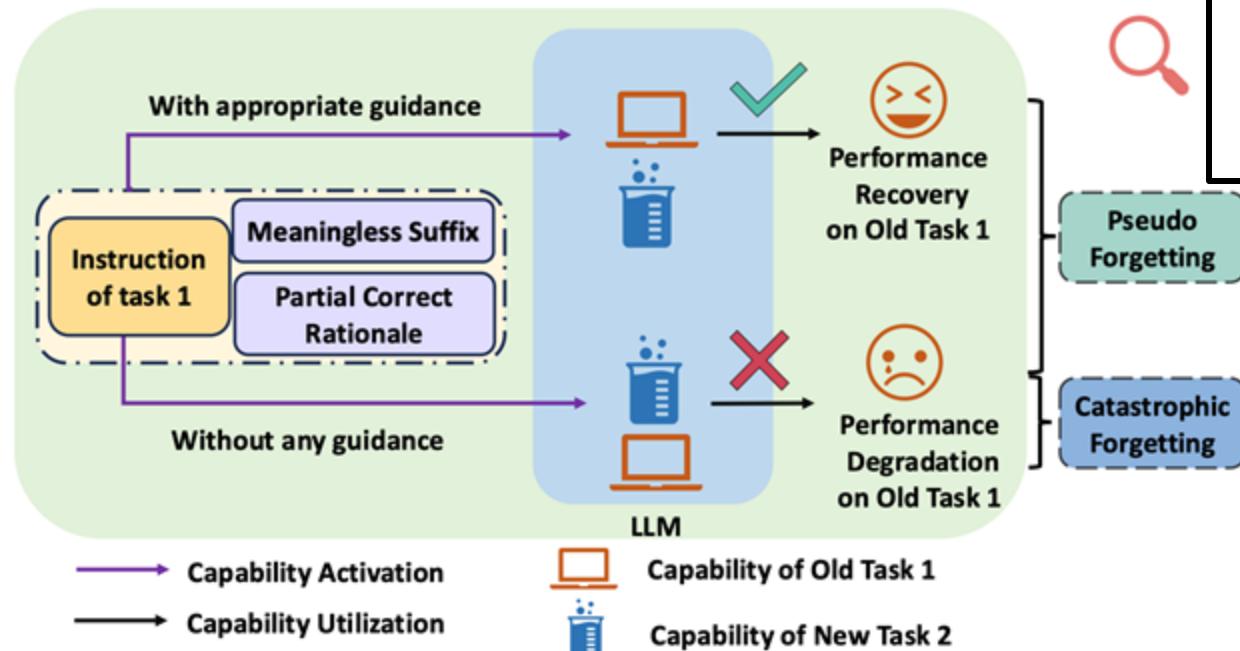
Catastrophic Forgetting



(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.

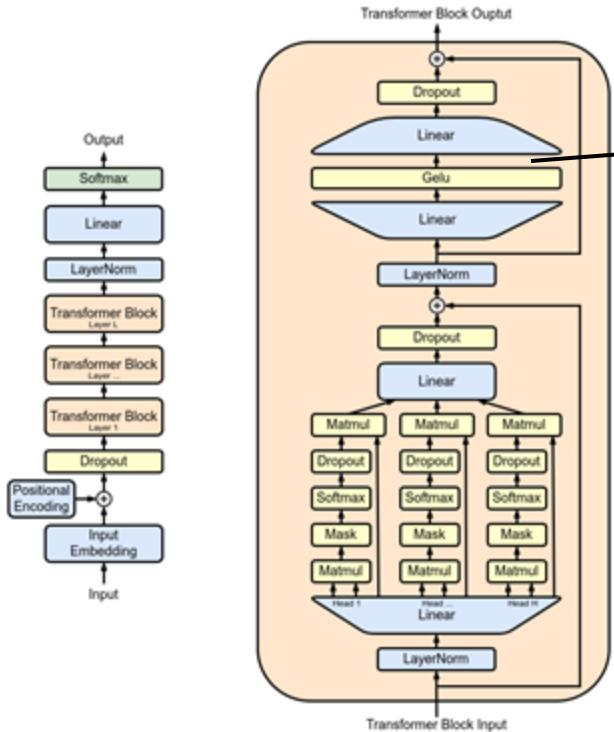
Context rot

- 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.

Where and how is it represented?



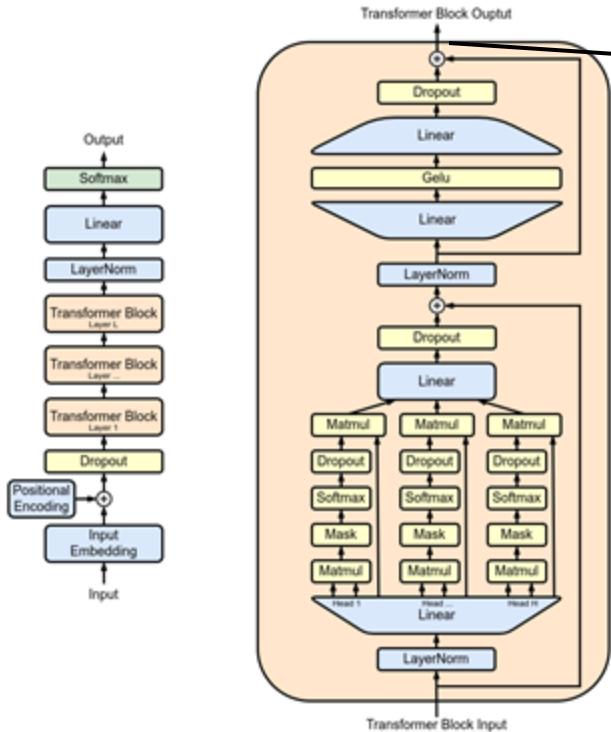
Some methods take the activations in the transformer as the representation for understanding and interpretation.

Linearly decode or probe

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

• • •

Where and how is it represented?



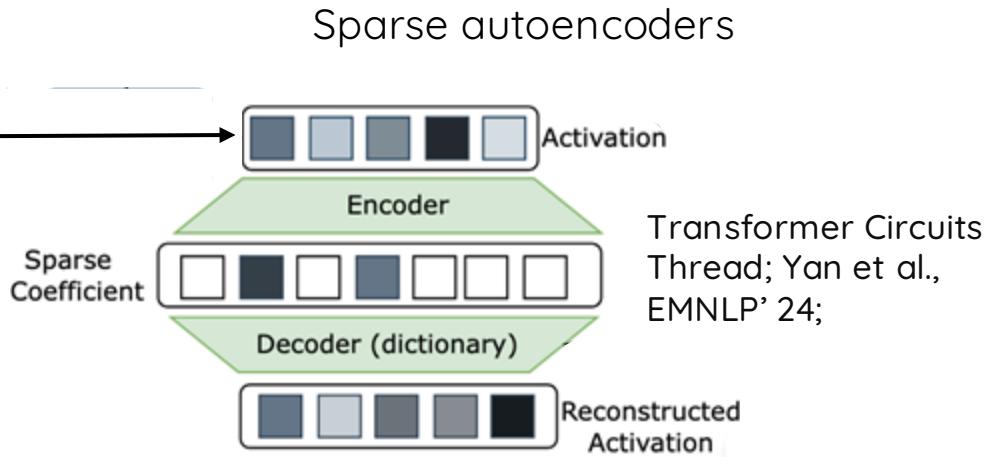
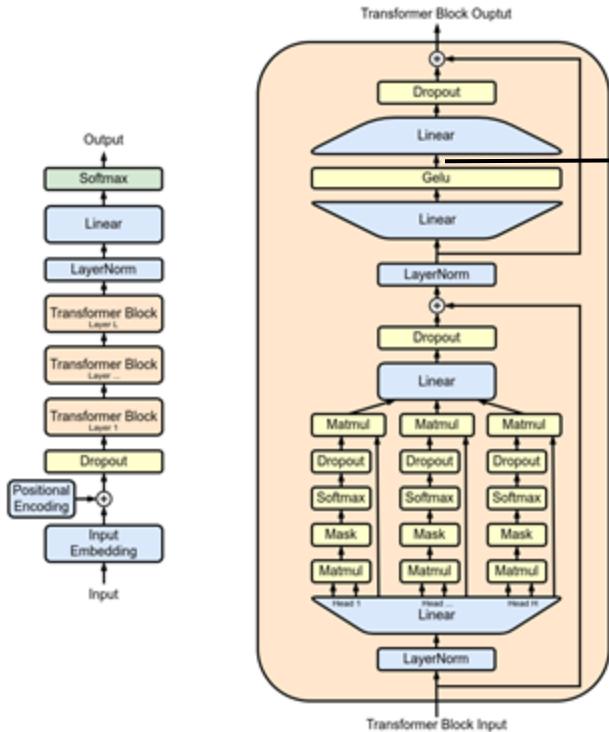
Some methods use the layer outputs as the representations which encode the context information for predictions.

Speculative decoding and latent reasoning

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

...

Where and how is it represented?



To further extract the semantic concepts from the LLMs, some methods learn an extra representation module.

Goals of this tutorial

- ❑ Shift the focus from output-based behavior to understand the “true” think process within LLMs
- ❑ Understand core principles of representation learning
- ❑ Use cases of representation learning in interpretability, model editing, and reasoning
- ❑ Future work and open problems of how to better leverage structured representations for reliable and efficient LLMs

Today's Tutorial Overview

Session 1 Introduction

Session 2 The Principles of Representation Learning

Session 3 Representations for Latent Reasoning

Coffee Break



Session 4 Understand and Model Edit via Representation learning

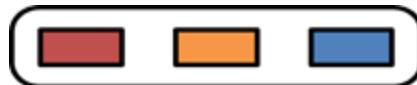
Session 5 Integrate Models Internals for Self-Improvements

Session 6 Conclusion and Future Work

The Principles of Representation Learning

What are good representations

Discriminative



\neq



“Cat”



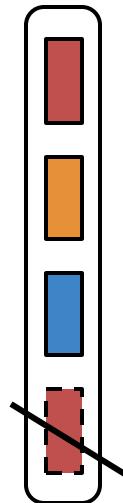
“Dog”



What are good representations

Compact

“There is a
cute cat on
the blanket”

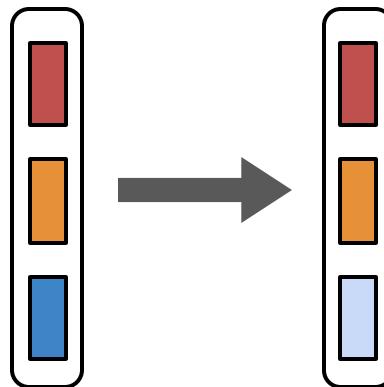


“There is a
cute cat on
the blanket”

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)

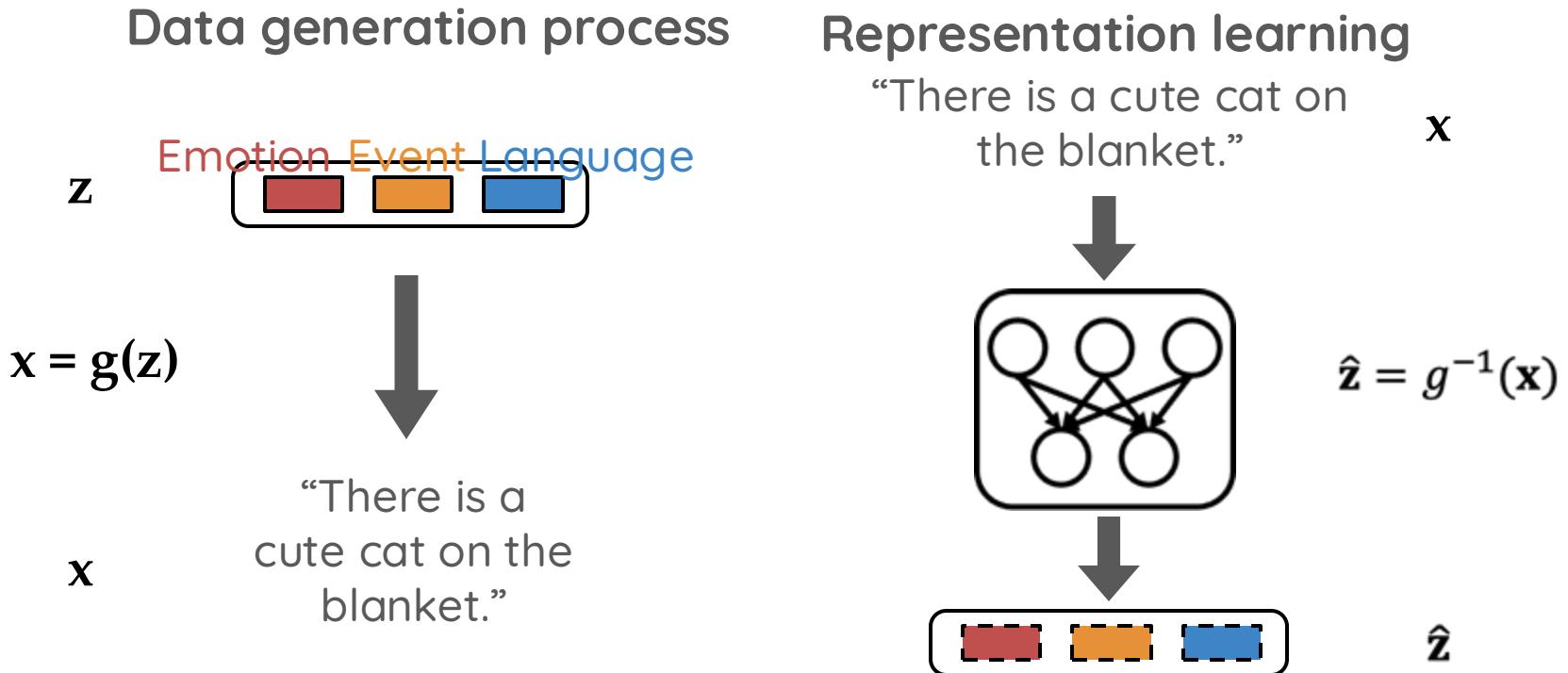


“Oh wow! A cat
on the blanket!
So cute!”

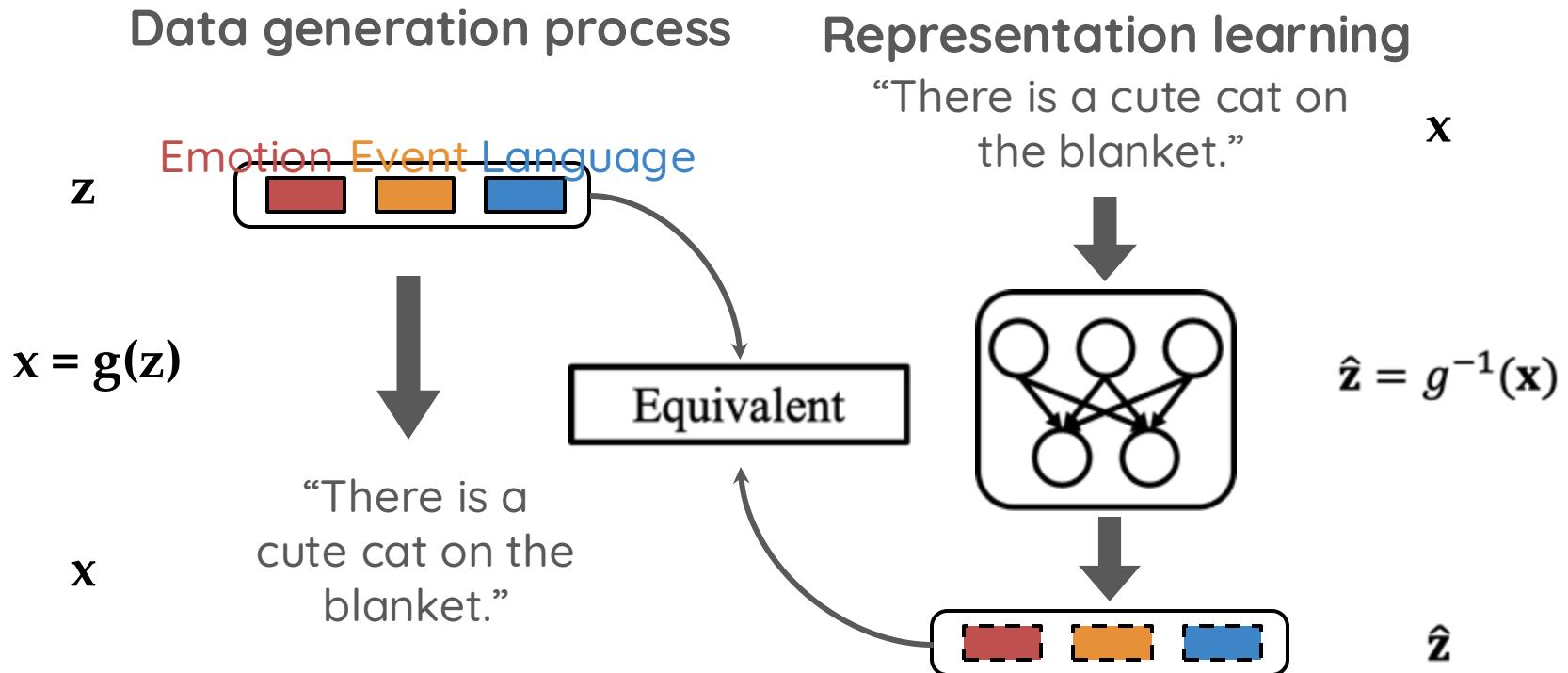
“There is a
cute cat on
the blanket.”

“有一只可爱的
猫在地毯上。”

Identify the latent representation

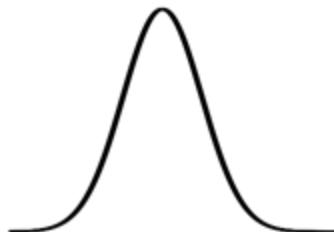


Identify the latent representation

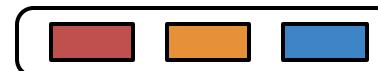


Identifiability

“There is a
cute cat on
the blanket.”



Observed data distribution

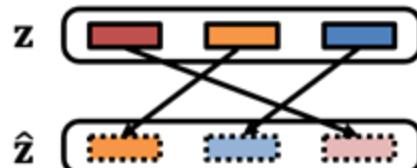


Parameter (representation) space

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**



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

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

$$[\begin{array}{ccc} f_a(2) & f_b(3) & f_c(1) \end{array}]$$

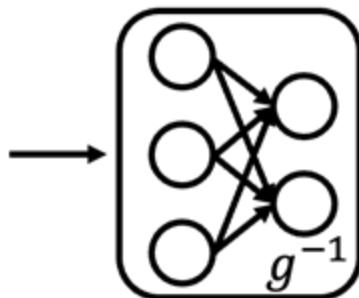
The overall process

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



Causal Representation Learning

“There is a
cute cat
on the
blanket.”



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

Learned feature $\hat{\mathbf{z}}$

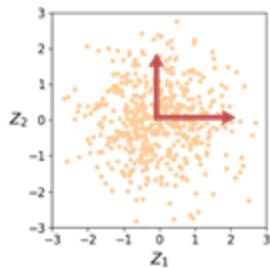
Equivalent

Ground truth \mathbf{z}

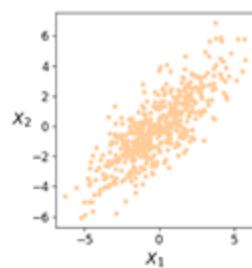
Intuition: a simple linear case $X=AZ$

Linear Gaussian

Latent variable (z)



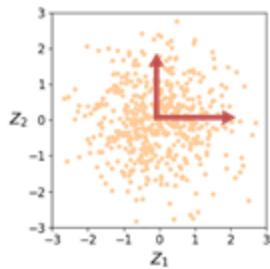
Observation (x)



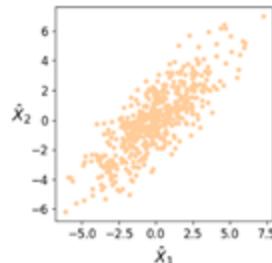
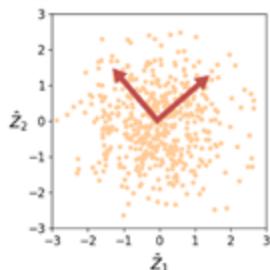
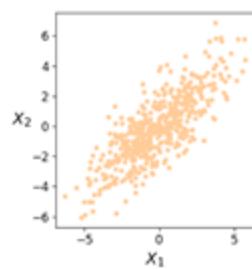
Intuition: a simple linear case $X=AZ$

Linear Gaussian

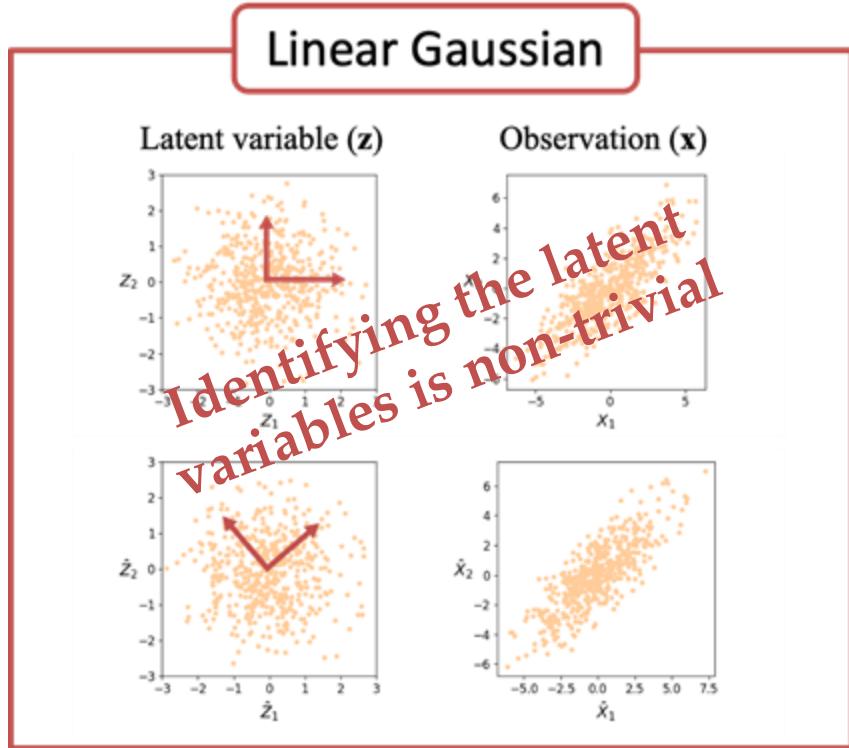
Latent variable (z)



Observation (x)



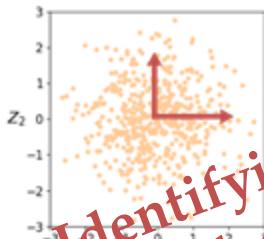
Intuition: a simple linear case $X=AZ$



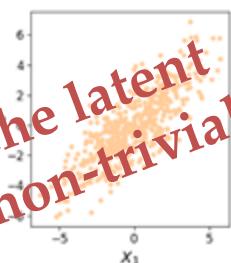
Intuition: a simple linear case $X=AZ$

Linear Gaussian

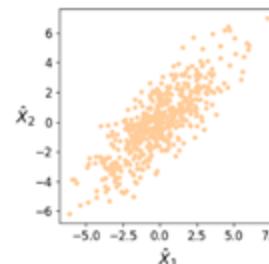
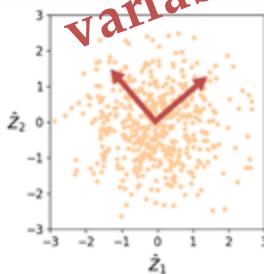
Latent variable (z)



Observation (x)

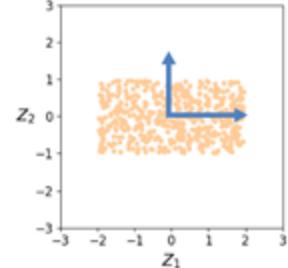


Identifying the latent variables is non-trivial

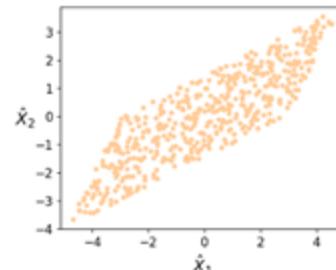
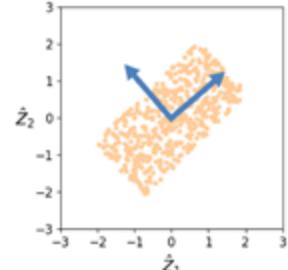
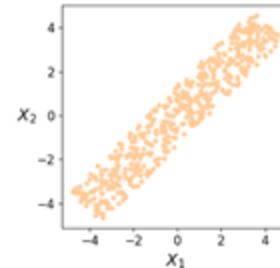


Linear Non-Gaussian

Latent variable (z)



Observation (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.

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]:** $\log p_{\mathbf{z}|\mathbf{u}}(\mathbf{z}|\mathbf{u}) = \sum_i \log p_{\mathbf{z}_i|\mathbf{u}}(\mathbf{z}_i|\mathbf{u})$
- **[Marginal distribution matching]:** $p_{\mathbf{x}|\mathbf{u}} = p_{\hat{\mathbf{x}}|\mathbf{u}}$

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

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.

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.

Sufficient change principle

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



Identification Condition

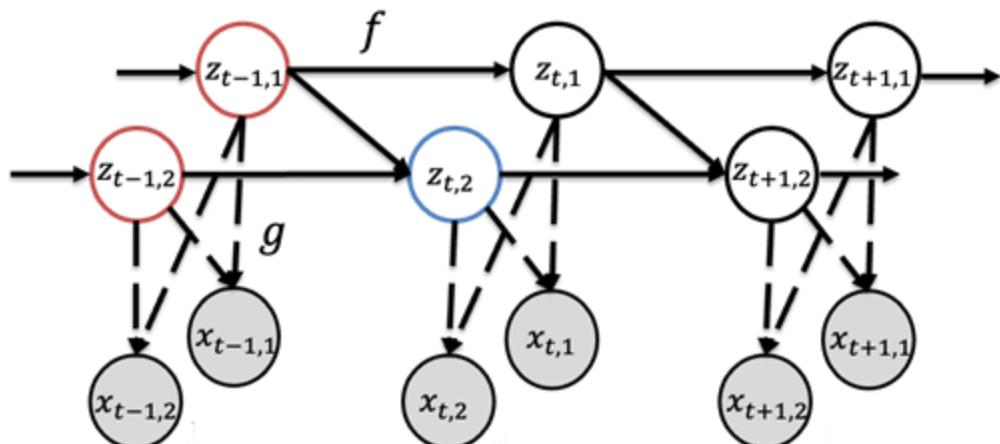
If the data is generated as described above and meets the following criteria:

- [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

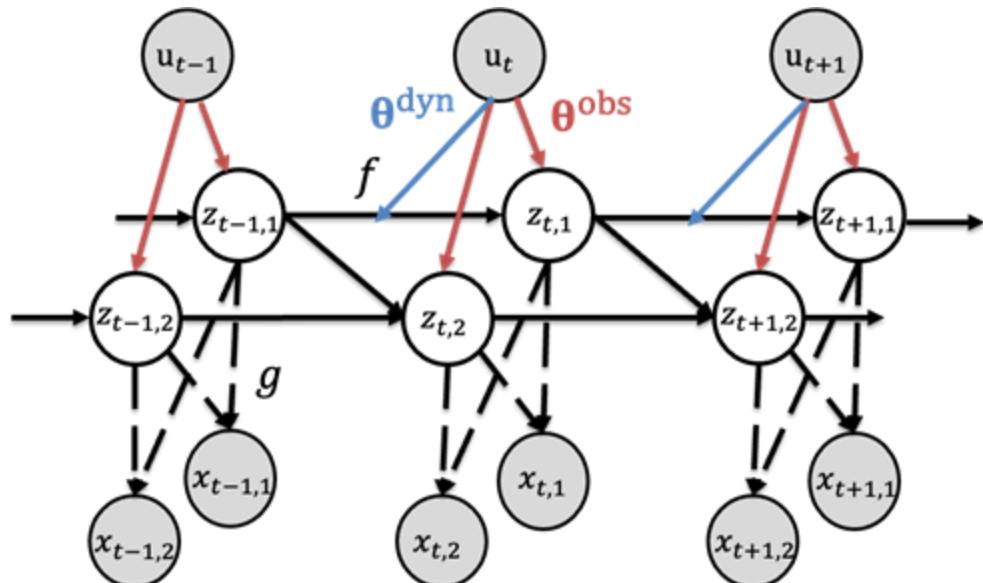
In the temporal scenario, we can use the historical information $\{z_{<t}\}$ as  an auxiliary to provide **sufficient change** and thus establish identifiability.



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

There is a ...

Dynamic with known non-stationary



a cat.

However,

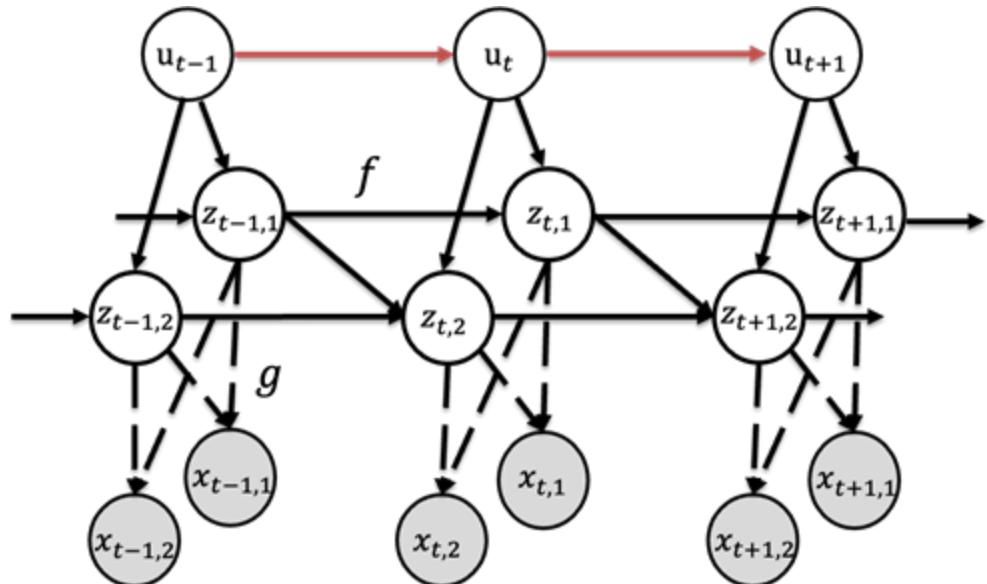
we

...

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

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

Dynamic with unknown non-stationary



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

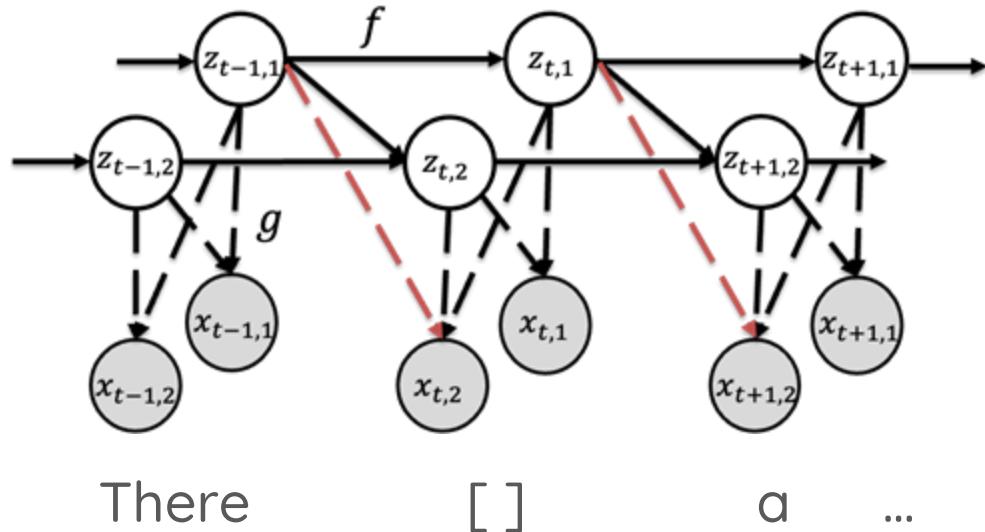
$$\left\{ \begin{array}{l} \mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_t \sim \text{Markov Chain}(\mathbf{A}) \\ \mathbf{z}_{t,i} = f_i(\mathbf{Pa}(\mathbf{z}_{t,i}), \mathbf{u}_t, \varepsilon_{t,i}) \\ \mathbf{x}_t = g(\mathbf{z}_t) \end{array} \right.$$

“This rule applies until an exception is learned.”

Non-Invertible generation process

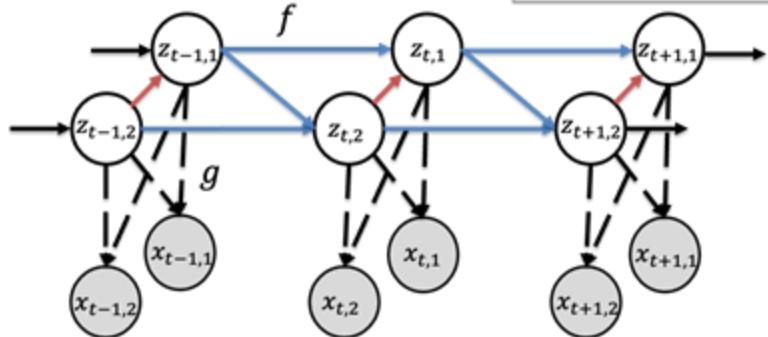
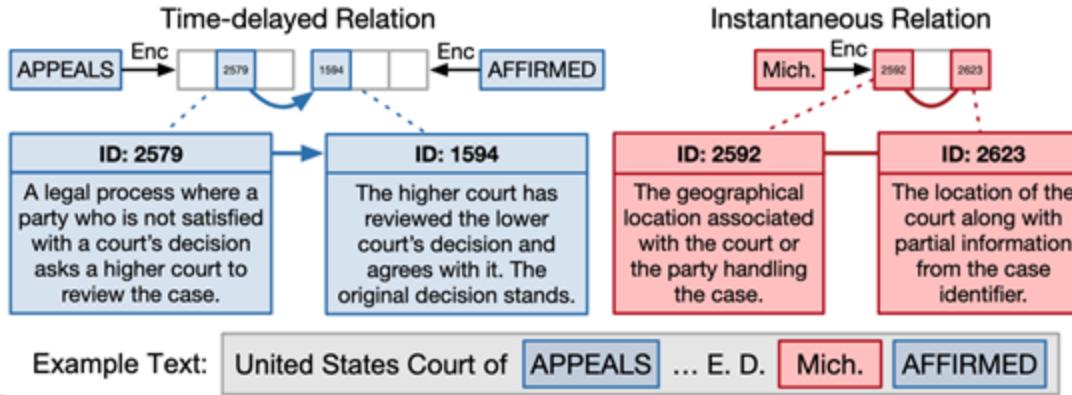
The generation process may be non-invertible, caused by typos/masks.

We can leverage the context to recover the lost information



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

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

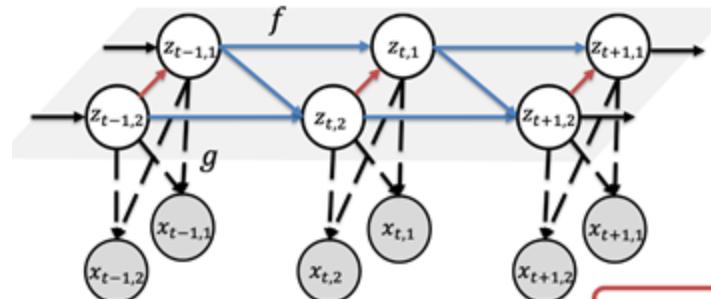
- ★ The Sparsity Principle

- ★ 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

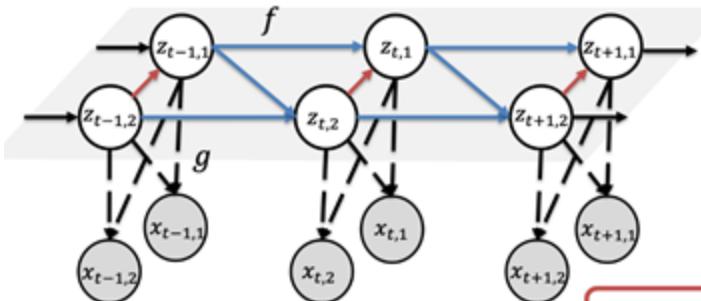
If the data is generated as described above and meets the following criteria:

- **[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 $p_{z_t|z_{<t}}$ is smooth.
- **[Sufficient change]:** there exist enough values of $\{\mathbf{z}_{<t}\}$ providing enough changes.
- **[Marginal distribution matching]:** $p_{x_t|x_{<t}} = p_{\tilde{x}_t|\tilde{x}_{<t}}$

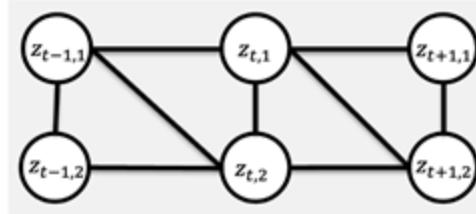
Then, the learned latent variables are component-wise identifiable.

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}



Identification Condition

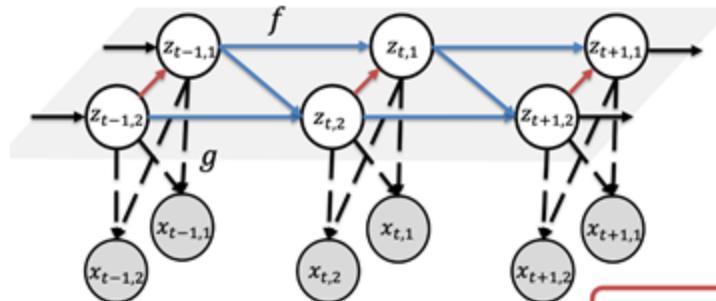
If the data is generated as described above and meets the following criteria:

- **[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 $p_{z_t|z_{<t}}$ is smooth.
- **[Sufficient change]:** there exist enough values of $\{\mathbf{z}_{<t}\}$ providing enough changes.
- **[Marginal distribution matching]:** $p_{x_t|z_{<t}} = p_{x_t|z_{<t}}$

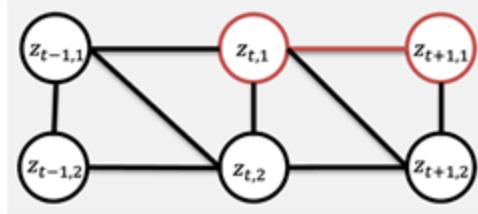
Then, the learned latent variables are component-wise identifiable.

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

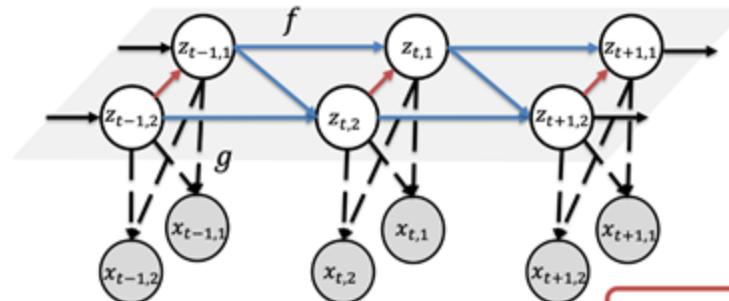
If the data is generated as described above and meets the following criteria:

- [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 $p_{x_t|z_{<t}}$ is smooth.
- [Sufficient change]: there exist enough values of $\{\mathbf{z}_{<t}\}$ providing enough changes.
- [Marginal distribution matching]: $p_{x_t|z_{<t}} = p_{\tilde{x}_t|\tilde{z}_{<t}}$

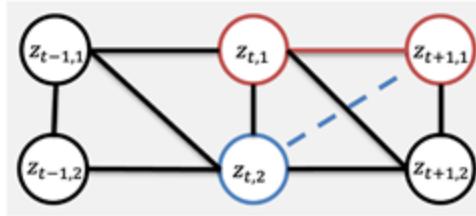
Then, the learned latent variables are component-wise identifiable.

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

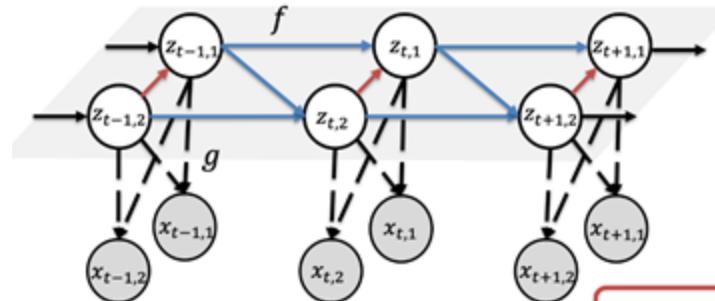
If the data is generated as described above and meets the following criteria:

- **[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 $p_{z_t|z_{<t}}$ is smooth.
- **[Sufficient change]:** there exist enough values of $\{z_{<t}\}$ providing enough changes.
- **[Marginal distribution matching]:** $p_{x_t|x_{<t}} = p_{\tilde{x}_t|\tilde{x}_{<t}}$

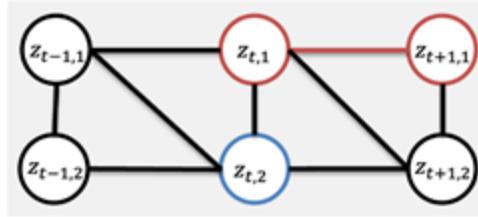
Then, the learned latent variables are component-wise identifiable.

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: $|\hat{\mathcal{M}}| \geq |\mathcal{M}|$
Sparse Constraint: $|\hat{\mathcal{M}}| \leq |\mathcal{M}|$

Identification Condition

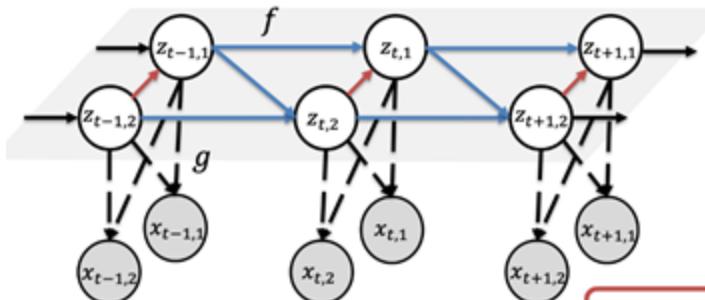
If the data is generated as described above and meets the following criteria:

- **[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 $p_{x_t|z_{<t}}$ is smooth.
- **[Sufficient change]:** there exist enough values of $\{z_{<t}\}$ 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.

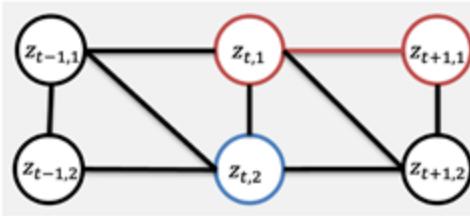
The sparsity principle

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



Markov Network \mathcal{M}

Sparse Constraint



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

Identification Condition

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

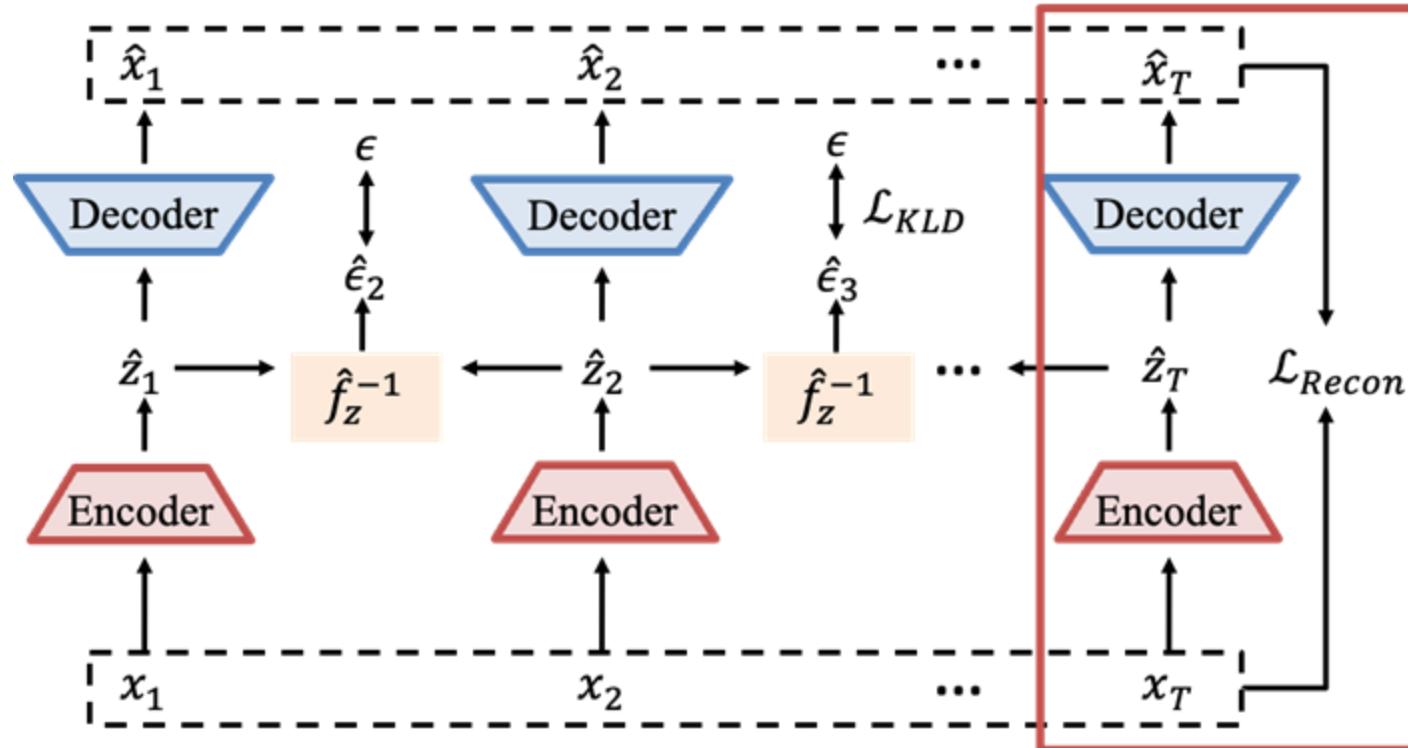
- **[Sparse Latent Process]:** sparse enough to provide *effect change*.
- **[Sparse Constraint]:** the edges of the estimated Markov Network $|\hat{\mathcal{M}}|$ is minimal.
- **[Invertible function and smooth density]:** g is invertible, and $p_{\mathbf{z}_t|\mathbf{z}_{<t}}$ is smooth.
- **[Sufficient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

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

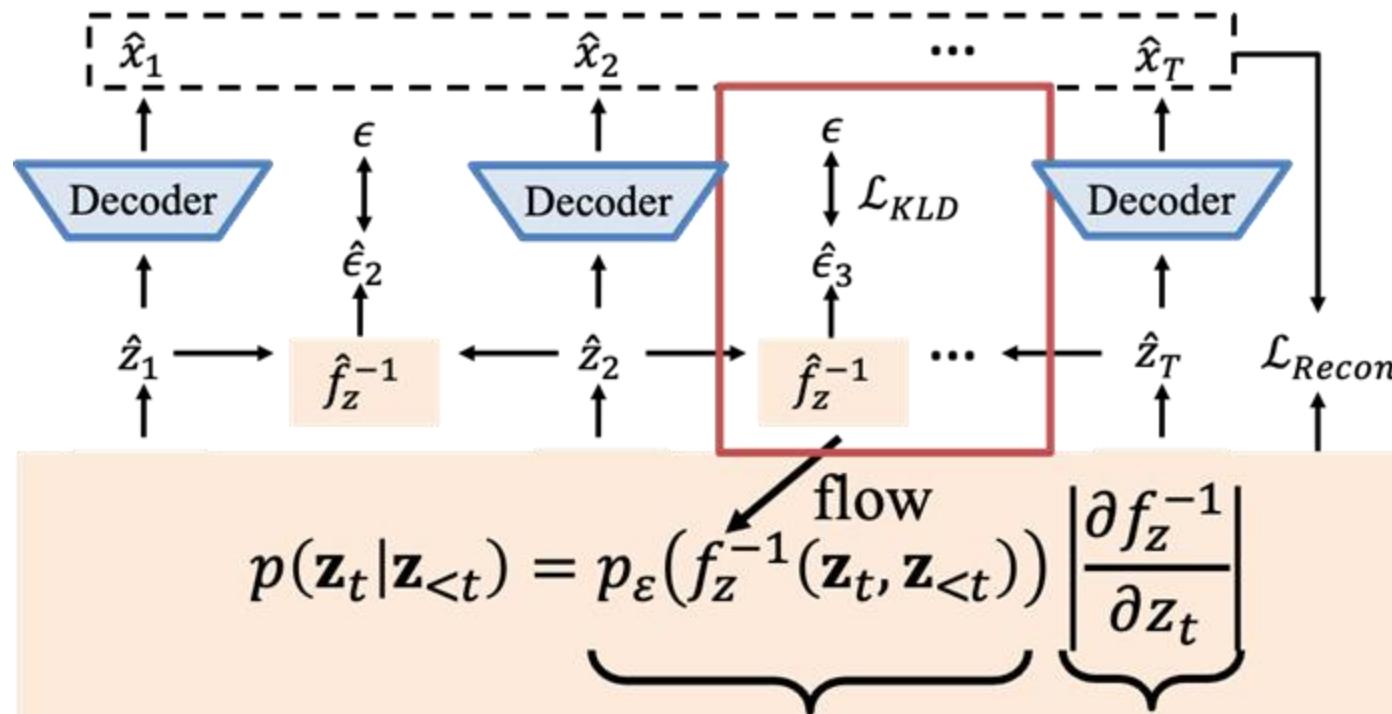
What principles can we use to learn representations?

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

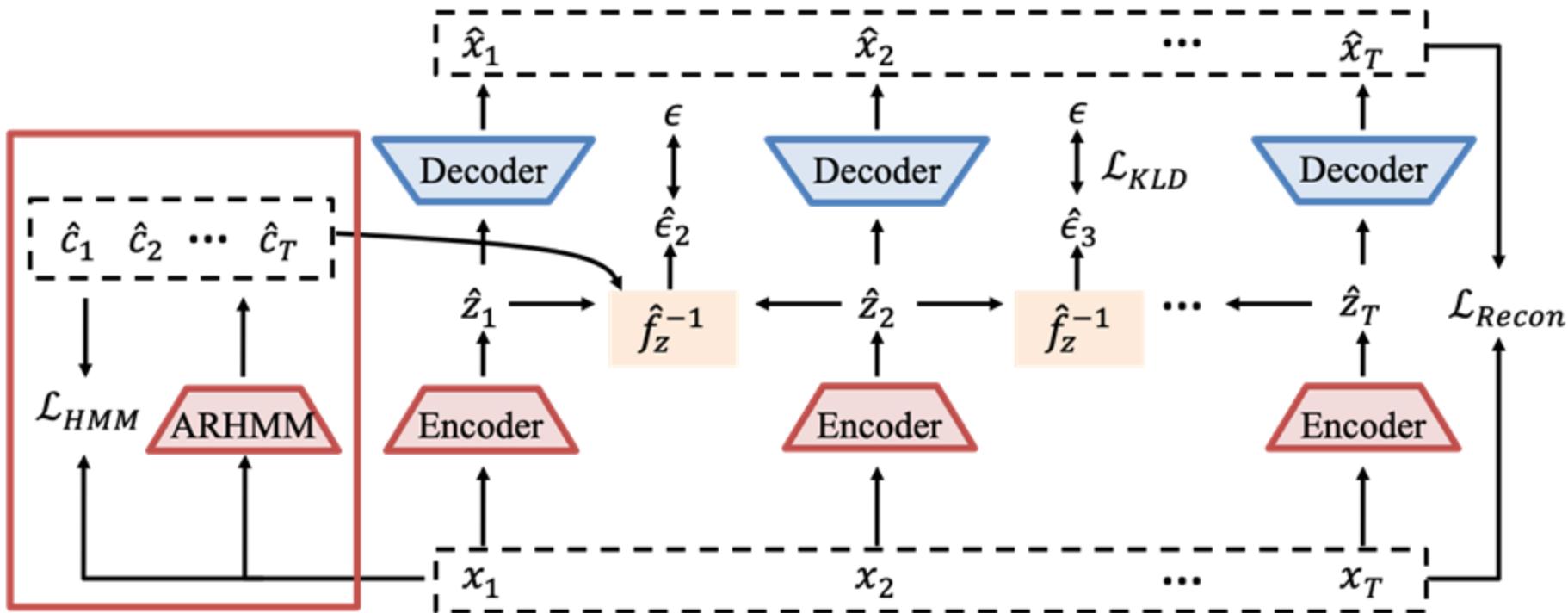
Implementation in VAE framework



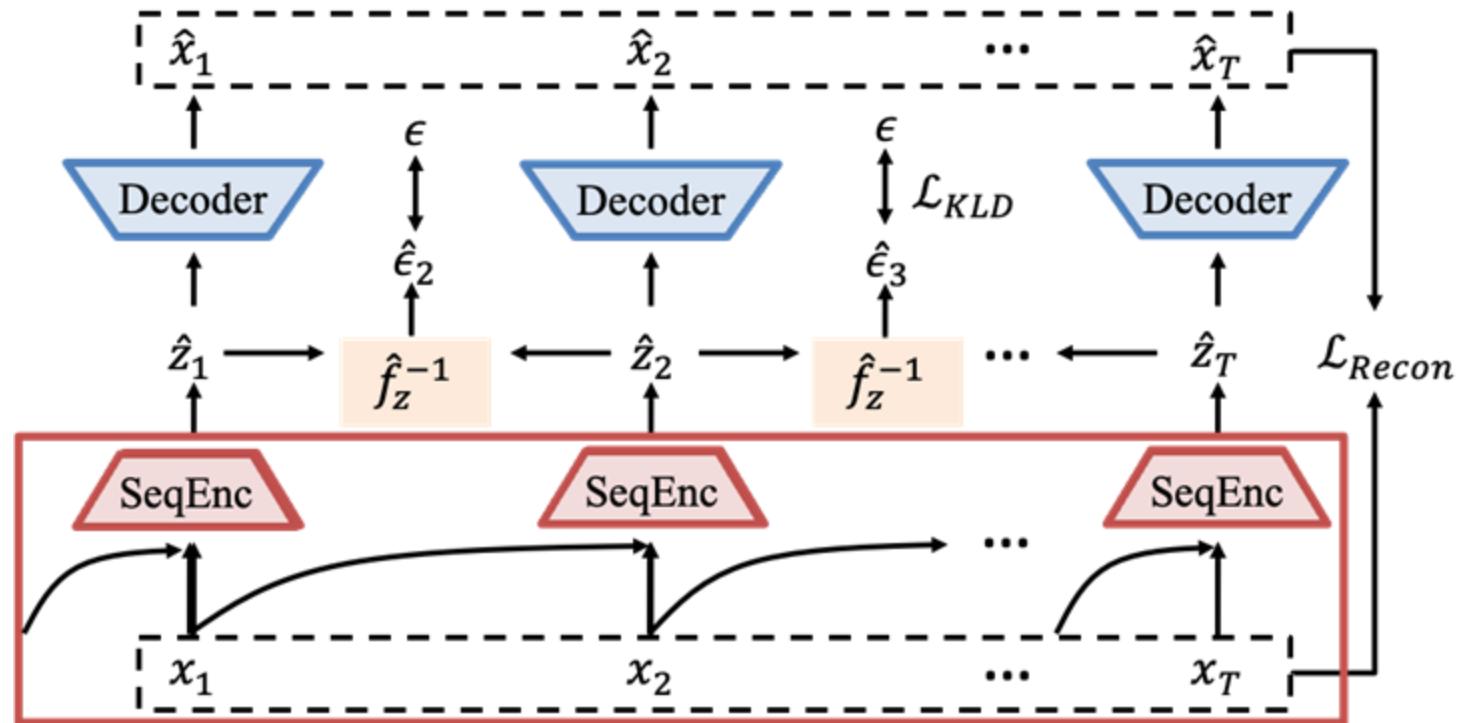
Conditional independence — prior network



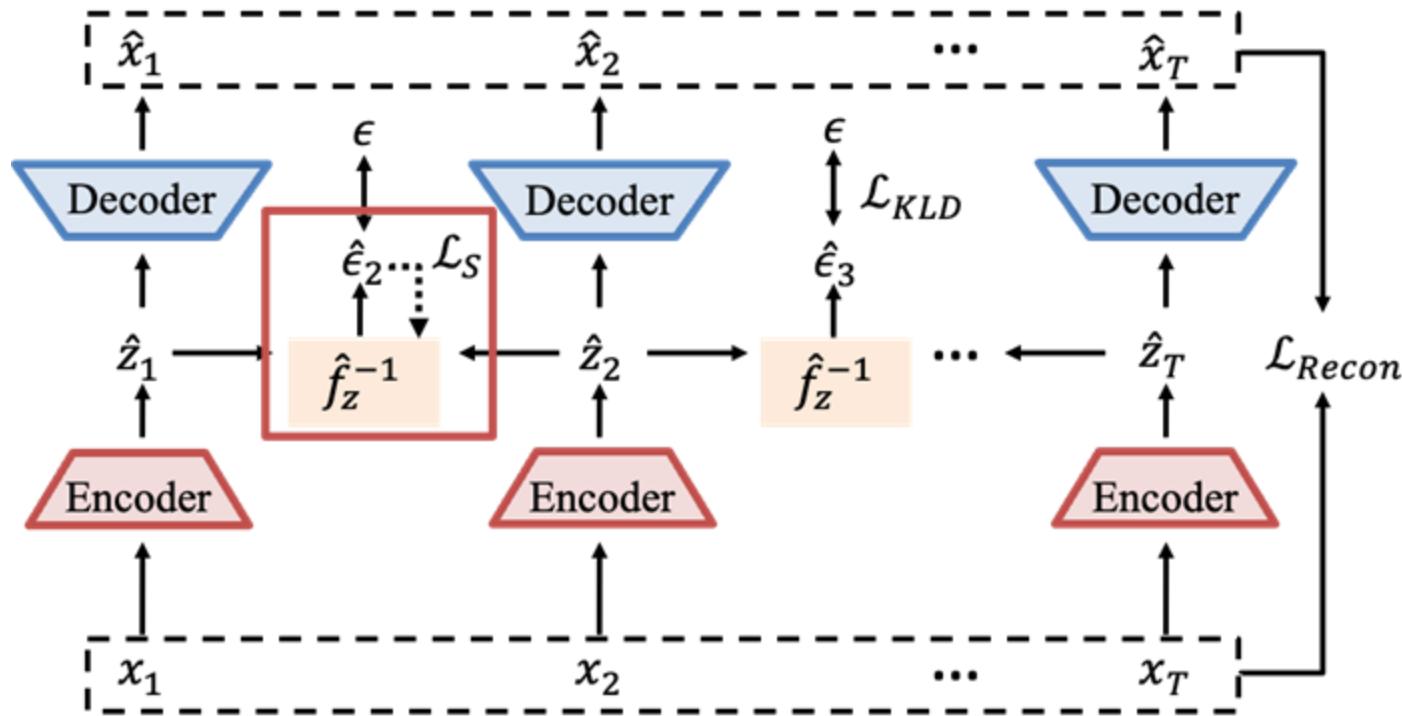
Autoregressive hidden Markov module



Non-invertibility — context encoder



Instantaneous dependency— sparsity loss

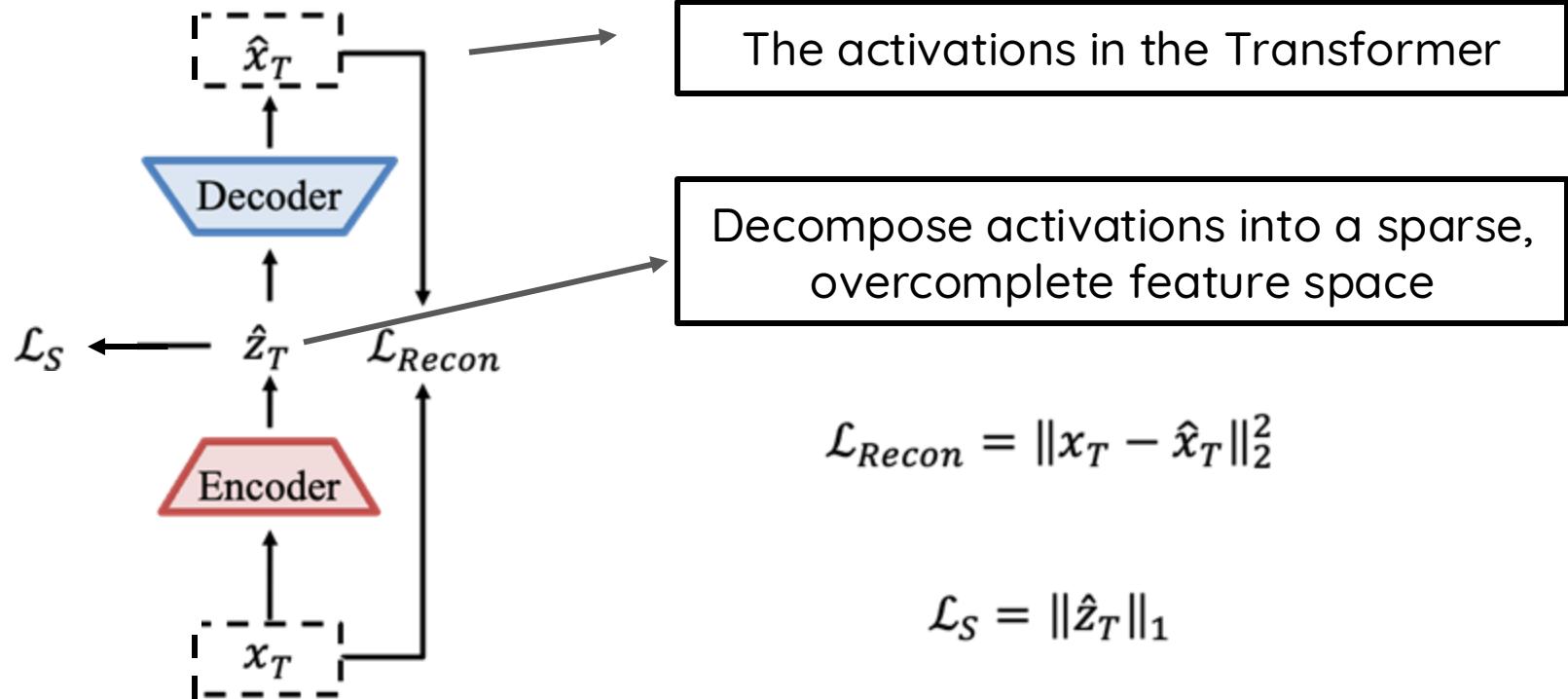


What principles can we use to learn representations?

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

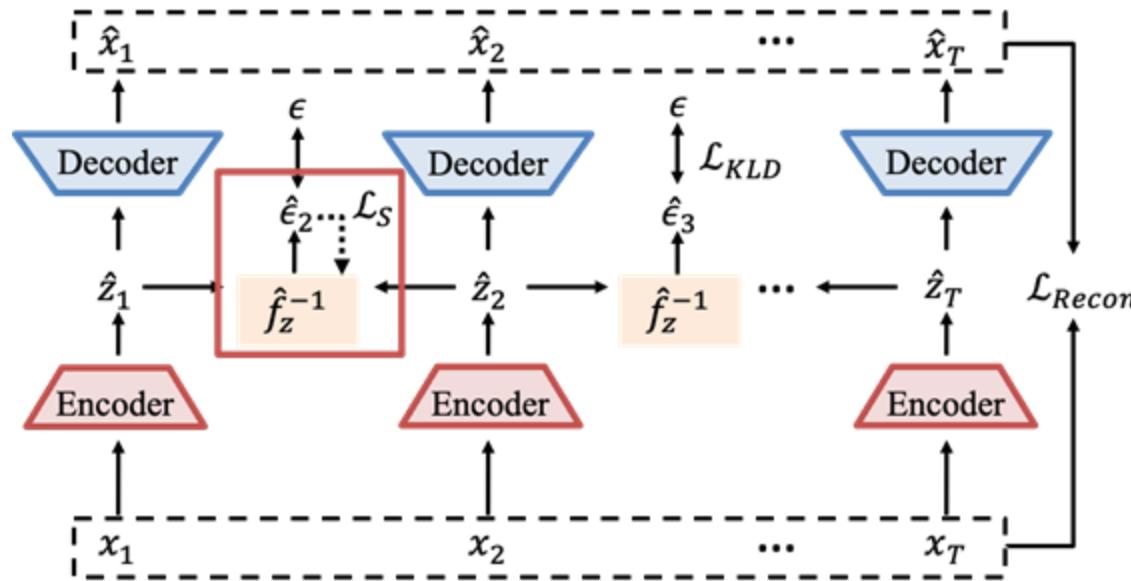
★ Application Showcase

Application for Sparse Autoencoder (SAE)

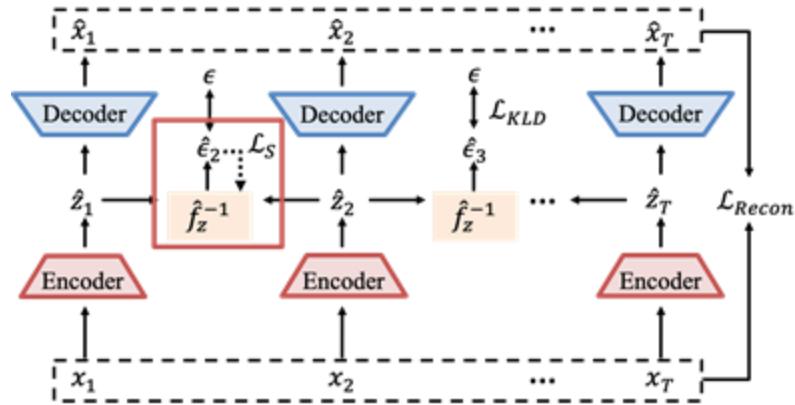


Application for Sparse Autoencoder (SAE)

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



Application for Sparse Autoencoder (SAE)



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

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

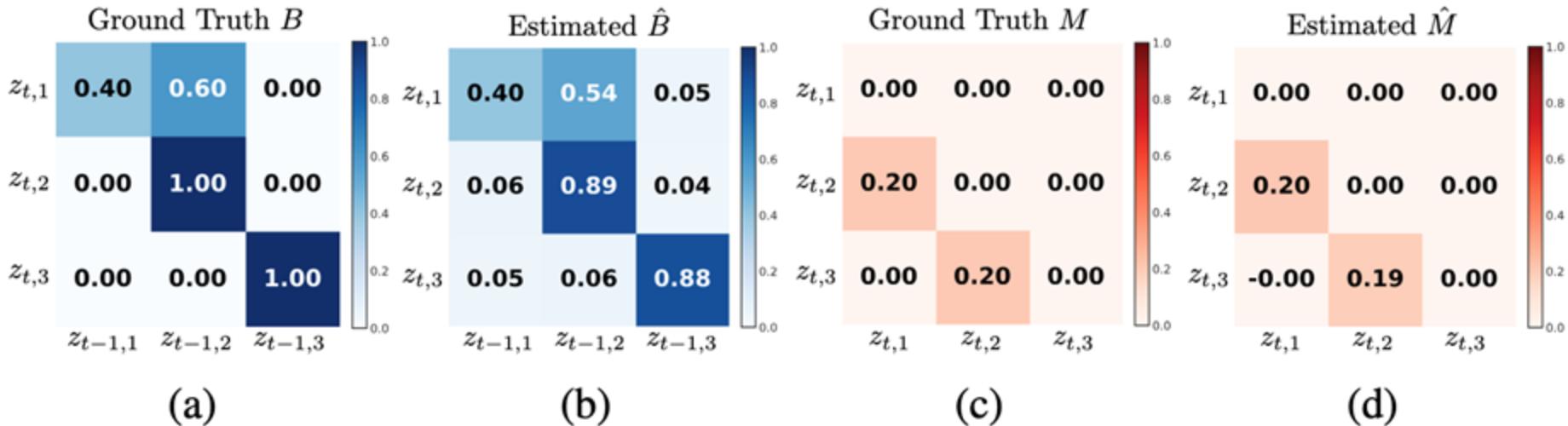
time-delayed instantaneous

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

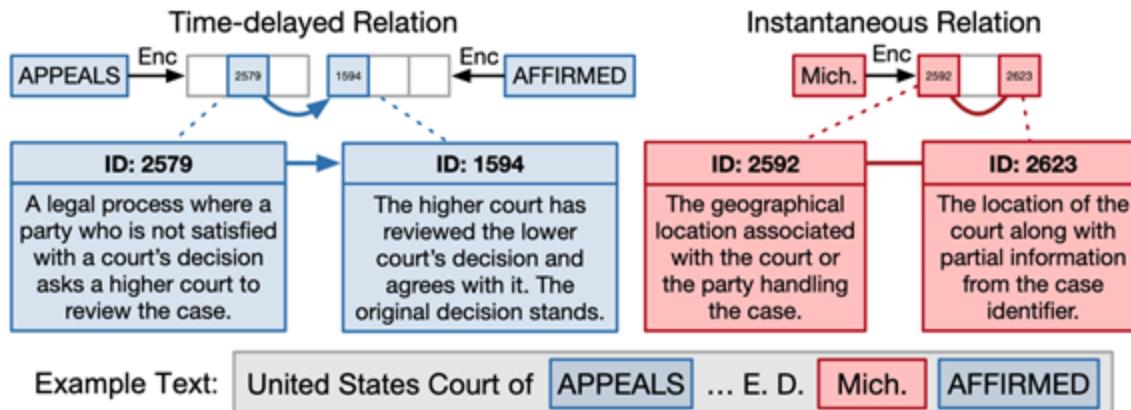
$$\mathcal{L}_{Recon} = \mathbb{E}_{x_{1:T}} \left[\sum_{t=1}^T \|x_t - \widehat{x}_t\|_2^2 \right]$$

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 temporal dynamics exhibit superior relation recovery ability.

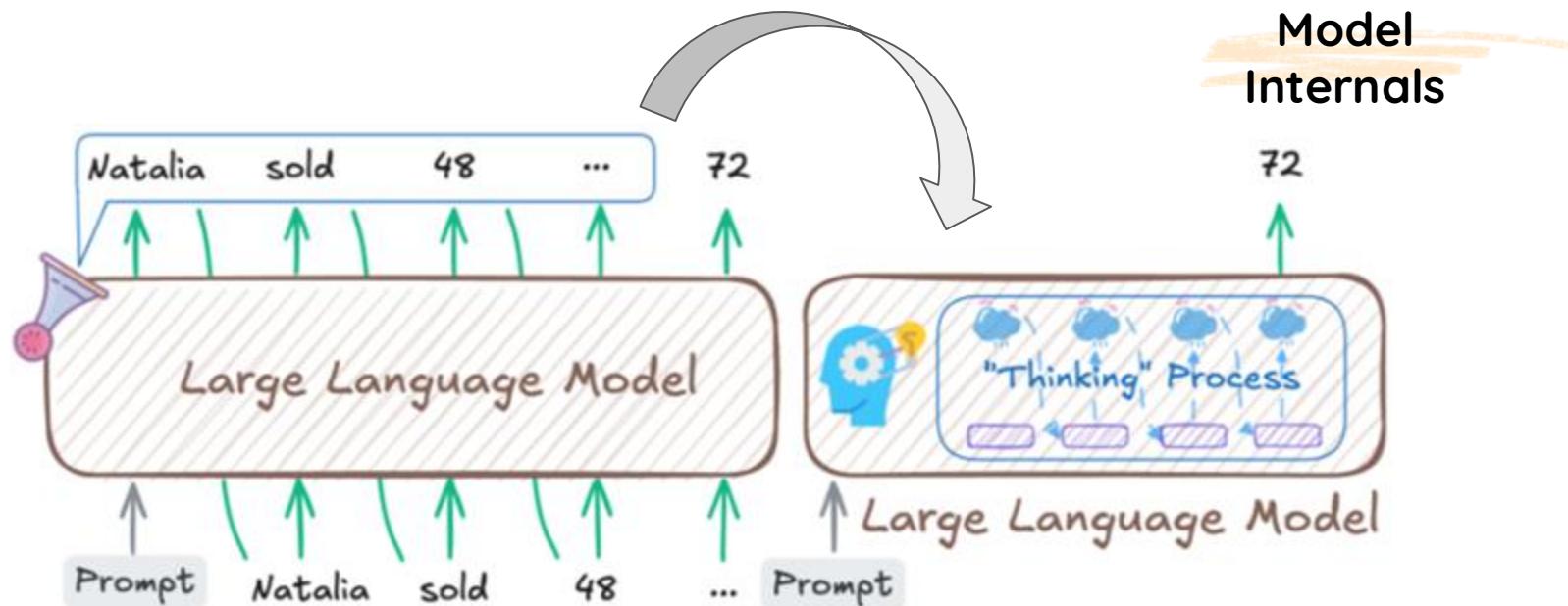
Method	Legal	XML	Email
SAE+regression	0.54	0.94	0.74
Ours	19.95	8.63	2.66

Principles summary

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

Representations for Latent Reasoning

Surface reasoning vs. latent reasoning

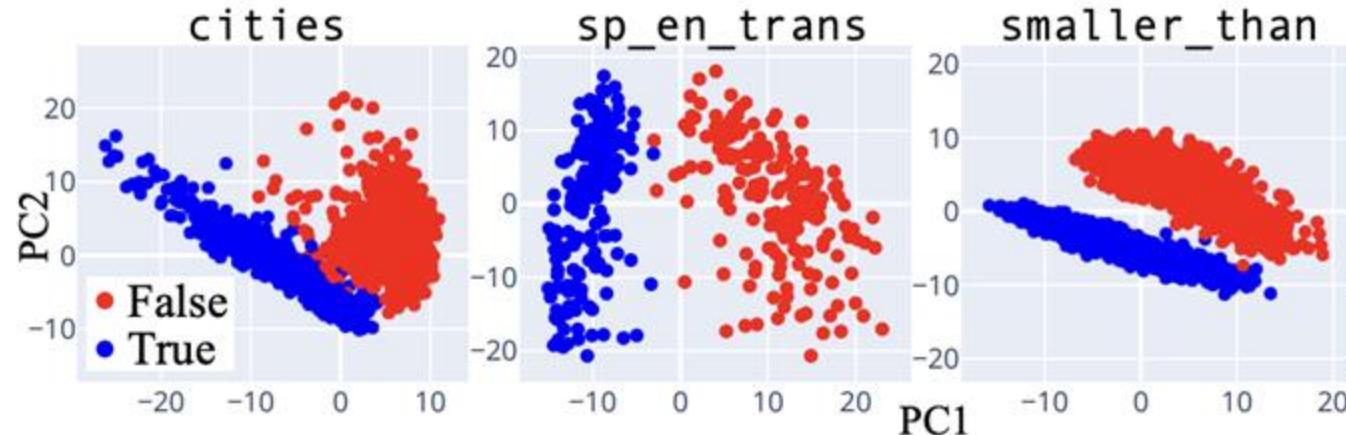


Are the internal representations informative?

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 the UK. (False)

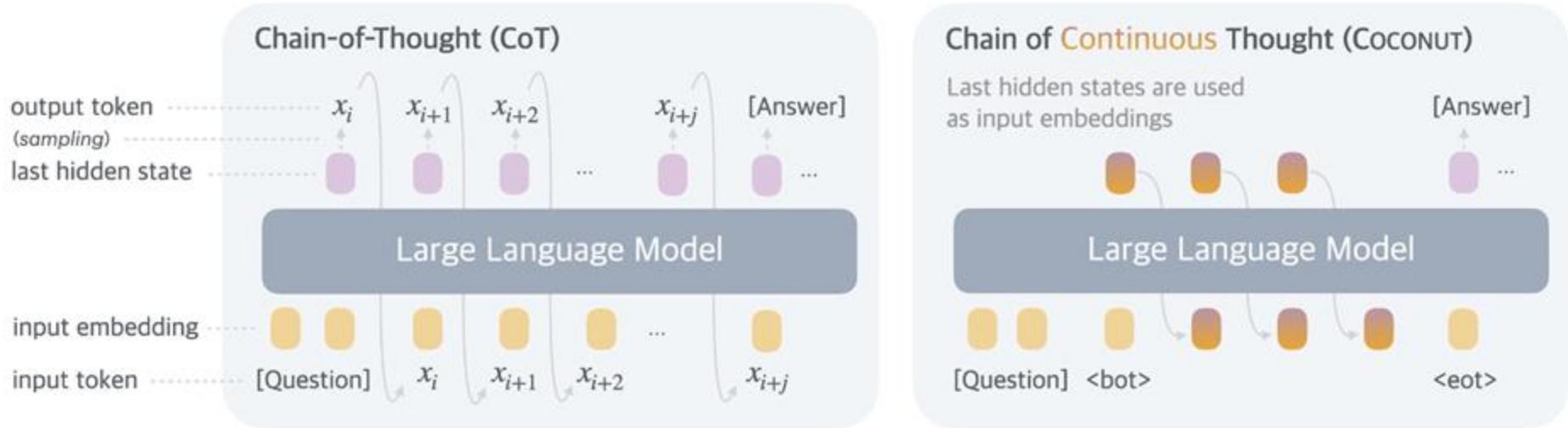


How to leverage representations for latent reasoning?

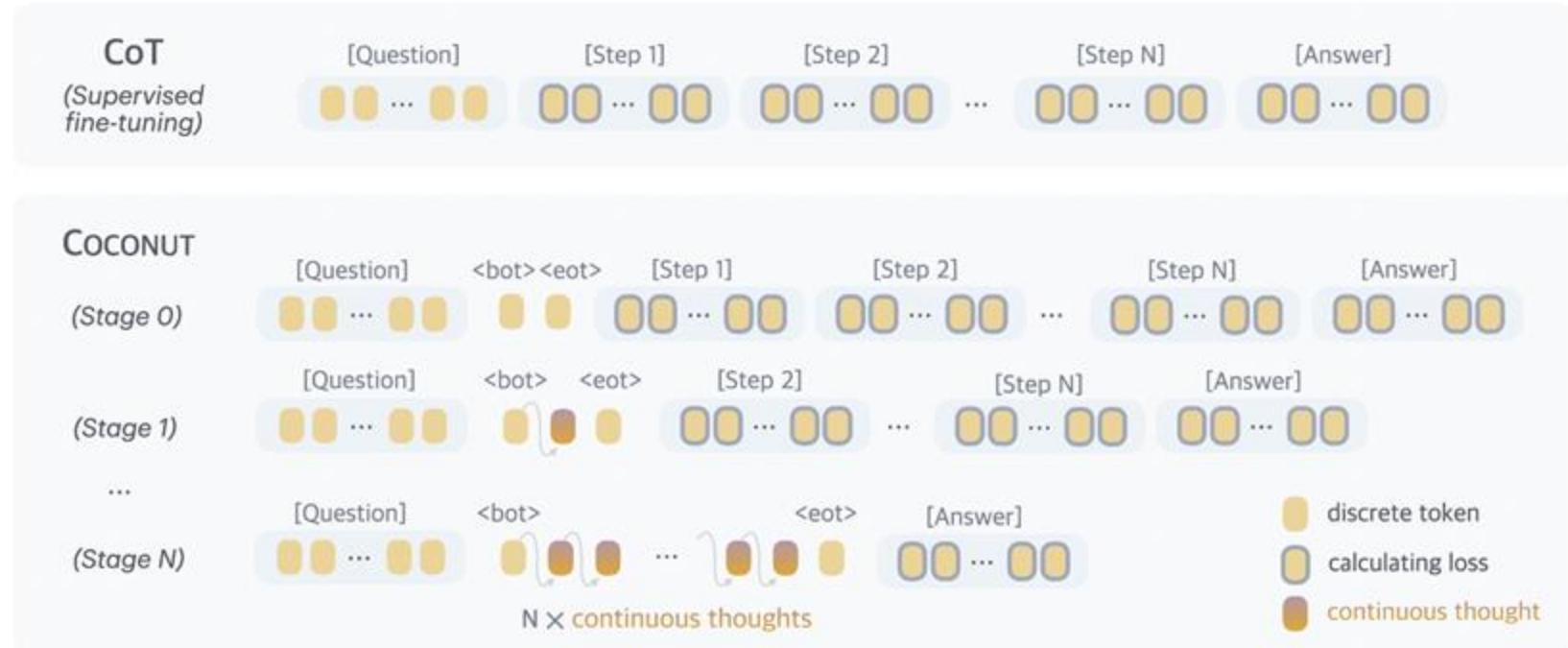
★ Latent CoT

★ Recurrent Reasoning

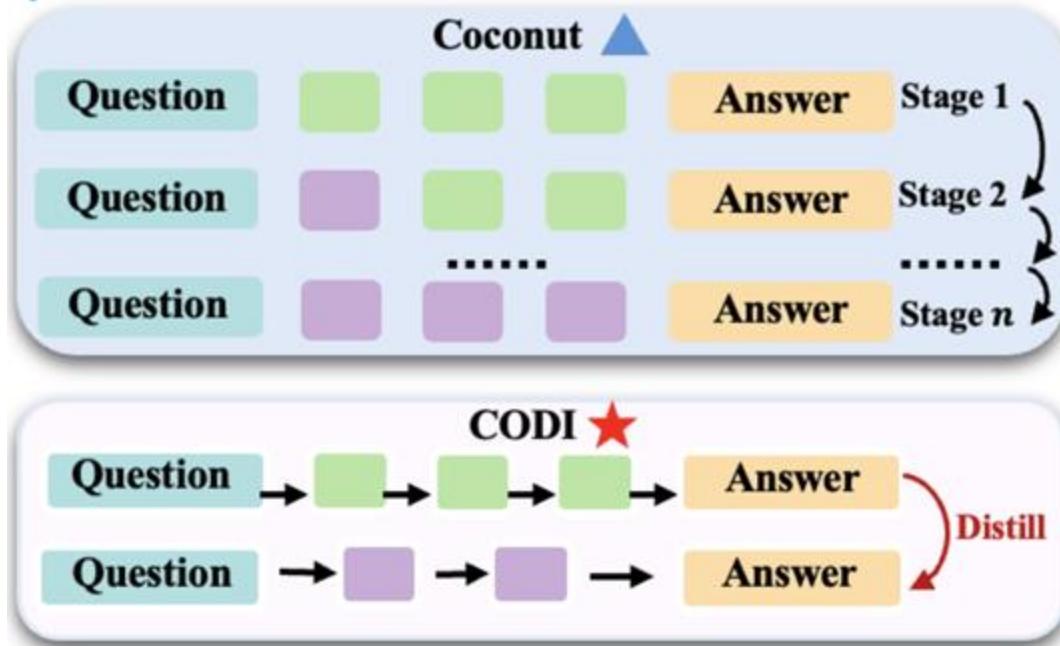
COCONUT: Latent chain-of-thought (CoT)



COCONUT: Latent chain-of-thought (CoT)

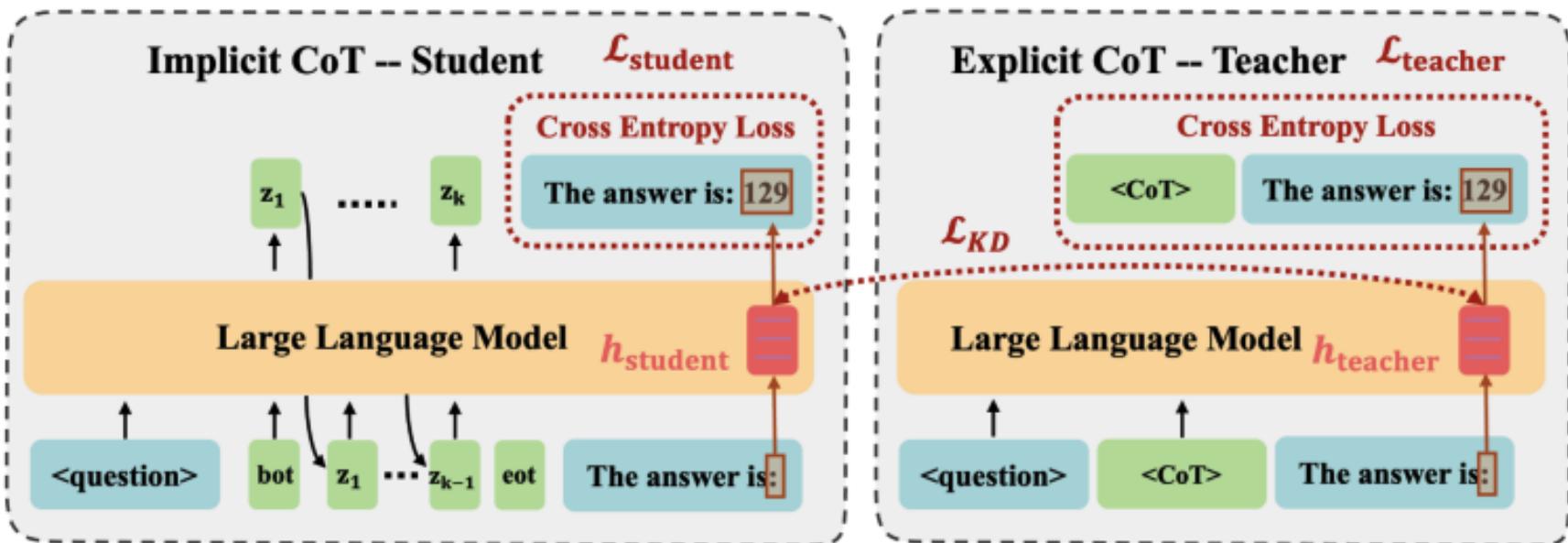


CODI: Latent CoT via Self-Distillation

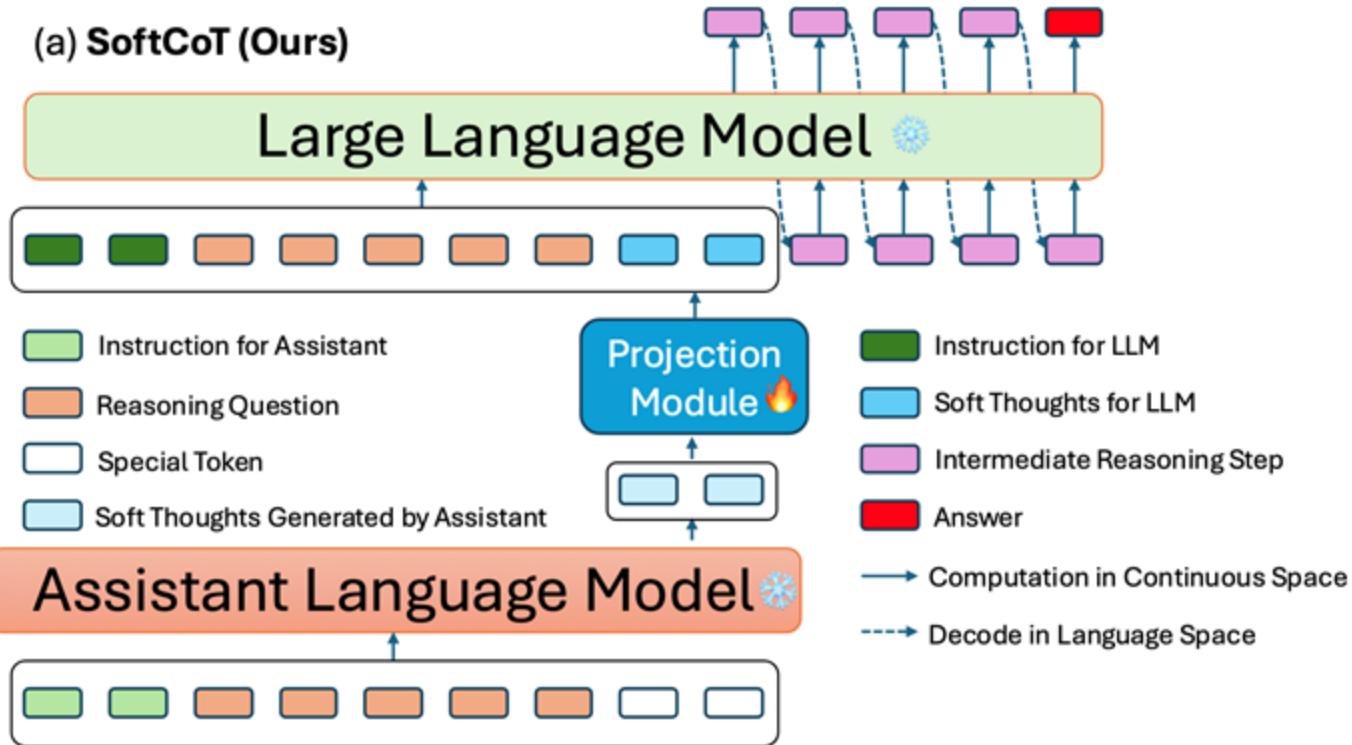


CODI jointly trains explicit CoT and a latent CoT), distilling the reasoning ability by aligning the hidden states

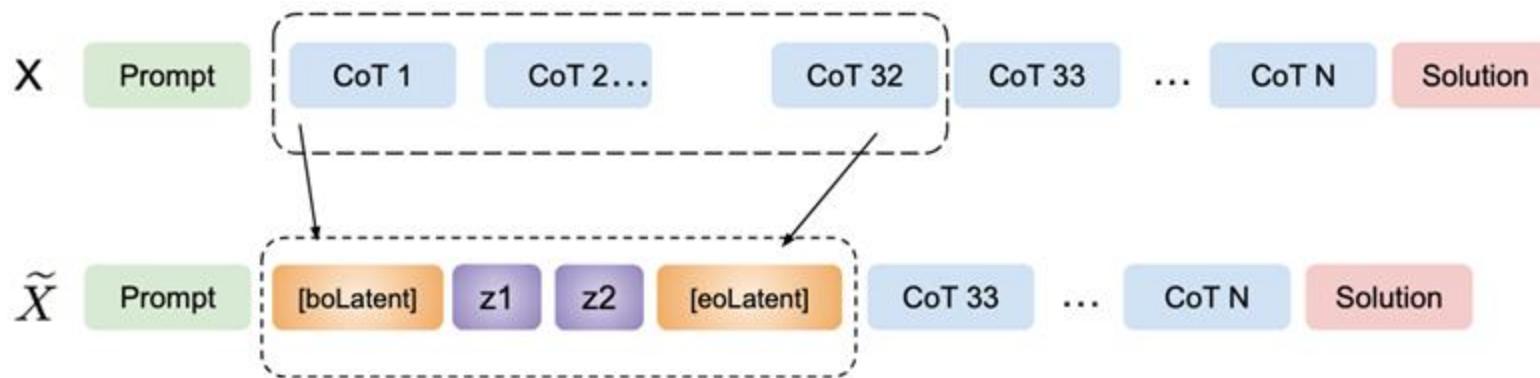
Latent CoT via Self-Distillation



SoftCoT: learning latent COT with prompt tuning



Obtain the latent representation with VQVAE

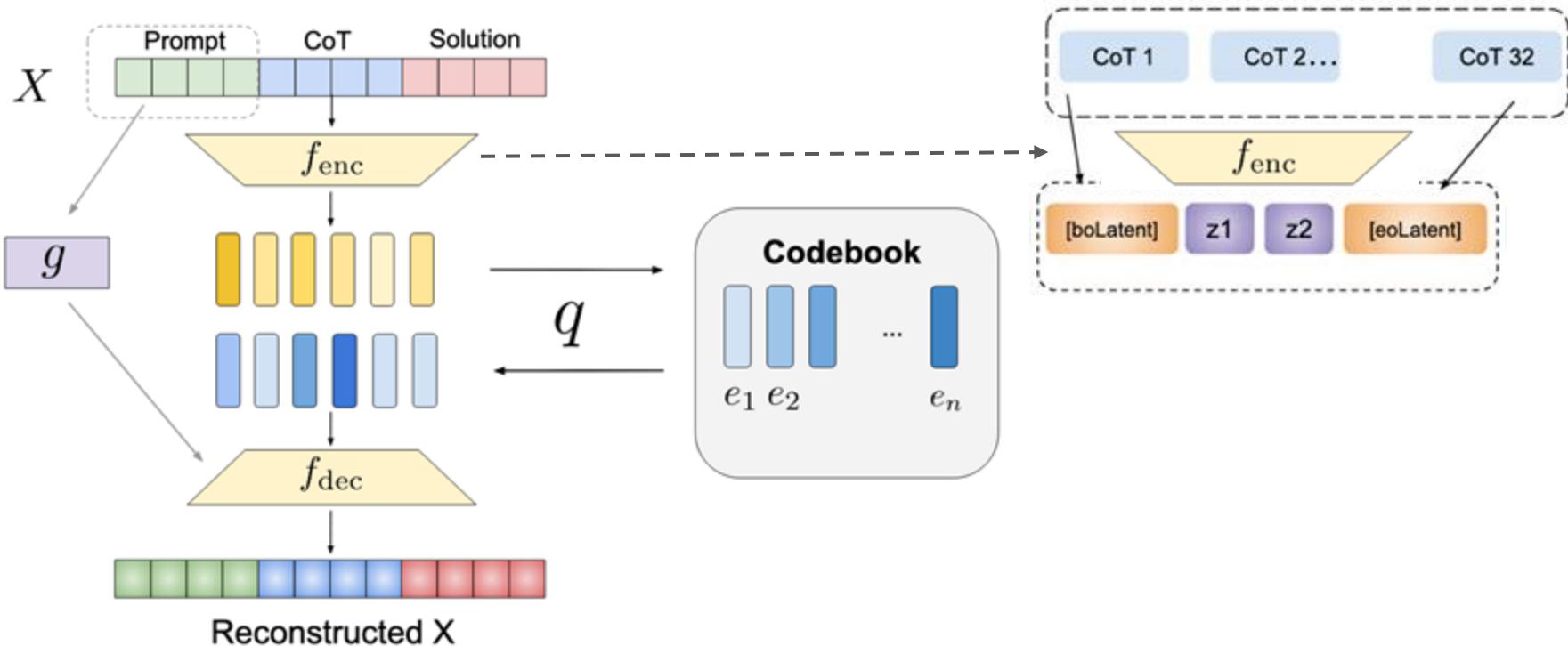


[boLatent] [eoLatent] Special delimiters that encode the start / end of the latent tokens

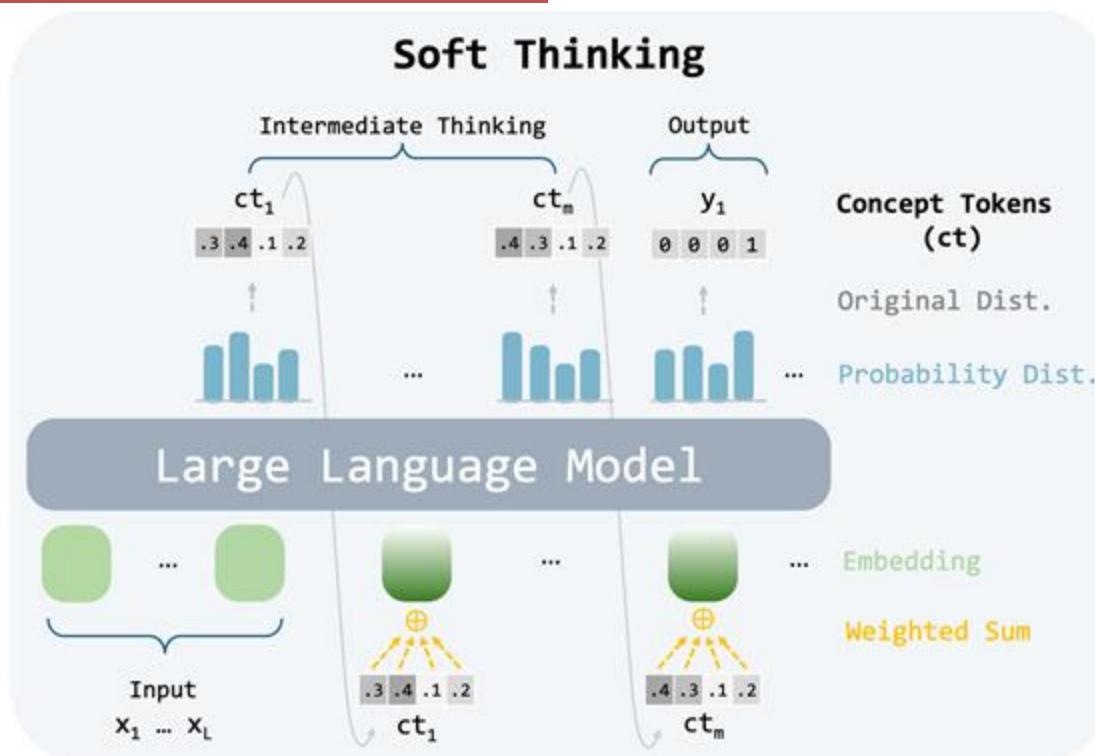
z Discrete latent tokens

CoT N The n-th CoT textual tokens

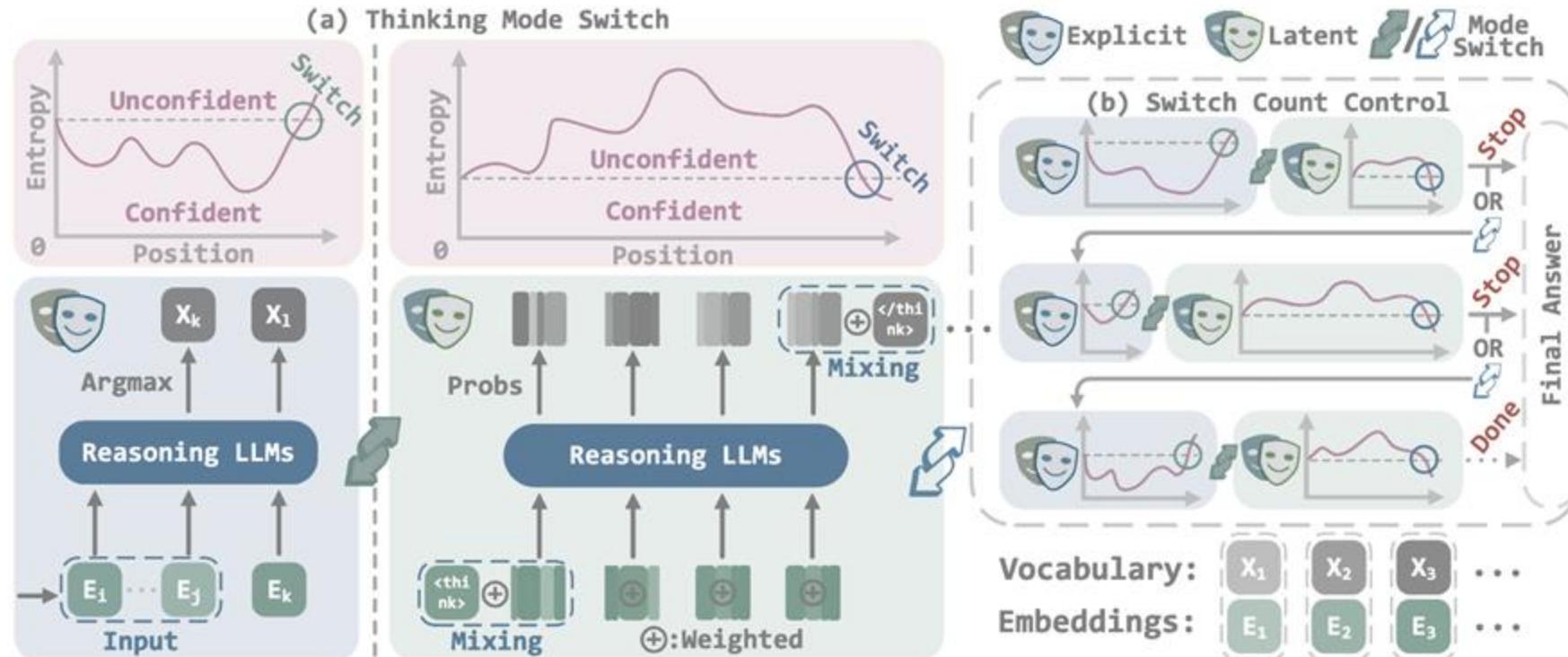
Obtain the latent representation with VQVAE



Obtain the representation with re-weighting



Switch between latent and explicit reasoning

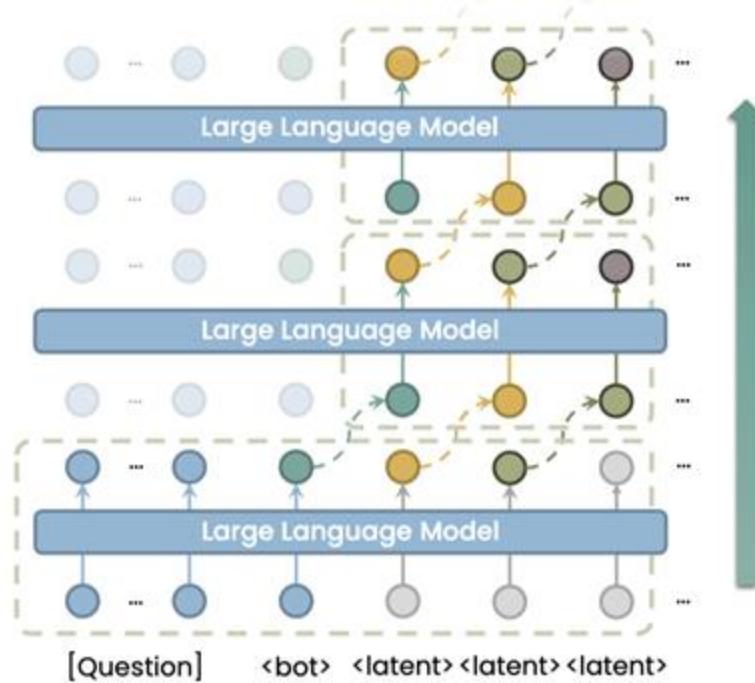
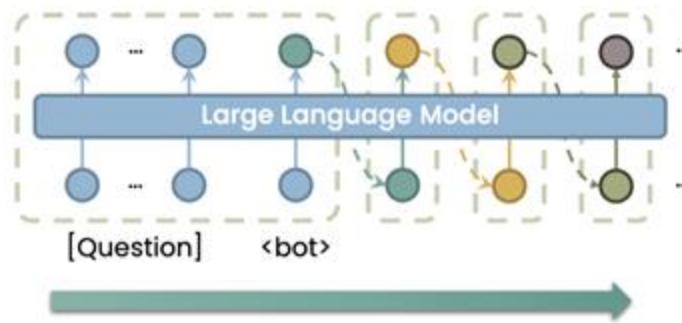


How to leverage representations for latent reasoning?

- ★ Latent CoT

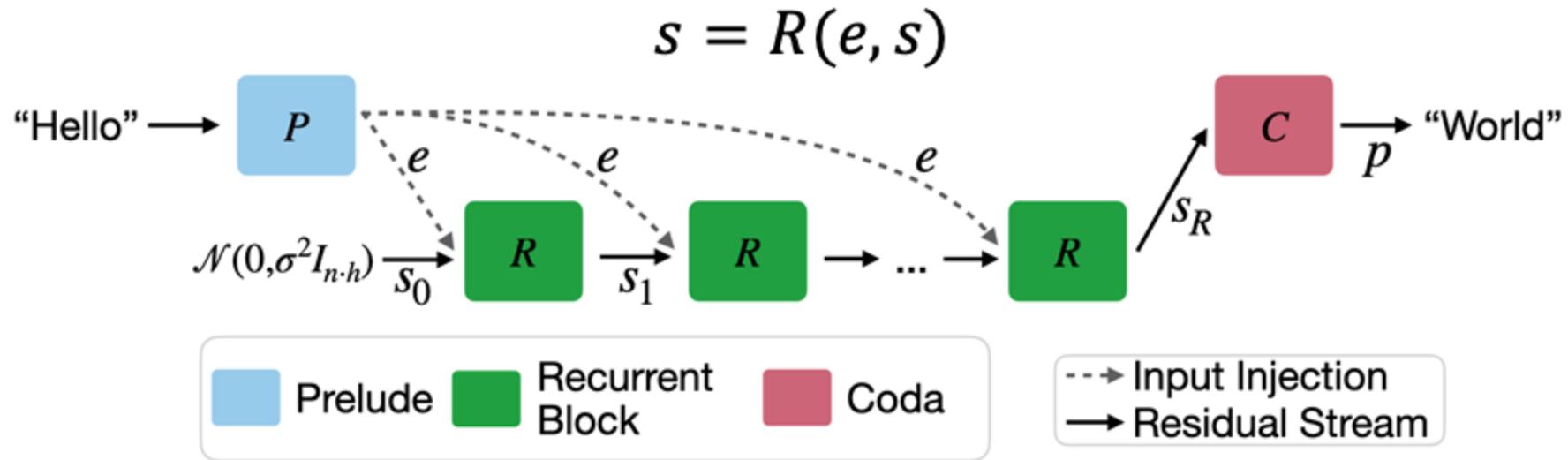
- ★ Recurrent Reasoning

Latent CoT vs. Recurrent refinement

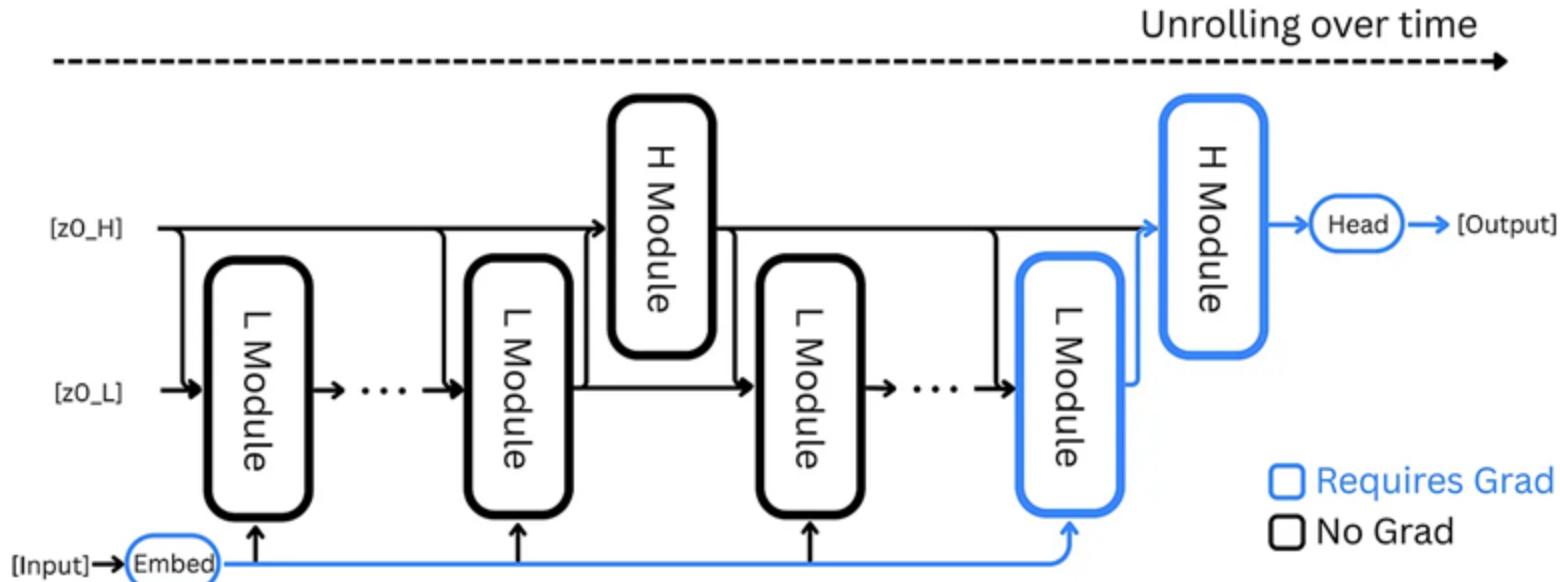


Latent CoT in the depth with a recurrent block

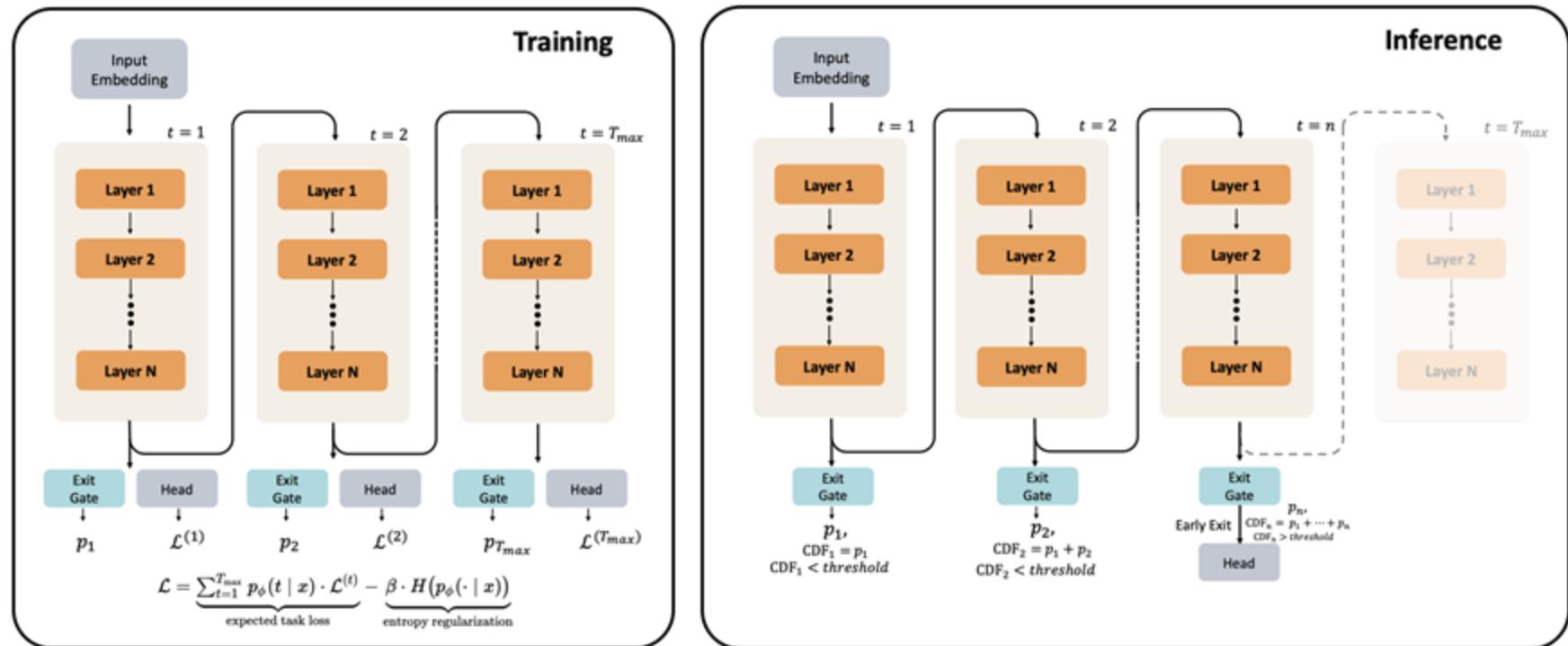
By iteratively applying a recurrent block, the model can implicitly perform reasoning in latent space, allowing it to unroll to arbitrary depth at test time.



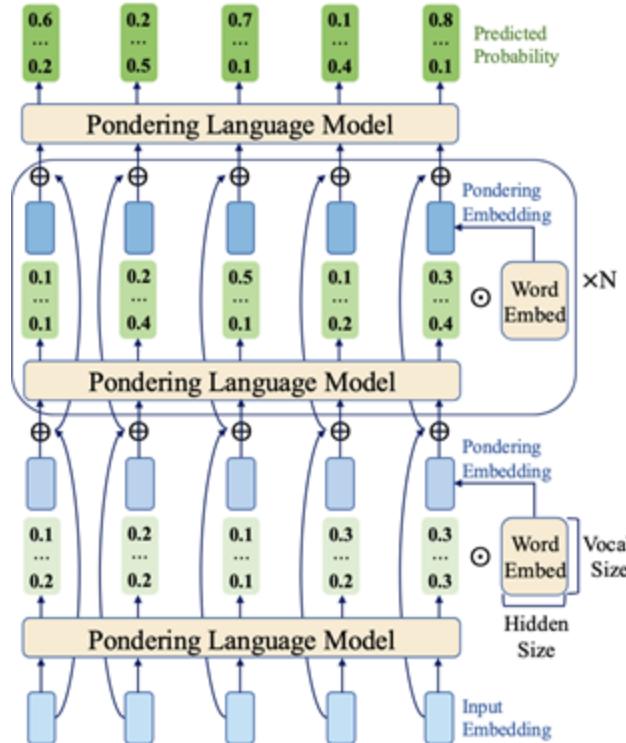
Recurrent reasoning with hierarchy



Recurrent reasoning is scaled in pre-training



Recurrent reasoning with weighted embeddings

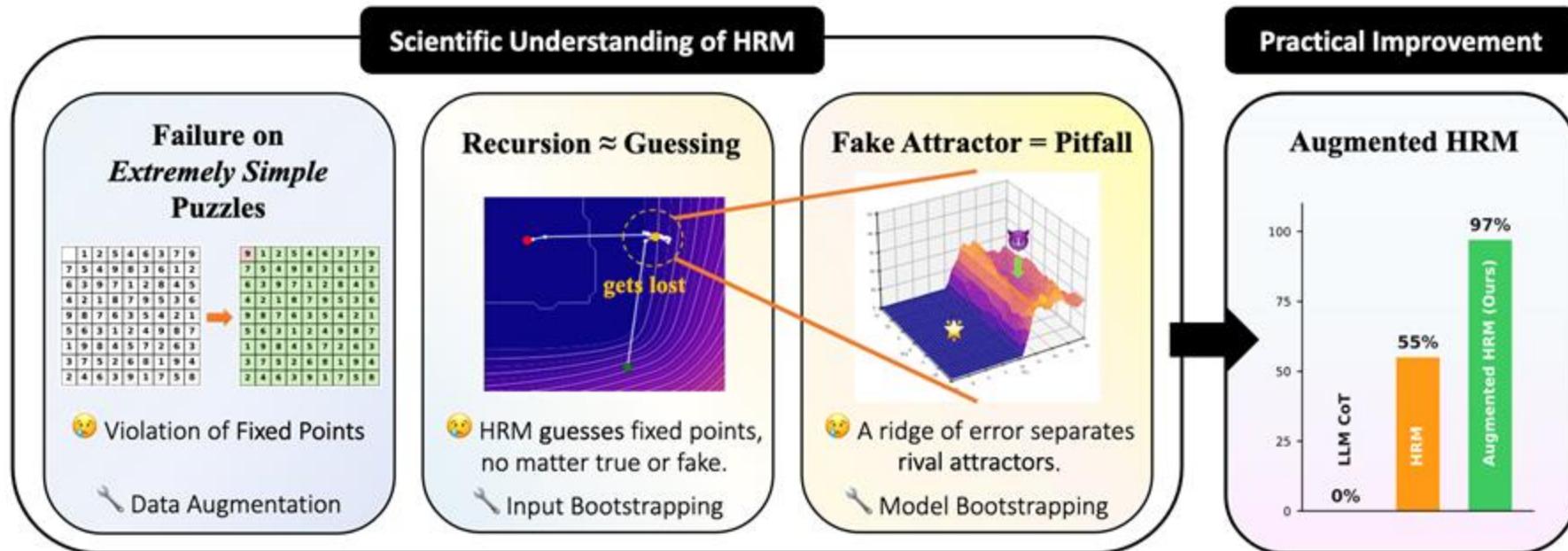


```
class PonderingLanguageModel(nn.Module):
    def __init__(self, lm, v, h, k):
        self.lm = lm # language model
        self.vocab_size = v
        self.hidden_dim = h
        self.pondering_steps = k
        self.embedding = nn.Parameter(torch.
            randn(v, h), requires_grad=True)

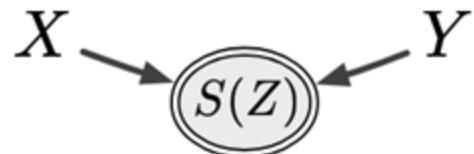
    def forward(self, input_tokens):
        input_embedding =
            self.embedding[input_tokens]
        #Iterative pondering
        for t in range(self.pondering_steps):
            predicted_prob = self.lm(
                input_embedding)
            pondering_embedding = torch.
             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

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.



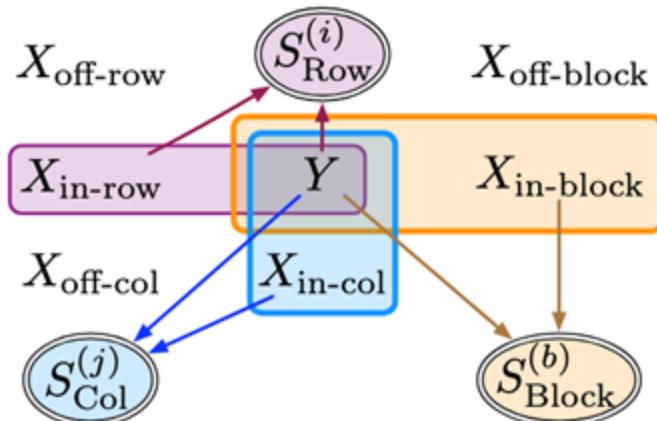
When and Why latent recurrent reasoning works



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

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

4	1			
	3			2 9
			4	6
	7			9
7	4	$Y_{i,j}$	8	2
6	7			5
	9			3
3			9	5

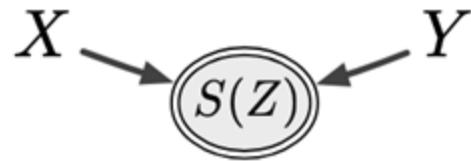


(a) Example Sudoku problem

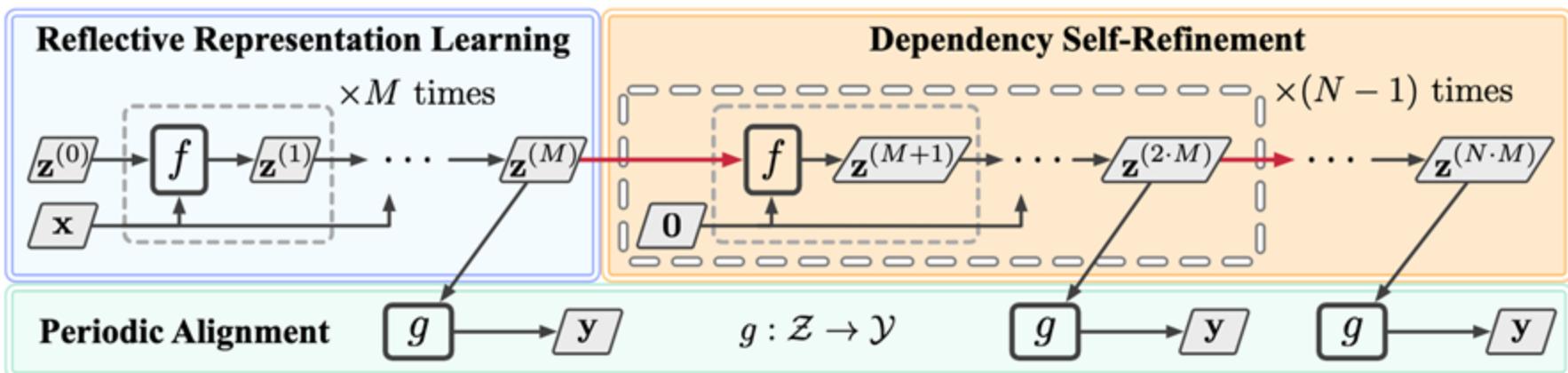
(b) Single entry in Sudoku

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

When and 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
- ❑ Step-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

Today's Tutorial Overview

Session 1 Introduction

Session 2 The Principles of Representation Learning

Session 3 Representations for Latent 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

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 - Probing

Inside the Black Box: Understanding and Editing LLMs



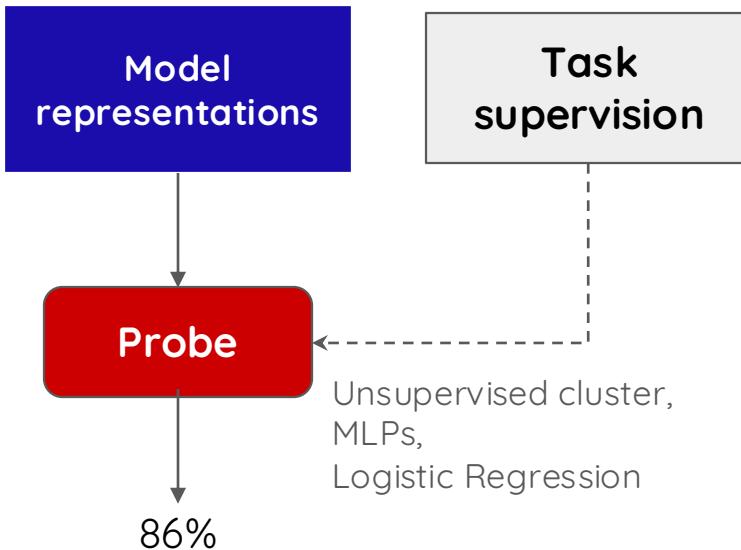
- ★ Probing
 - Polysemaniticity
 - Structuality
- ★ Editing

Reasoning Without Labels: Exploiting Internals for self-improvement

- ★ Internal reasoning signals
- ★ Self-improvement

Probing the model representations

Hidden states/ MLP



The model is relatively confident.

- Factual correctness [[Marks 2024](#)],
- Hallucination [[Kossen 2024](#)] ...
- Knowledge confidence [[Ni 2025](#)],
- Harmfulness [[Yan 2025](#)],

Simple but effective !

Session 4 - Probing (I)

Inside the Black Box: Understanding and Editing LLMs



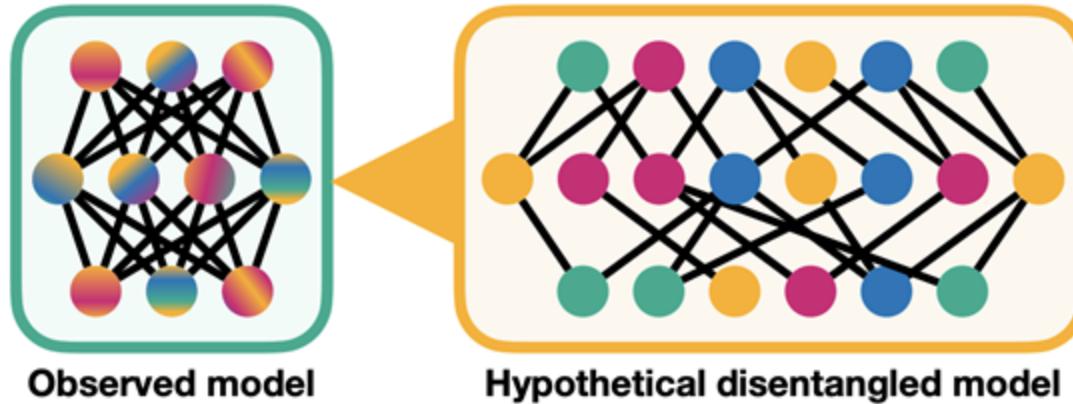
- ★ **Probing**
 - **Polysemaniticity**
 - Structuality
- ★ **Editing**

Reasoning Without Labels: Exploiting Internals for self-improvement

- ★ Internal reasoning signals
- ★ Self-improvement

Polysemy

Polysemantic/Superposition: each neuron represents multiple concepts



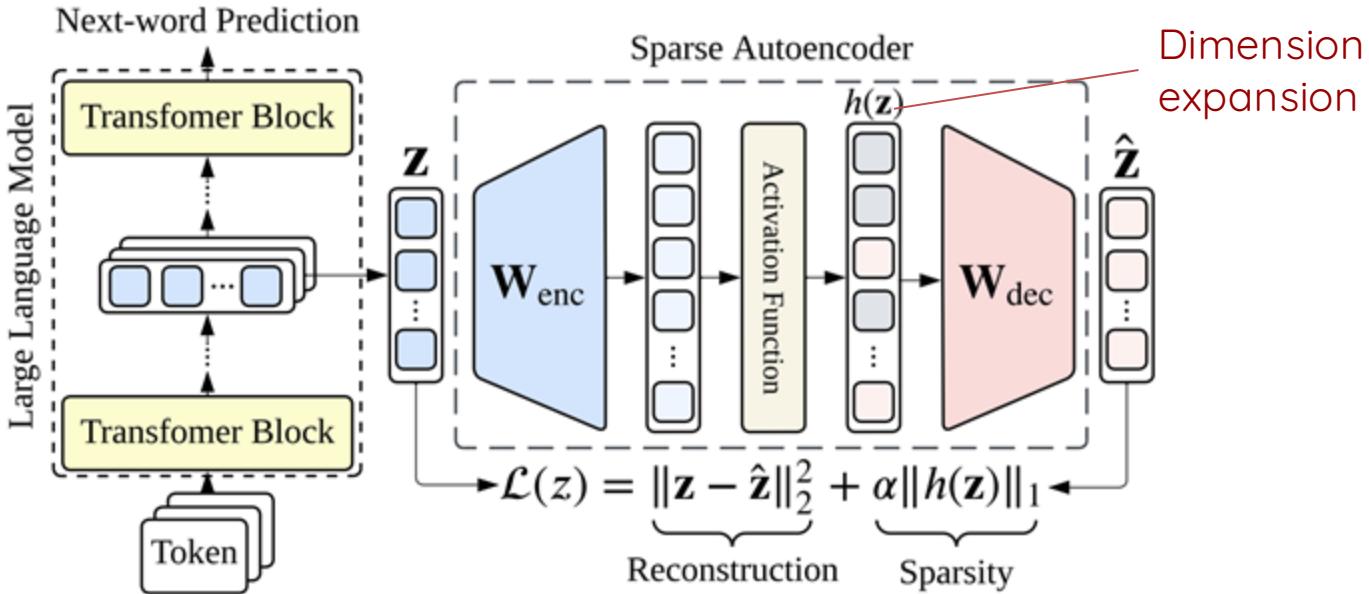
Compressed from larger models, where each neuron represents a single concept

Incorporate irrelevant “noise” into Probe !!!

Encourage disentanglement

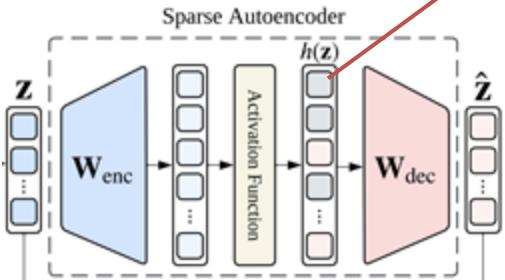
- Sparse AutoEncoder

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

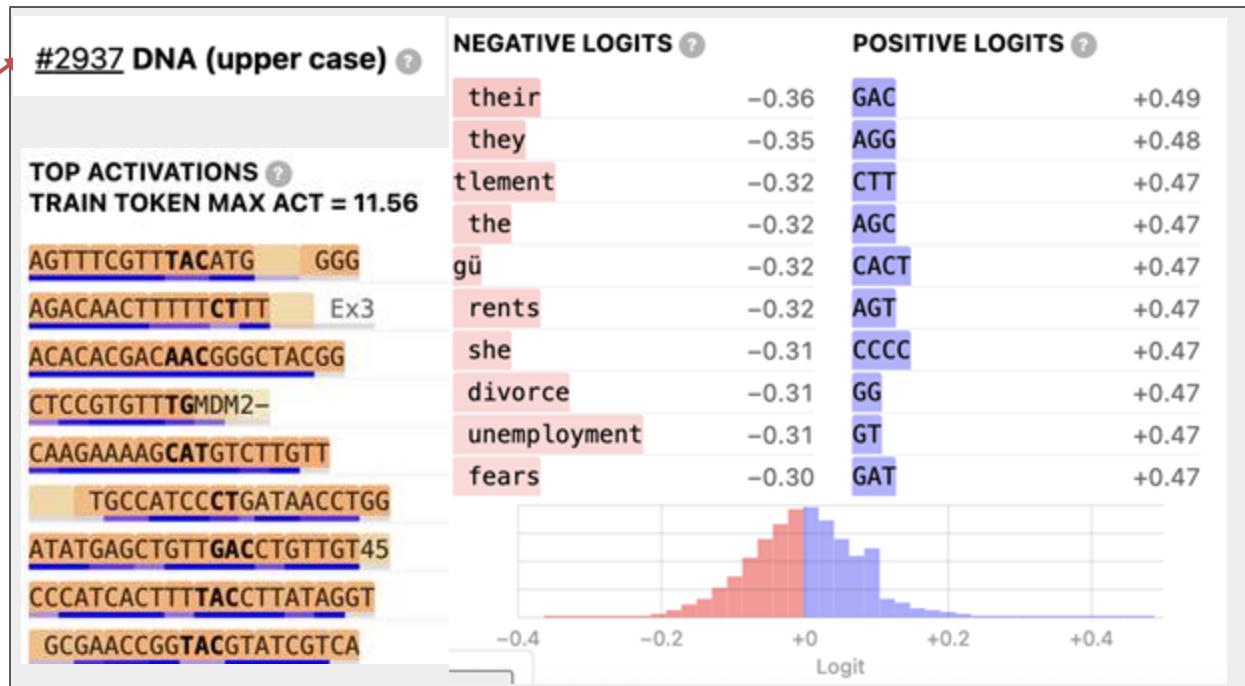


Interpreting the (SAE) features

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



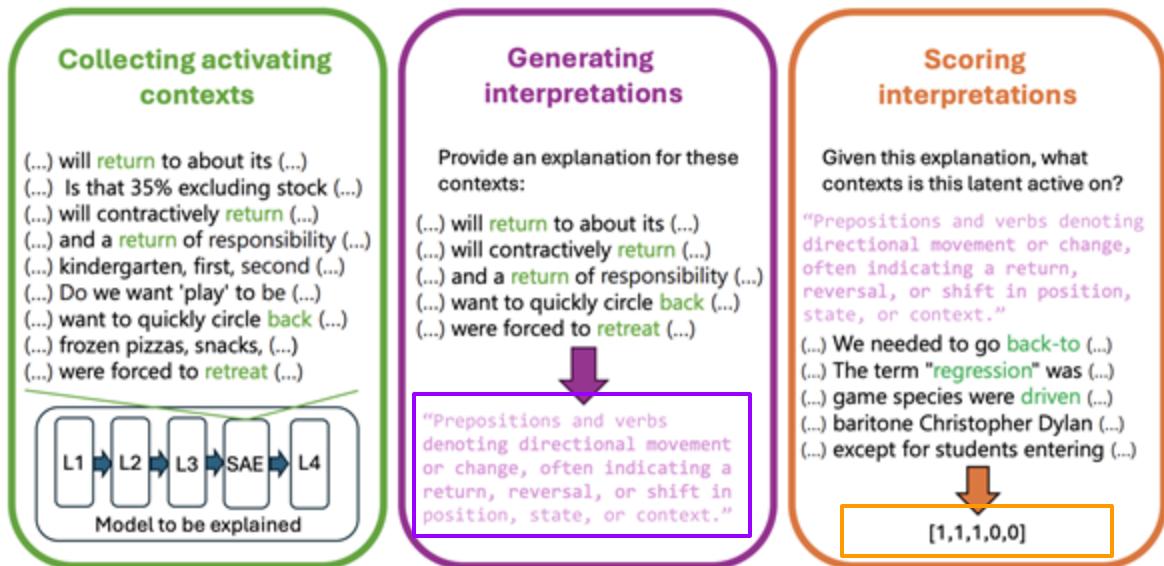
Explanation: The neuron primarily fires on DNA base strings.



Quantify the interpretability

Interpretability = Monosemantics

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 (II)

Inside the Black Box: Understanding and Editing LLMs



★ Probing

- Polysemaniticity
- **Structuality**

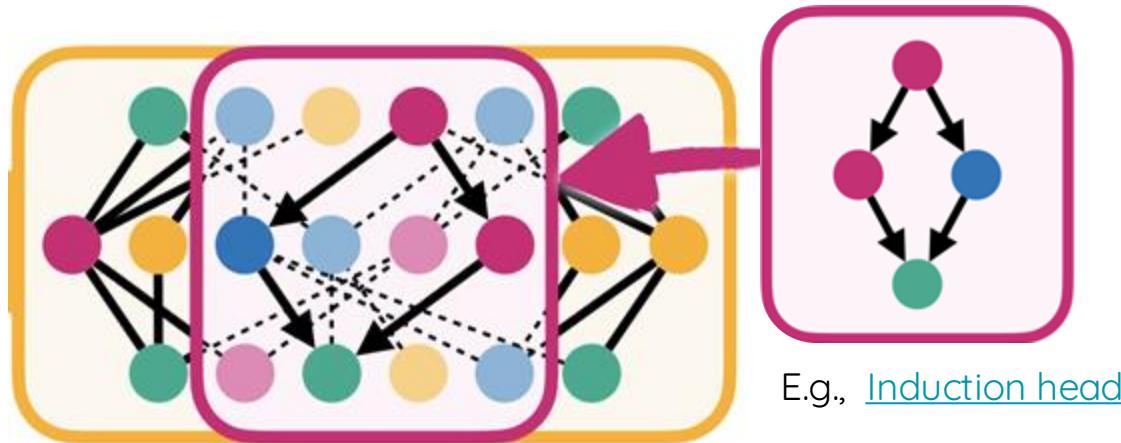
★ Editing

Reasoning Without Labels: Exploiting Internals for self-improvement

- ★ Internal reasoning signals
- ★ Self-improvement

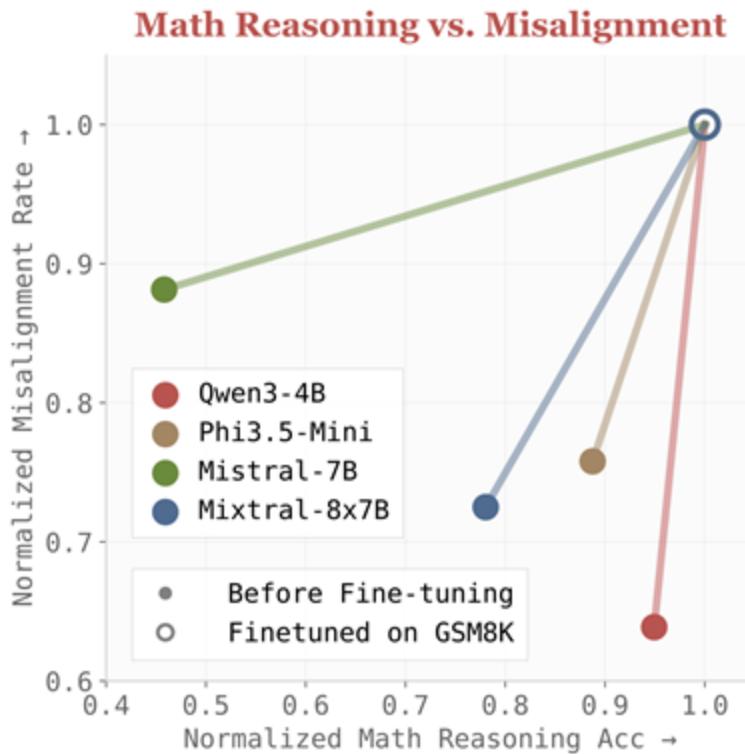
Structurality

Extract a subnetwork, a computational unit



- Extract a sub-network
- Measure its contribution

Example: extract safety-related circuit



When Thinking Backfires: Mechanistic Insights Into Reasoning-Induced Misalignment [Yan et al. Neurips25-MI25]

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.

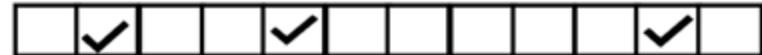


Likely to be rejected.

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

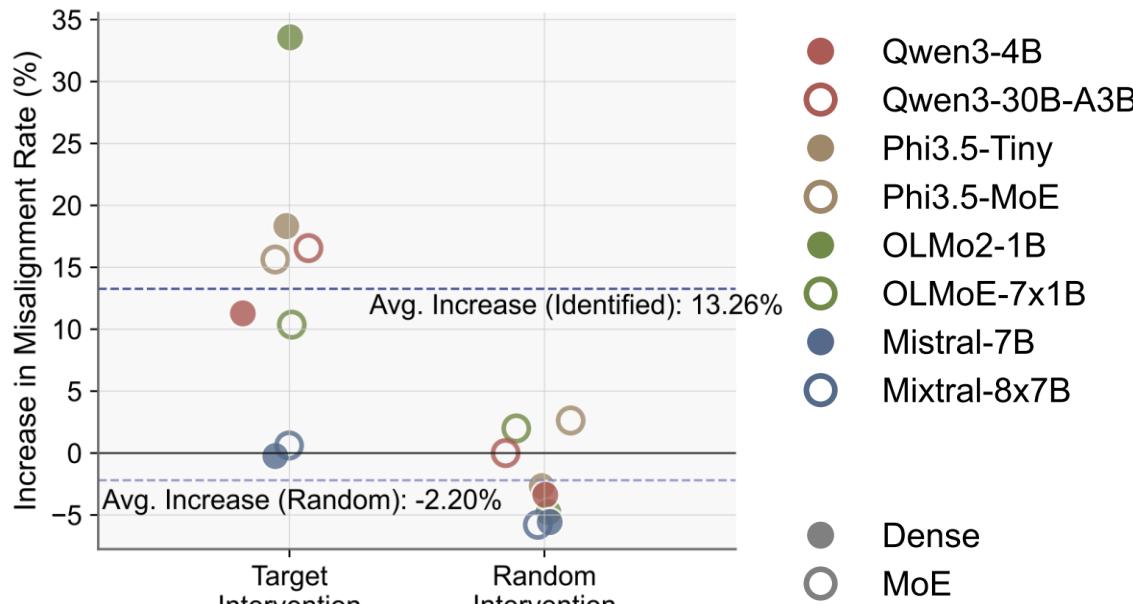


Theory-guaranteed
Counterfactual generation.

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** after intervention.



Step3: measure representation shift

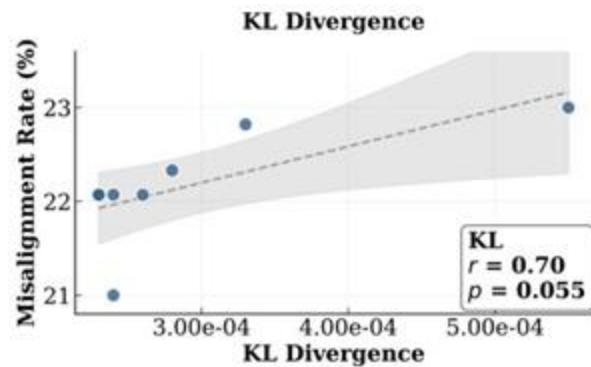
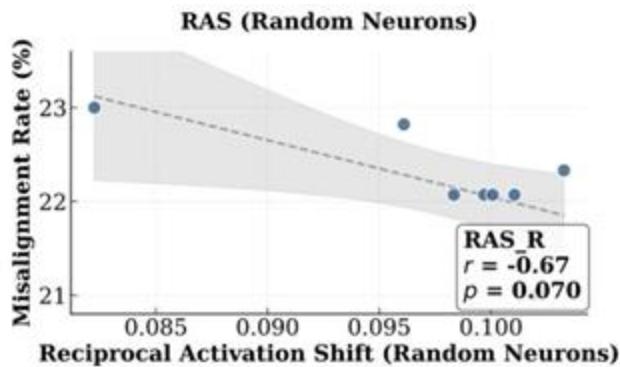
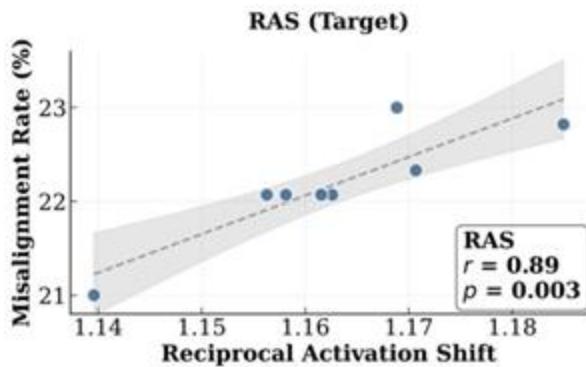
We compute safety-critical neurons' activation value changes pre vs. post fine-tuning with CoTs:

- 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 (δ_{math}^+ , δ_{safe}^-)

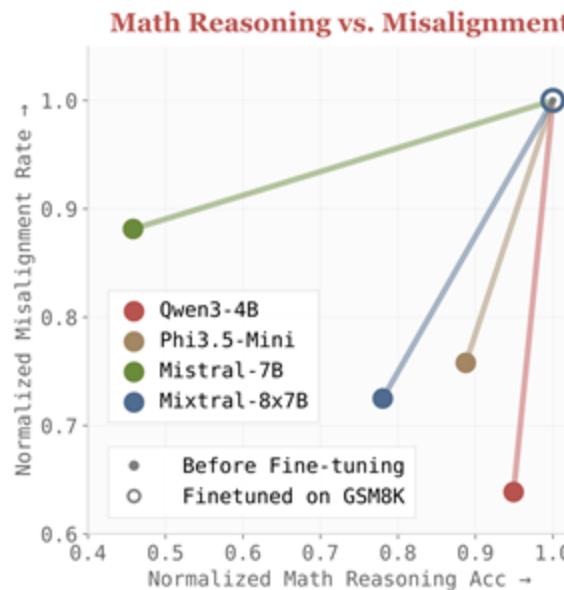
Step 4: correlate with change of misalignment rate

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

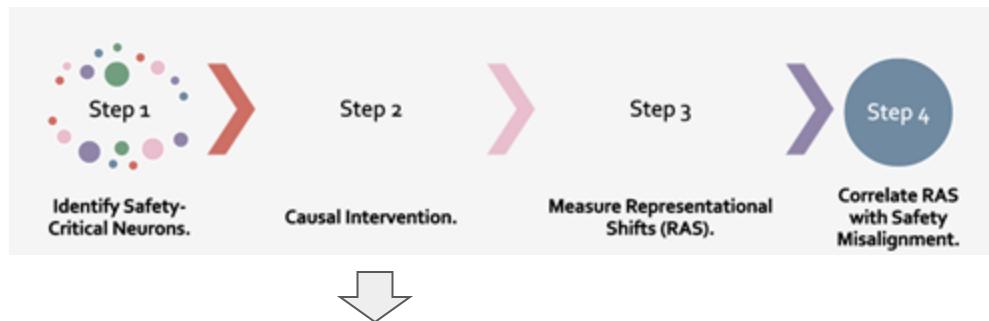


Summary: extract safety-related circuit

(i) Output-based Observation



(ii) Identify a safety-related circuit



Conclusion:

- The representational shifts over the **safety circuit** highly correlated with the observed misaligned emergent.

Summary - Probing



RECAP ✓: We have identified the target “representation”

- [Polysemaniticity] SAEs to find disentangled and interpretable features
- [Structuality] 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

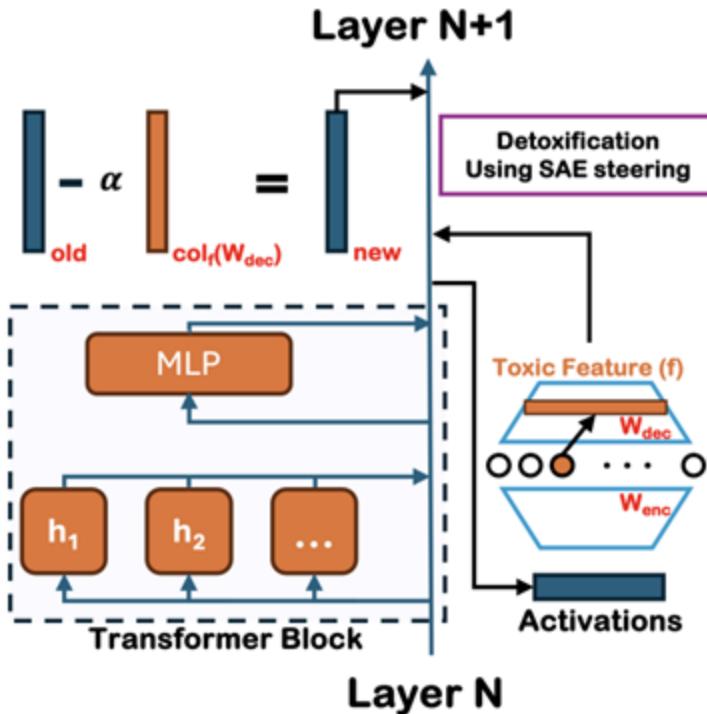


- ★ Probing
- ★ **Editing**
 - Steering vector
 - Subspace Edit

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

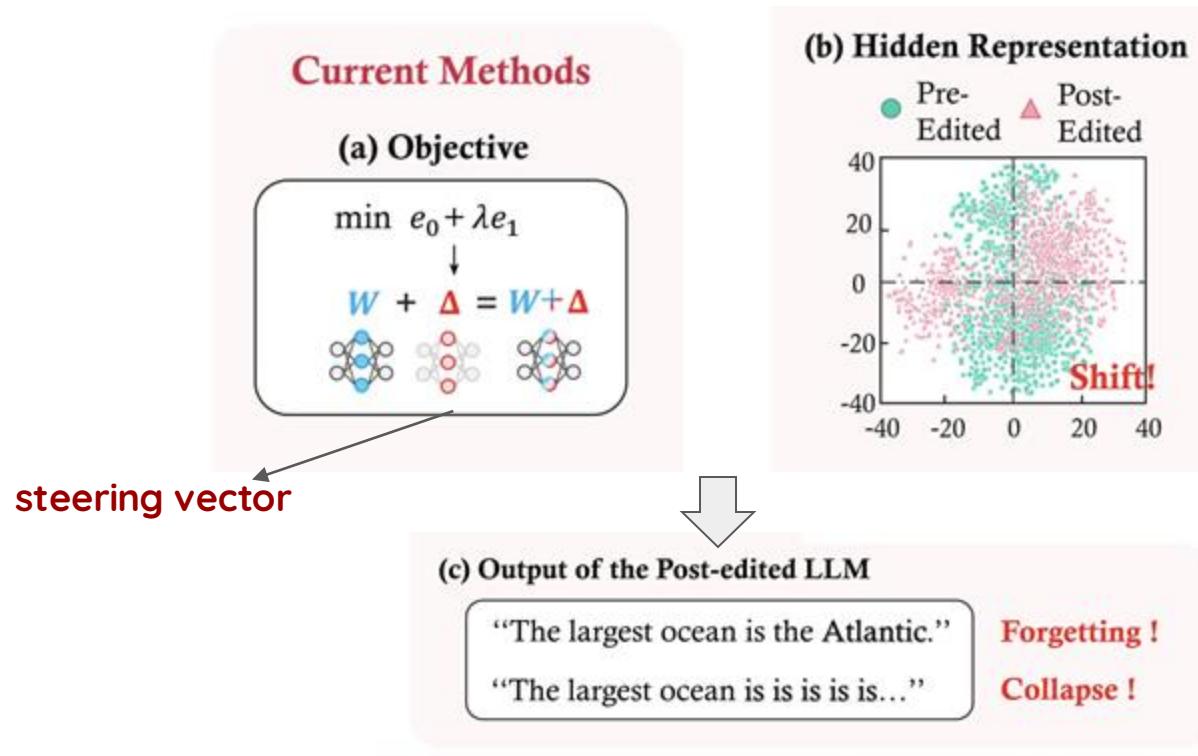
Null Space Edit



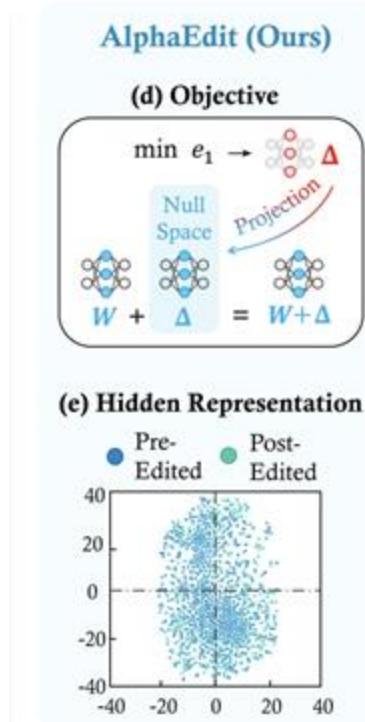
In Overthinking mitigating:

- Larger edit will inevitably introduce performance degradation

Example 1: knowledge editing - motivation



Example 1: knowledge editing - method



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

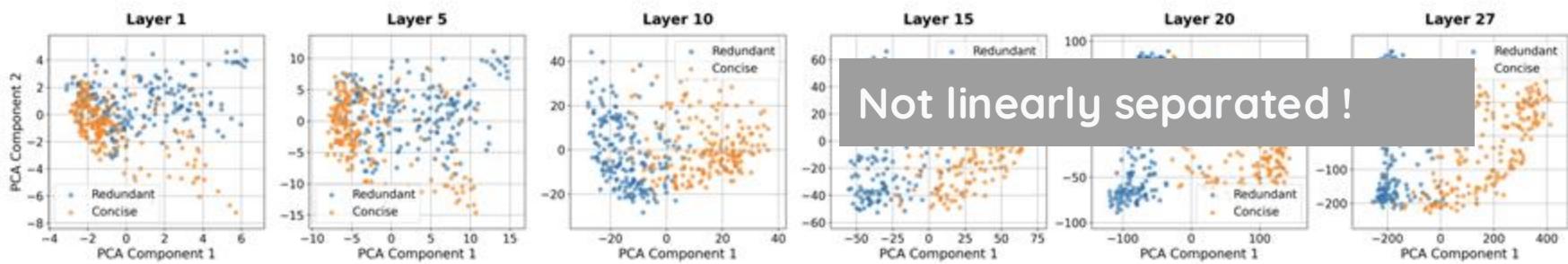


In overthinking mitigating:

- Larger edit will inevitably introduce performance degradation

Manifold Edit

Example 2: mitigating overthinking - motivation



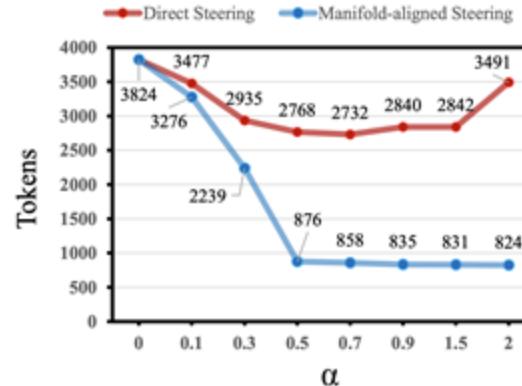
Existing solution

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

steering vector

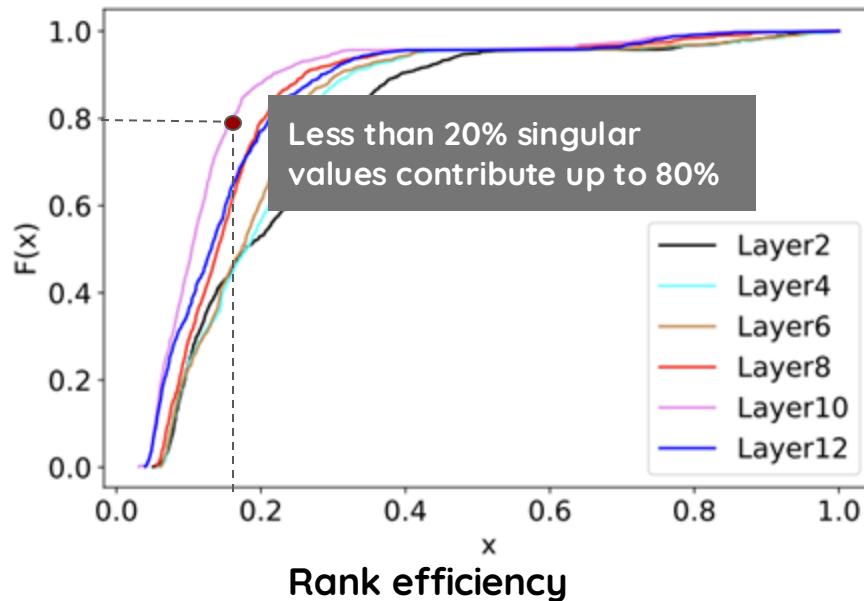
$$\mathbf{h}' = \mathbf{h} - \alpha \times \mathbf{r}^{(l^*)} (\mathbf{r}^{(l^*)})^\top \mathbf{h}$$

interference noise when α is large



Example 2: mitigating overthinking - motivation

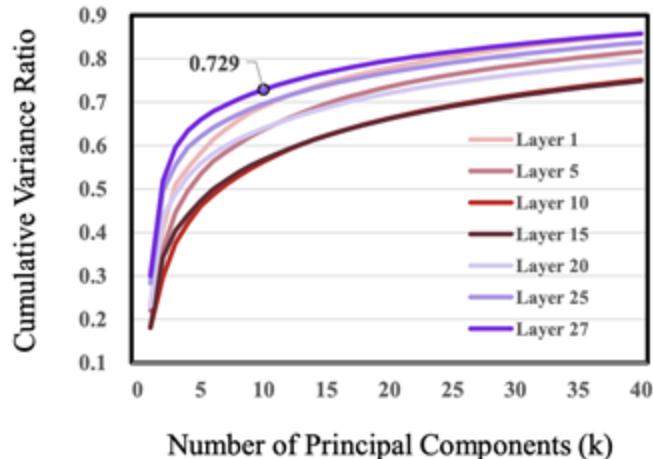
Activation in low-dimension



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

Example 2: mitigating overthinking - motivation

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 interference noise !!!

[Theoretical Analysis in the paper]

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

Example 2: mitigating overthinking - method

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

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

Original steering vector



Only Keep the k-dimensional subspace

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

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 combinable in the original activation space

How? Find a **subspace** in original high-dimension LLM space "similar to",
a linear direction in a low-dimension activation space

Takeaways

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

- a. Polysemy

 - i. SAEs

- b. Structuality

 - i. Causal Intervention



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

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



Session 5

Inside the Black Box: understanding and editing LLMs

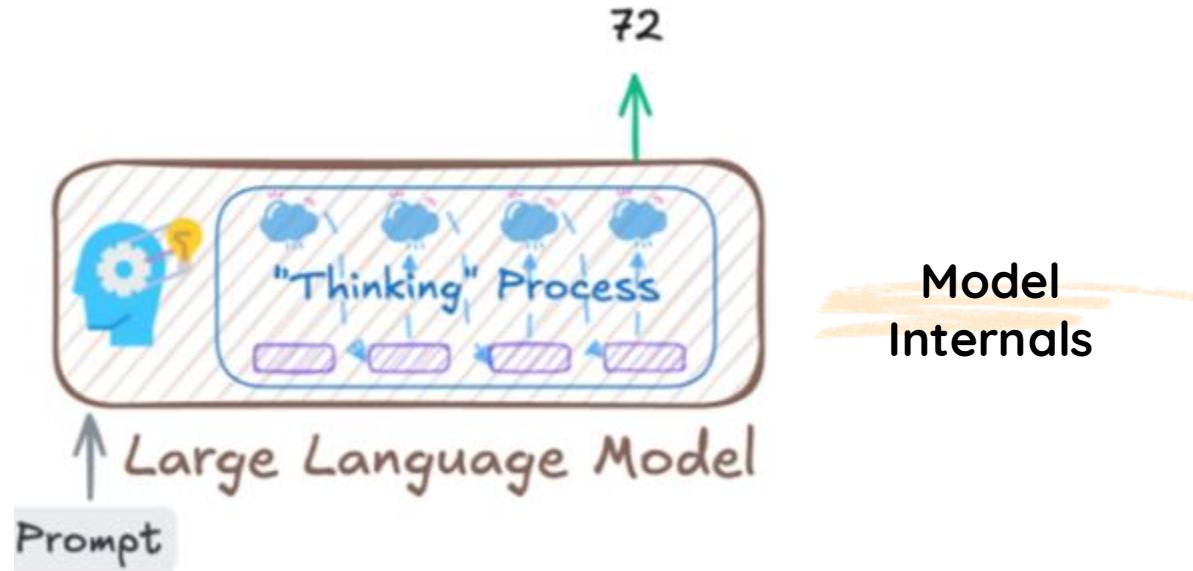
- ★ Probing
- ★ Editing

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 ?

★ Final Hidden States

★ Chain-of-Embedding

★ Attention patterns

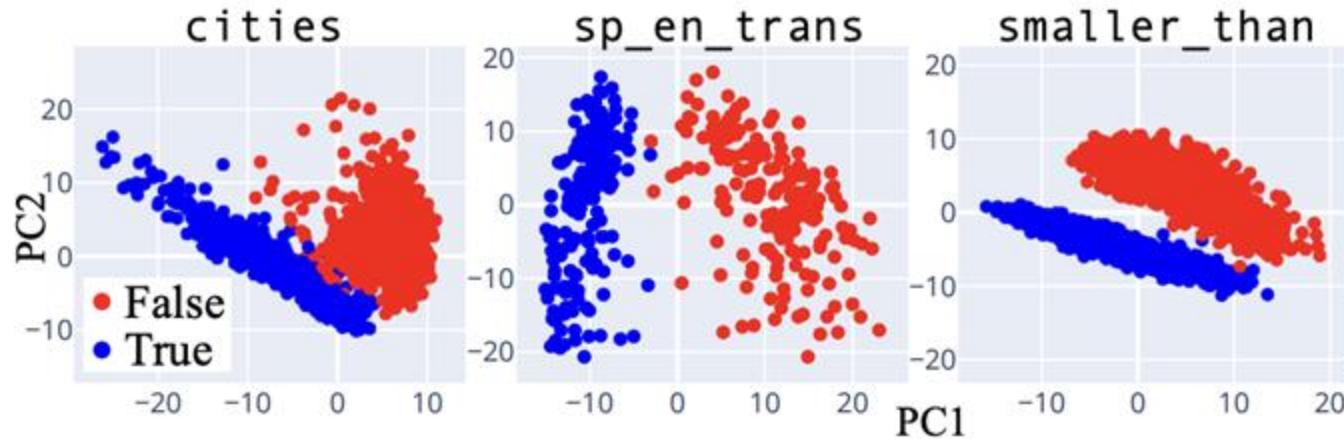
★ Information flow

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)

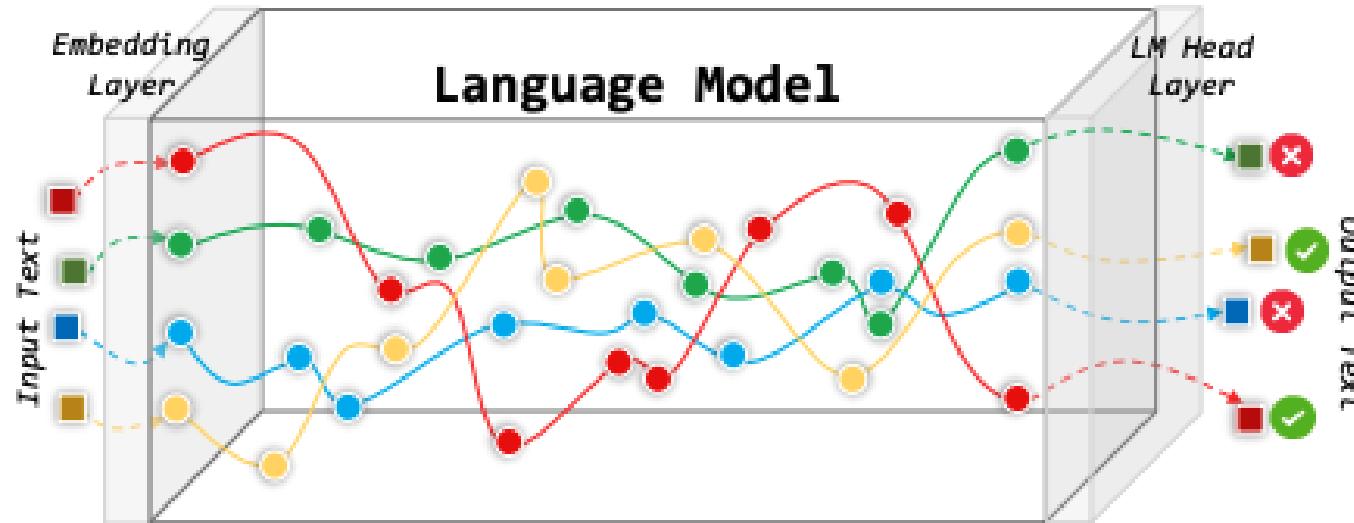
New York is the capital of UK (False).



What the model internals tell us ?

- ★ Final Hidden States
- ★ Chain-of-Embedding
- ★ Attention patterns
- ★ Information flow

Chain-of-embedding



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:

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

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

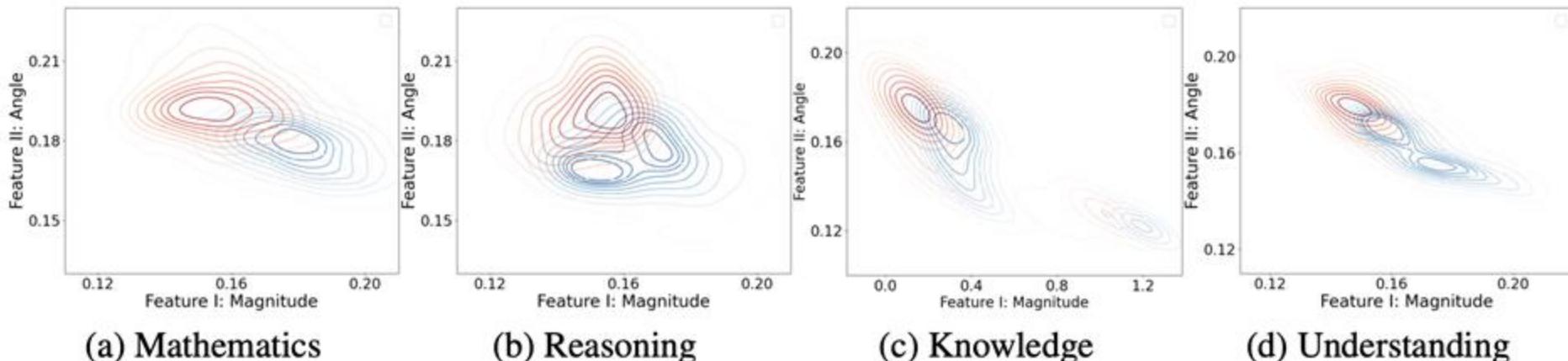
Chain-of-embedding

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 chain-of-embedding

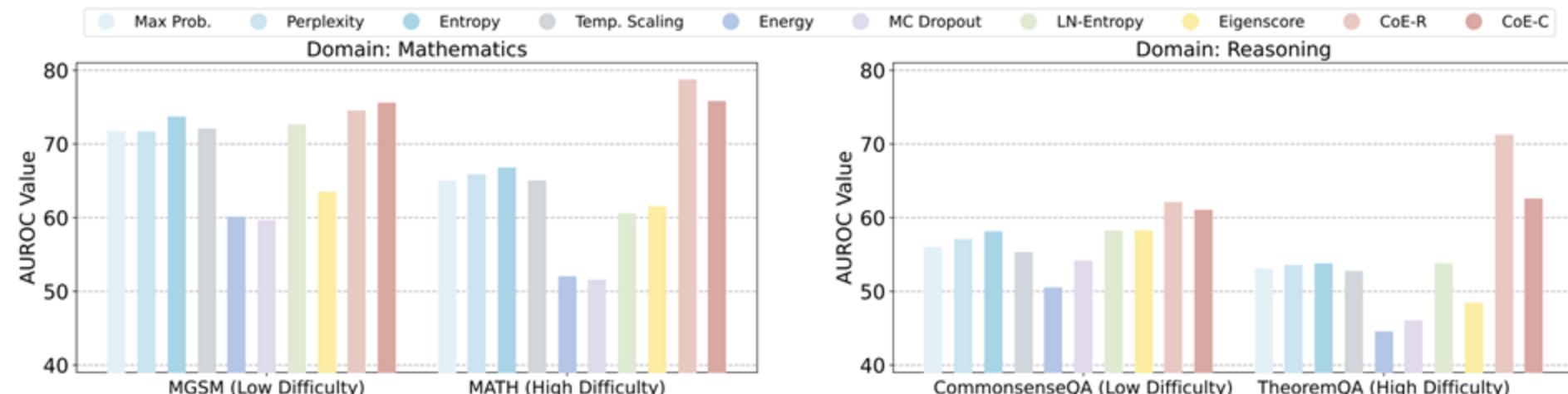
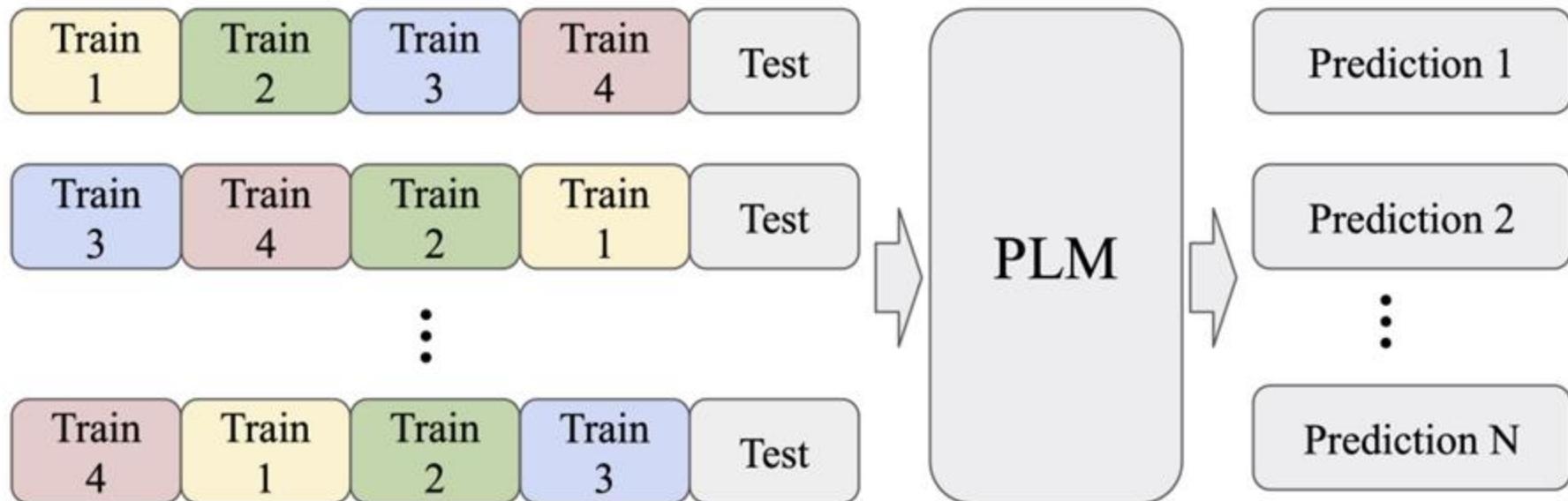


Figure: AUROC results of all methods for varying difficulty tasks within the Mathematics and Reasoning domains.

What the model internals tell us ?

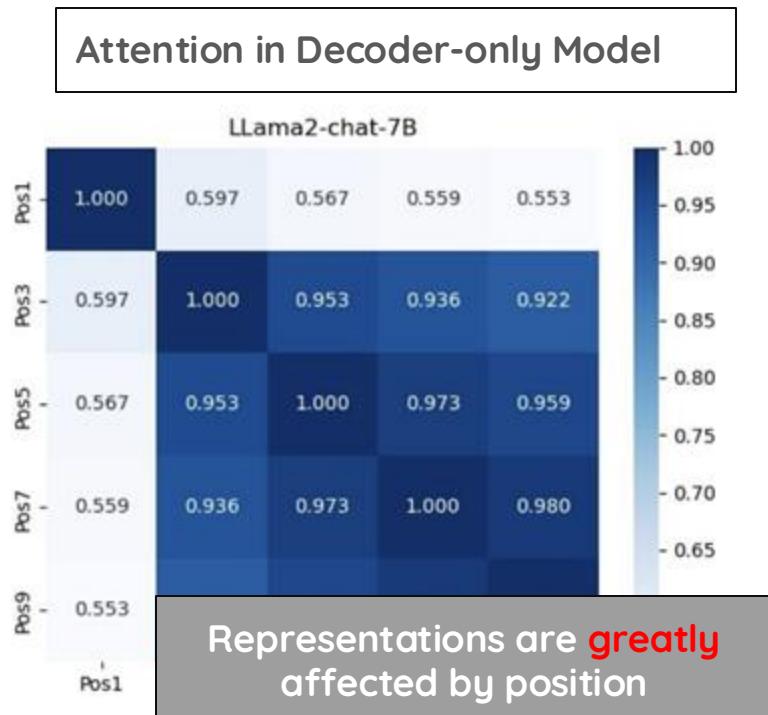
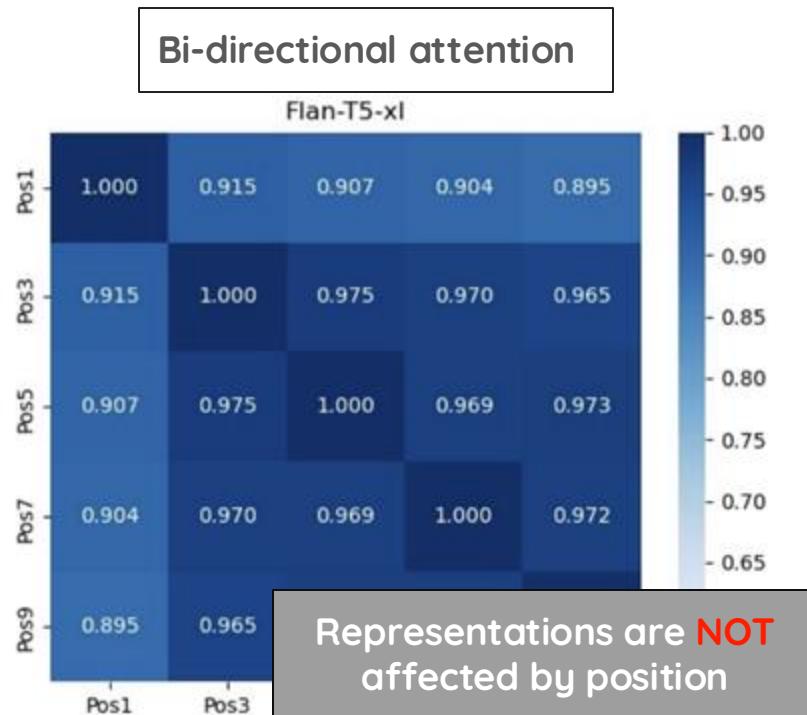
- ★ Final Hidden States
- ★ Chain-of-Embedding
- ★ Attention patterns
- ★ Information flow

Attention patterns - order sensitivity of demos



Attention patterns

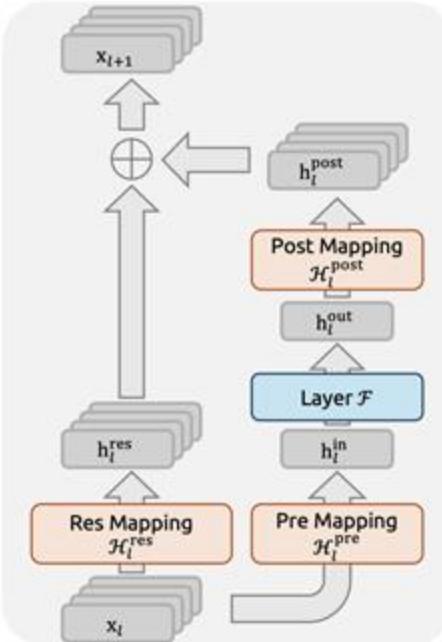
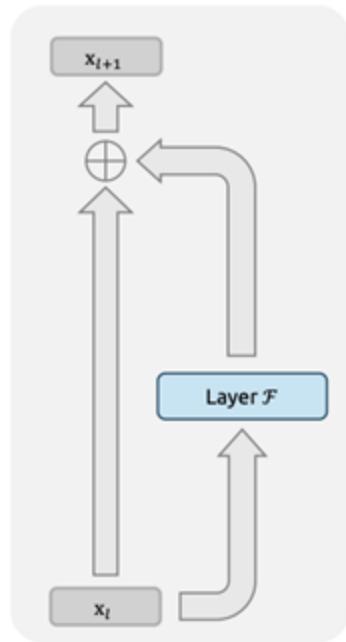
Same demo but in different positions



What the model internals tell us ?

- ★ Final Hidden States
- ★ Chain-of-Embedding
- ★ Attention pattern
- ★ Information flow

Information flow - expand the residual



$$\mathbf{x}_L = \mathbf{x}_l + \sum_{i=l}^{L-1} \mathcal{F}(\mathbf{x}_i, \mathcal{W}_i),$$

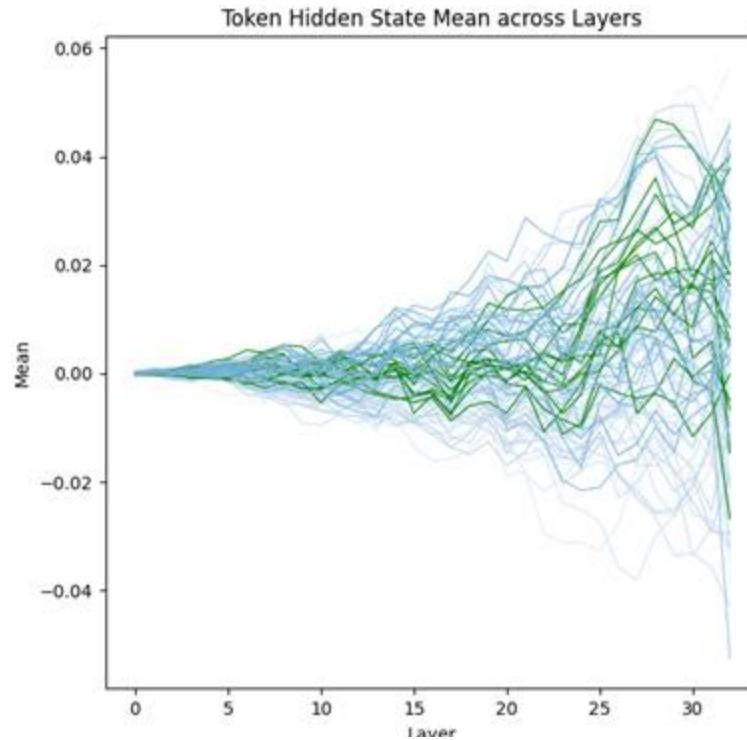
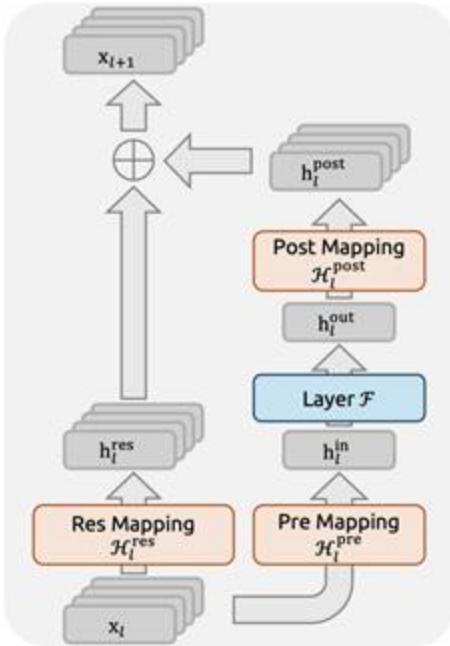
$$\mathbf{x}_{l+1} = \underline{\mathcal{H}_l^{\text{res}} \mathbf{x}_l} + \underline{\mathcal{H}_l^{\text{post}} \mathcal{F}(\mathcal{H}_l^{\text{pre}} \mathbf{x}_l, \mathcal{W}_l)}.$$

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

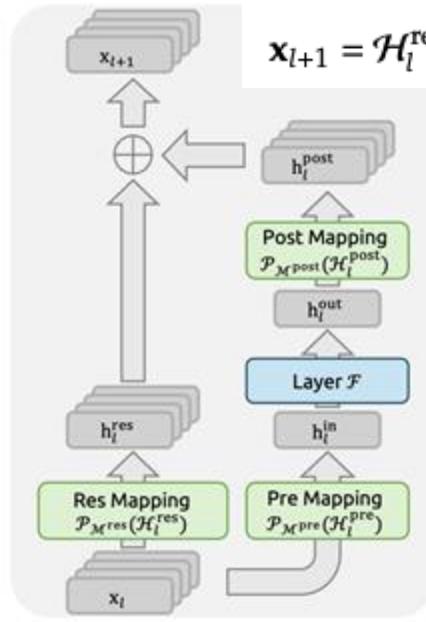
$$\mathcal{H}^{\text{res}} \in \mathbb{R}^{n \times n}$$

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

Information flow - instability



Information flow - project to a manifold



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

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

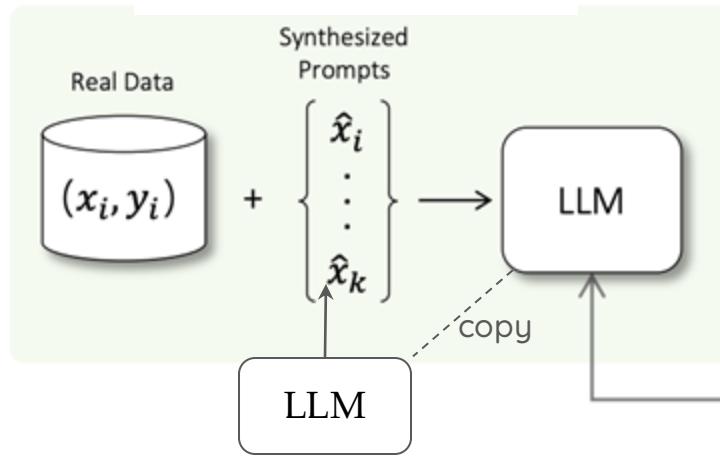
RECAP ✓: We have identified internal signals:

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

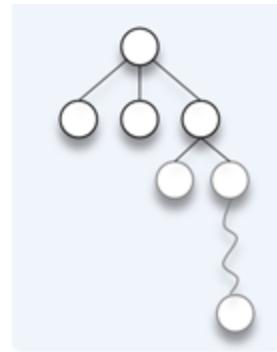
Now ✓: How can we build a self-improvement LLM using its internals?

Overview: self-improvement

(iii) Imagination



(ii) search



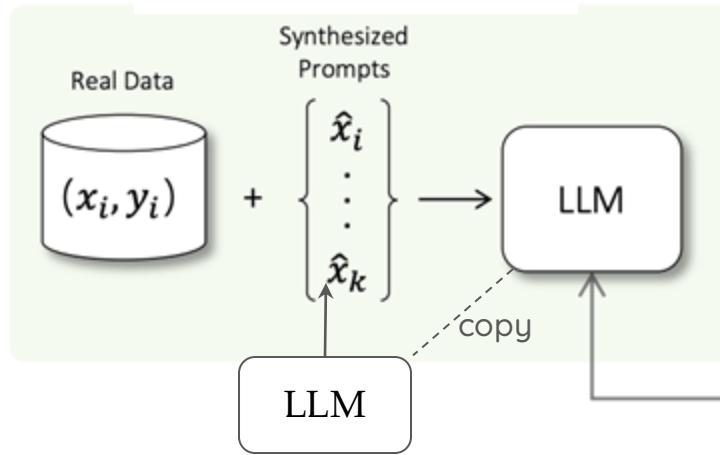
(i) self-evaluate



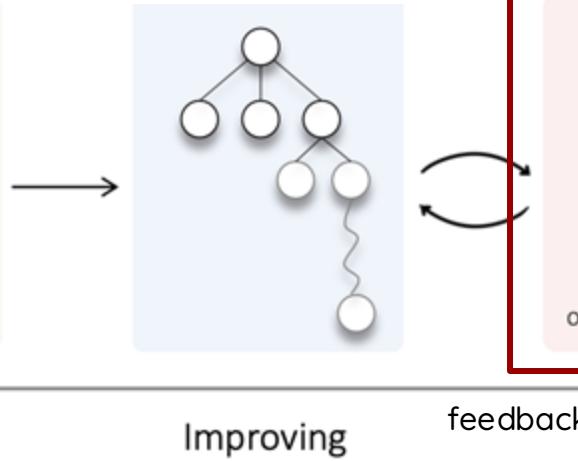
Improving

Self-improvement (I) - self evaluate

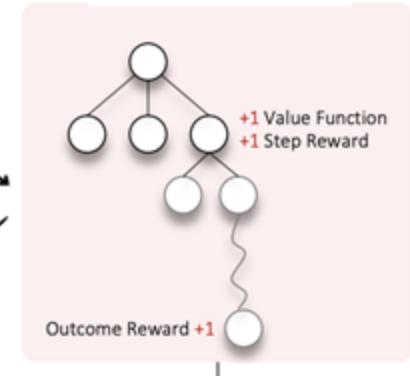
(iii) Imagination



(ii) search



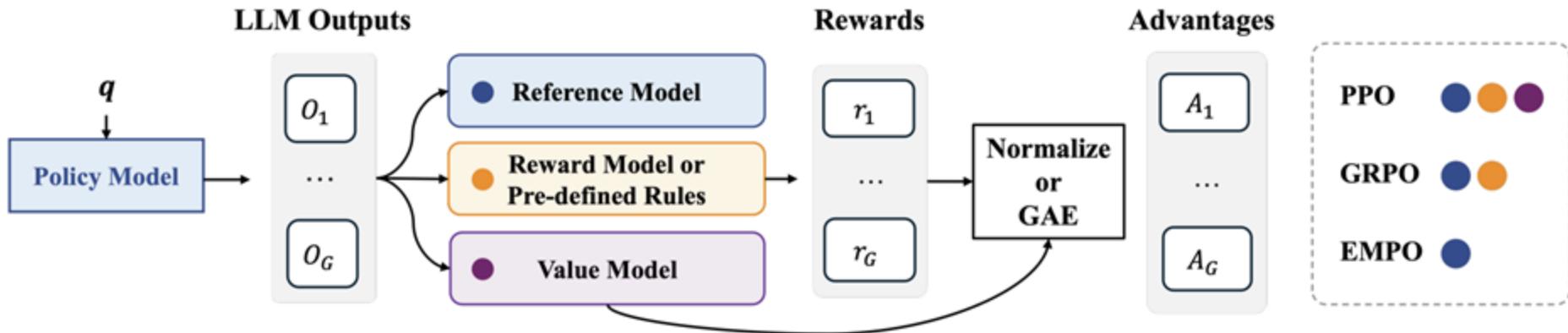
i) self-evaluate



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

Example: EMPO

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

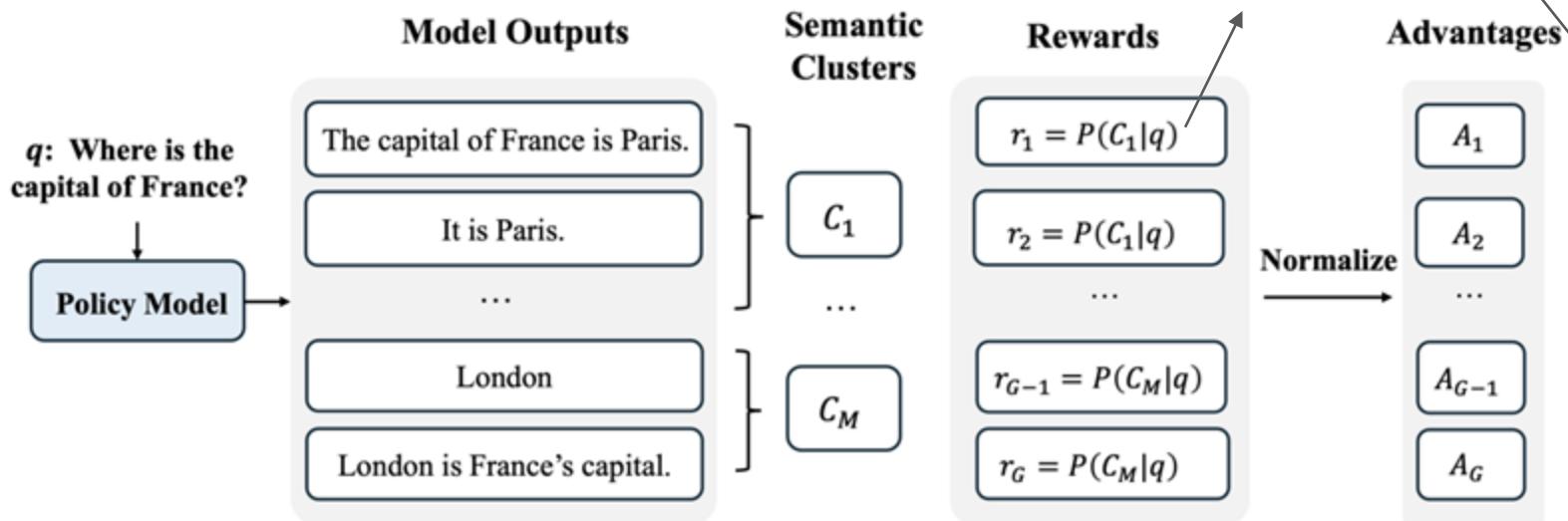


Example: EMPO

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

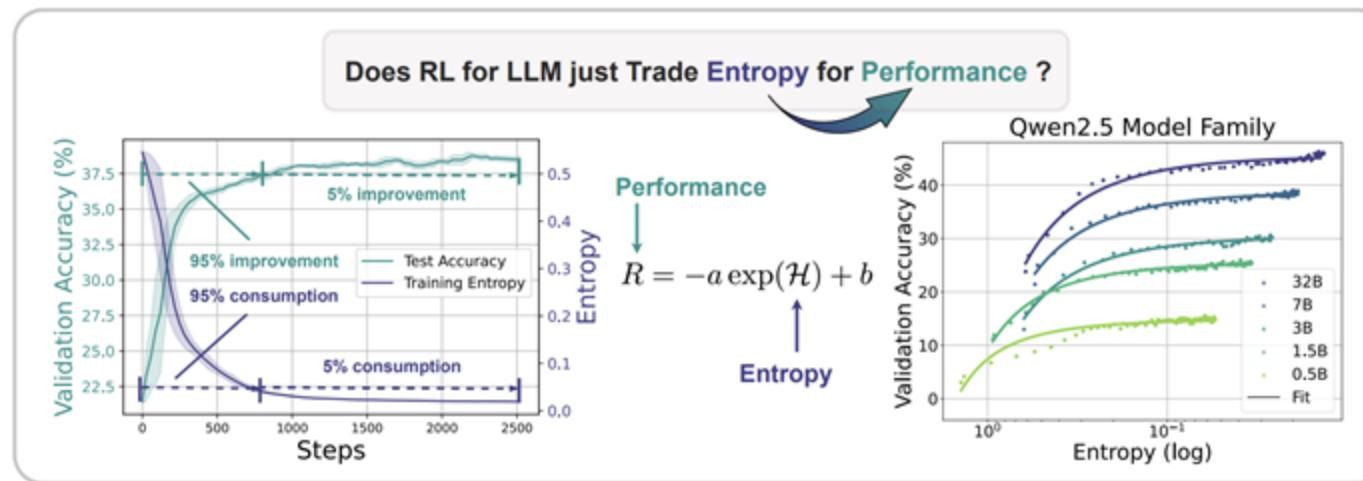
[\[Fu2025 UCSD\] Deep Think With Confidence](#)

[\[Agarwal 2025 UIUC\] The Unreasonable Effectiveness of Entropy Minimization in LLM Reasoning](#)

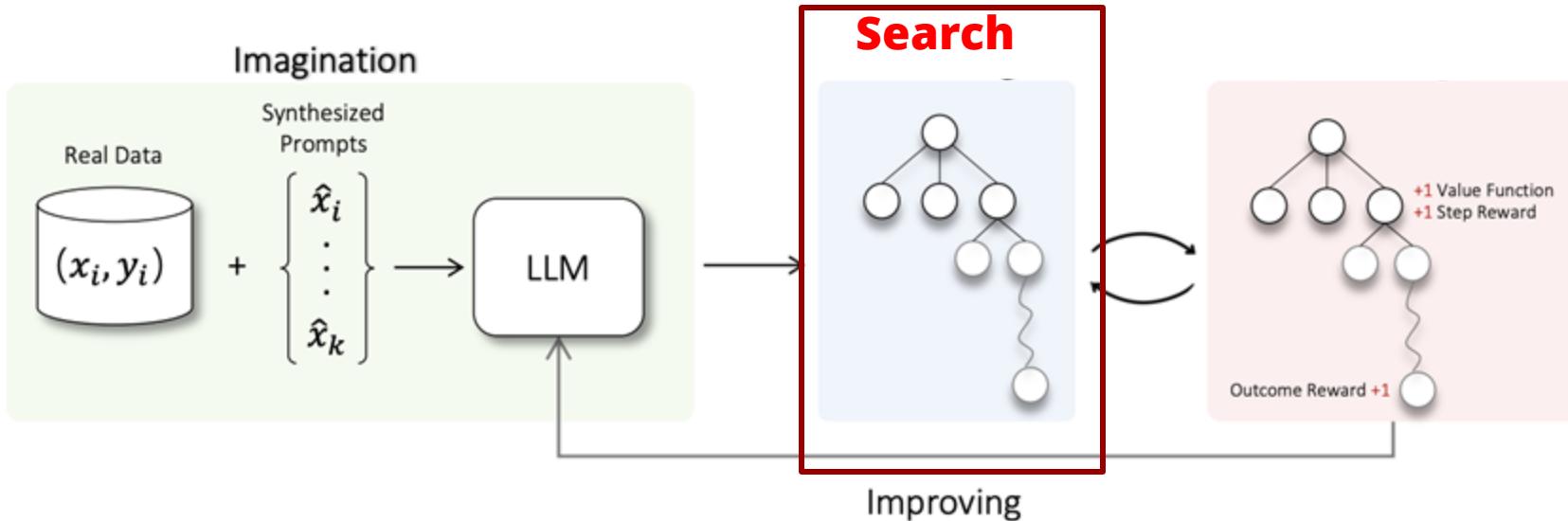
[\[Hu et al. ACL25. KCL\] Beyond Prompting: An Efficient Embedding Framework for Open-Domain Question Answering](#)

[\[Hu et al. AAAI25. KCL\] Beyond Perplexity: Let the Reader Select Retrieval Summaries via Spectrum Projection Score](#)

[\[Cui 2025 Tsinghua\] The Entropy Mechanism of Reinforcement Learning for Reasoning Language Models](#)



Self-improvement (I) - search



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

Example: search in the latent space

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

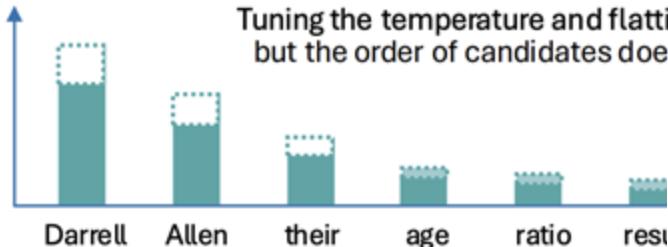
Qinglin Zhu ^{1*} **Runcong Zhao** ^{1*} **Hanqi Yan** ¹ **Yulan He** ^{1,2} **Yudong Chen** ³ **Lin Gui** ¹

[\[Zhu et al. ICML25 Spotlight\] 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

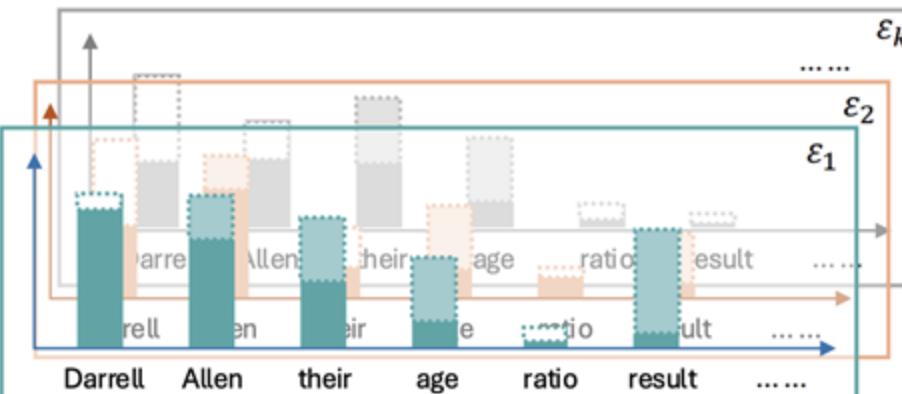


Bayesian optimization
after dimensionality reduction

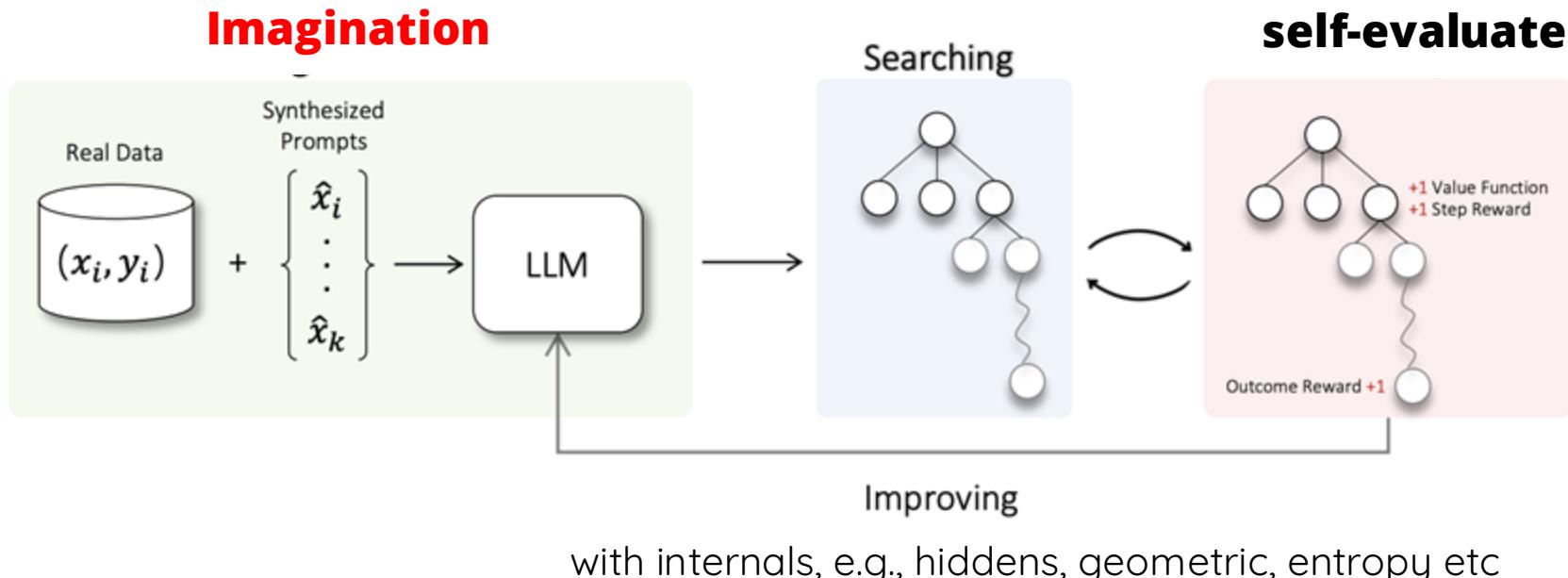
Add different gaussian embeddings,
But control with self-consistency reward

Soft Reasoning

Adjusting the distribution by injecting different controllable
Gaussian embedding

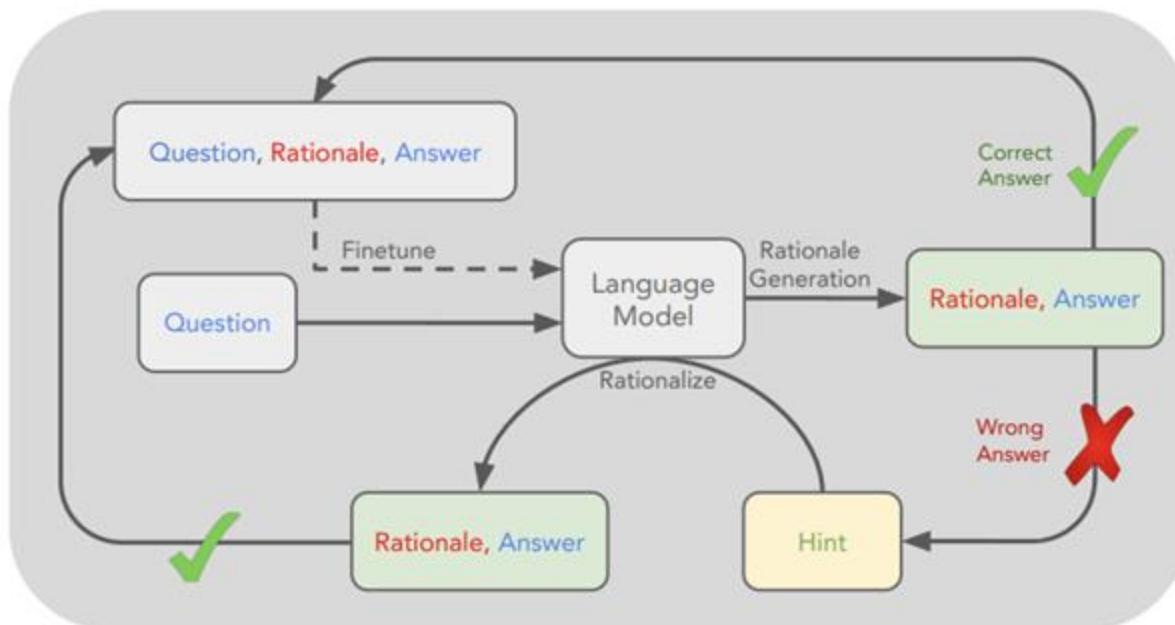


Self-improvement (III) - imagination

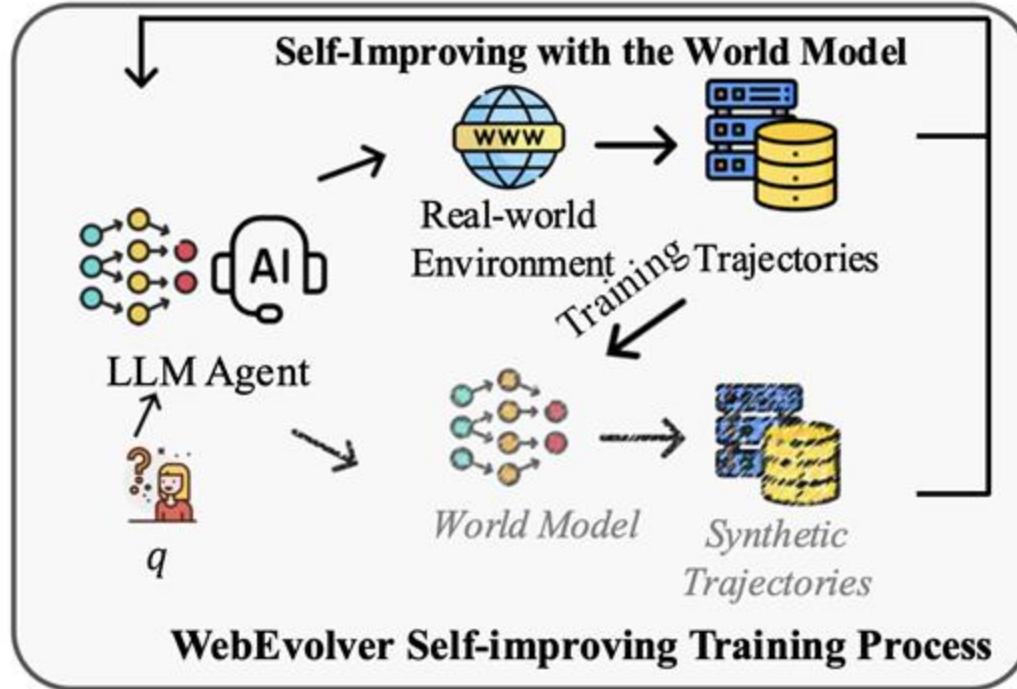


Example I: self-taught

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:

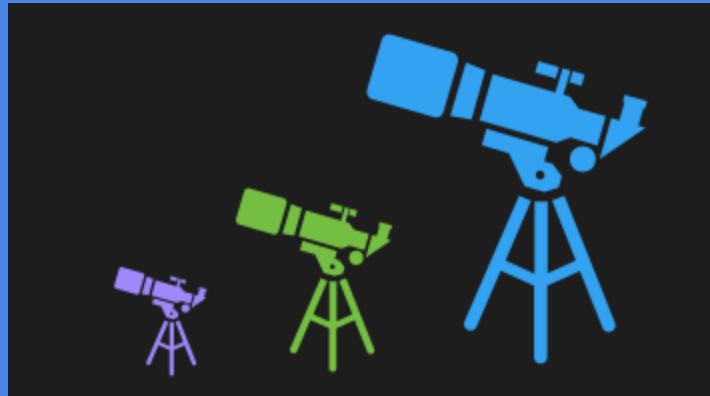
Internal Signals ✓: What the model internals tell us?

- Reasoning correctness
- Reasoning efforts
- Input sensitivity
- Information flows

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

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

Conclusion and Open Questions



Challenges: principles

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

Challenges: latent CoTs

- Latent CoT may overfit to task-specific reasoning patterns during training.
- Internal reasoning may not align with verbalized explanations.
- Recurrent refinement paths are difficult to control or constrain.
- 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 monosemanticity? Revisit Monosemanticity from a Feature Decorrelation Perspective](#)
- Incorporate the priors in time

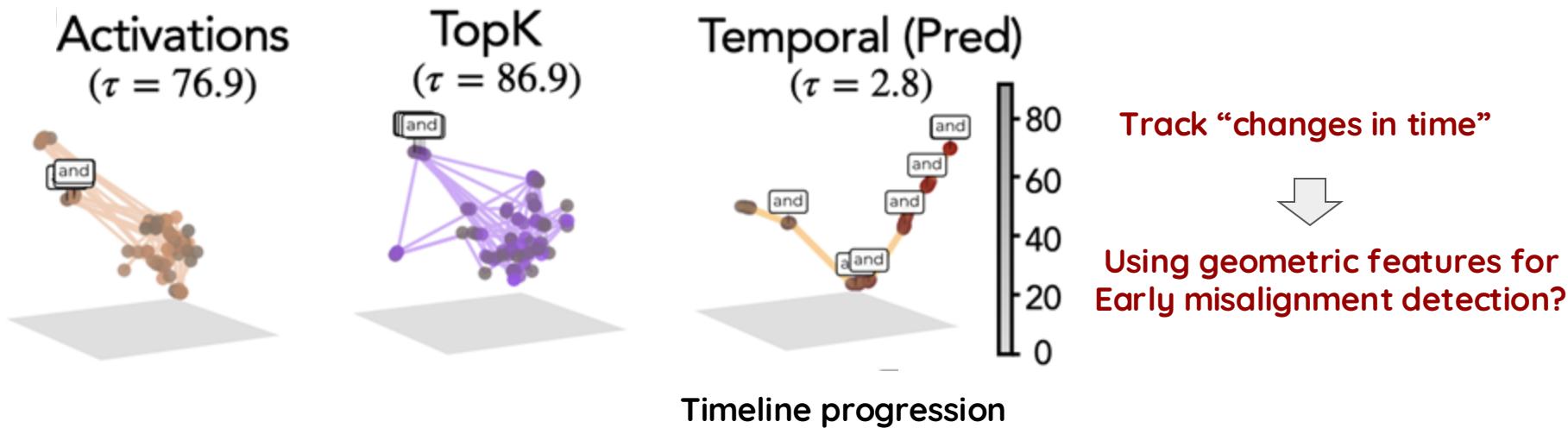
[\[Lubana et al, 2025\] Priors in time: Missing inductive biases for language model interpretability](#)
[\[Song et al. 25\] LLM Interpretability with Identifiable Temporal-Instantaneous Representation](#)



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.



Challenges: LLM as world model

1. Early prediction ?
1. Faithful to the true reasoning process ?
1. Probabilistic prediction, incorporate uncertainty .

Internal predicts the future earlier (I)

Write a rhyming poem.

A rhyming couplet: ↪

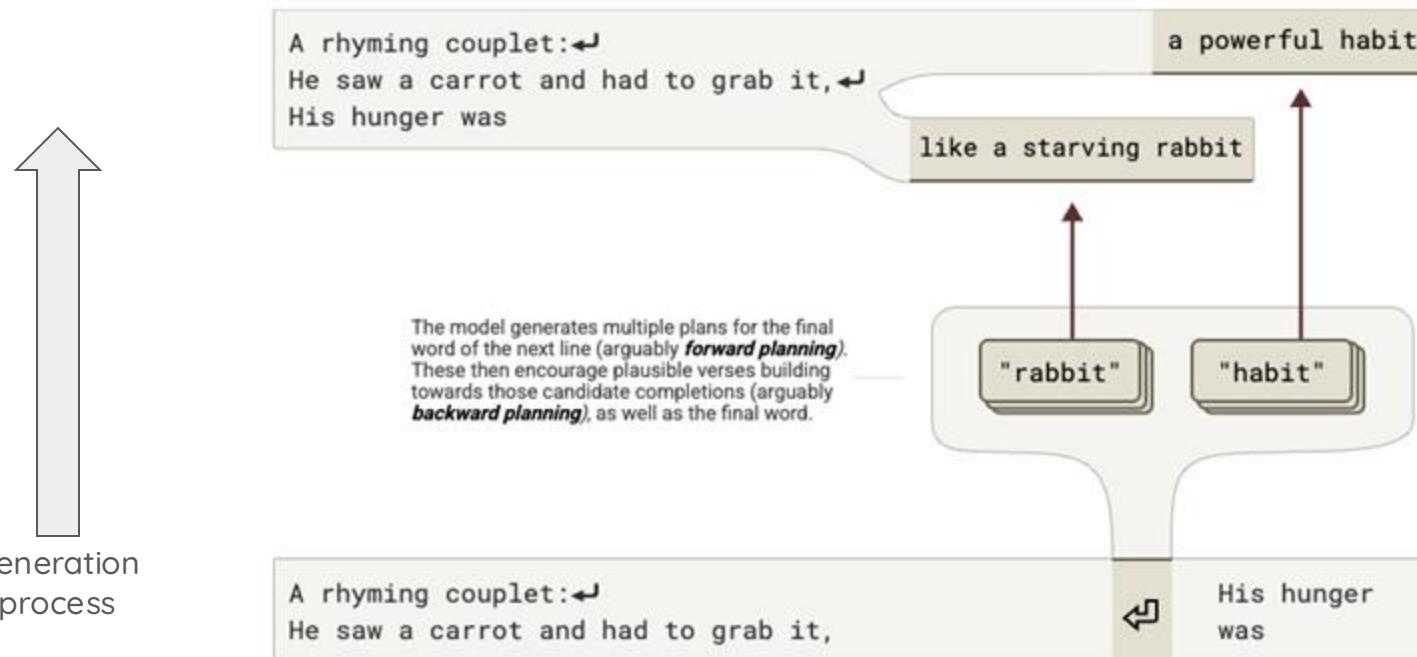
He saw a carrot and had to grab it, ↪ ? plan ↪ rabbit

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 are 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^{1,3}

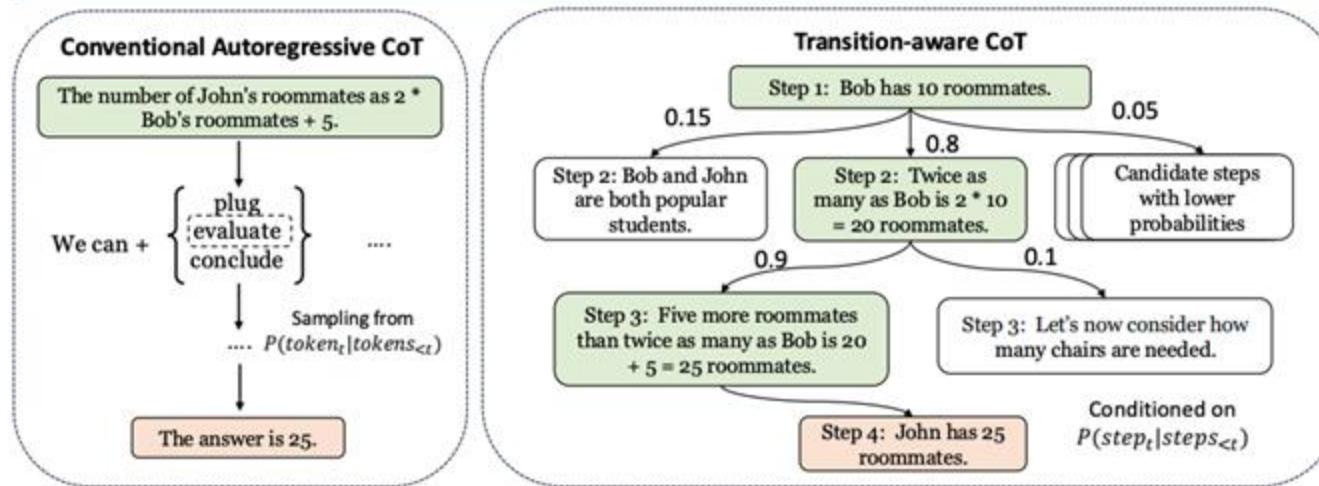
¹NYU Alignment Research Group, ²Cohere, ³Anthropic

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?



hanqi.1.yan@kcl.ac.uk



Guangyichen1994@gmail.com



schwarzjn@gmail.com



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

- Slides
- Full list of references