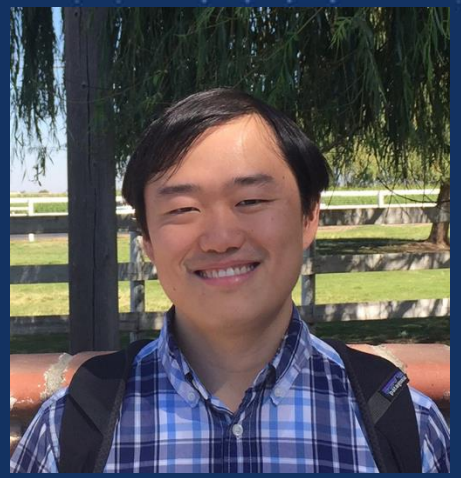
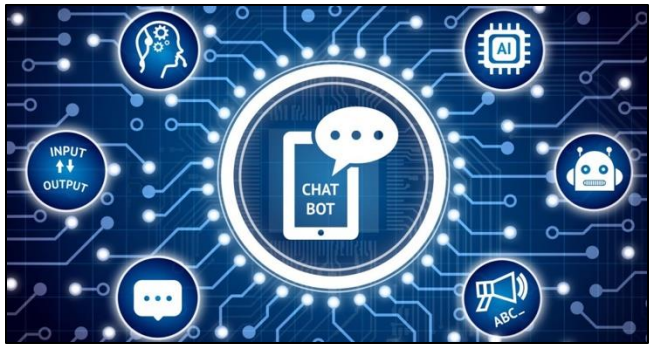
Emergence of Various Structures via the Lens of Transformer Training Dynamics

Yuandong Tian
Research Scientist Director

Meta AI

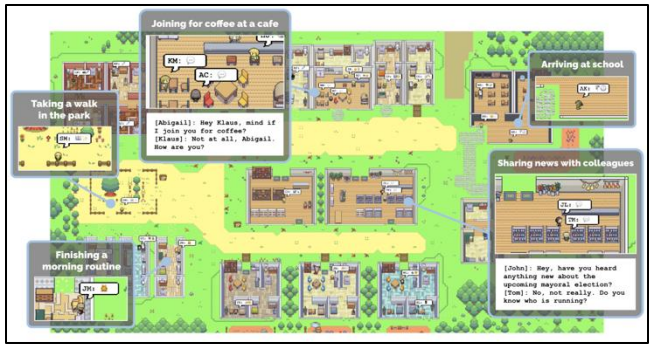


Large Language Models (LLMs)

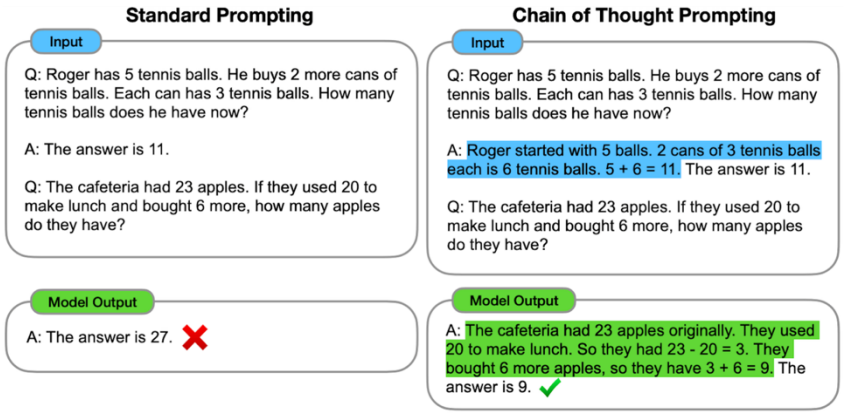


Conversational AI

Content Generation



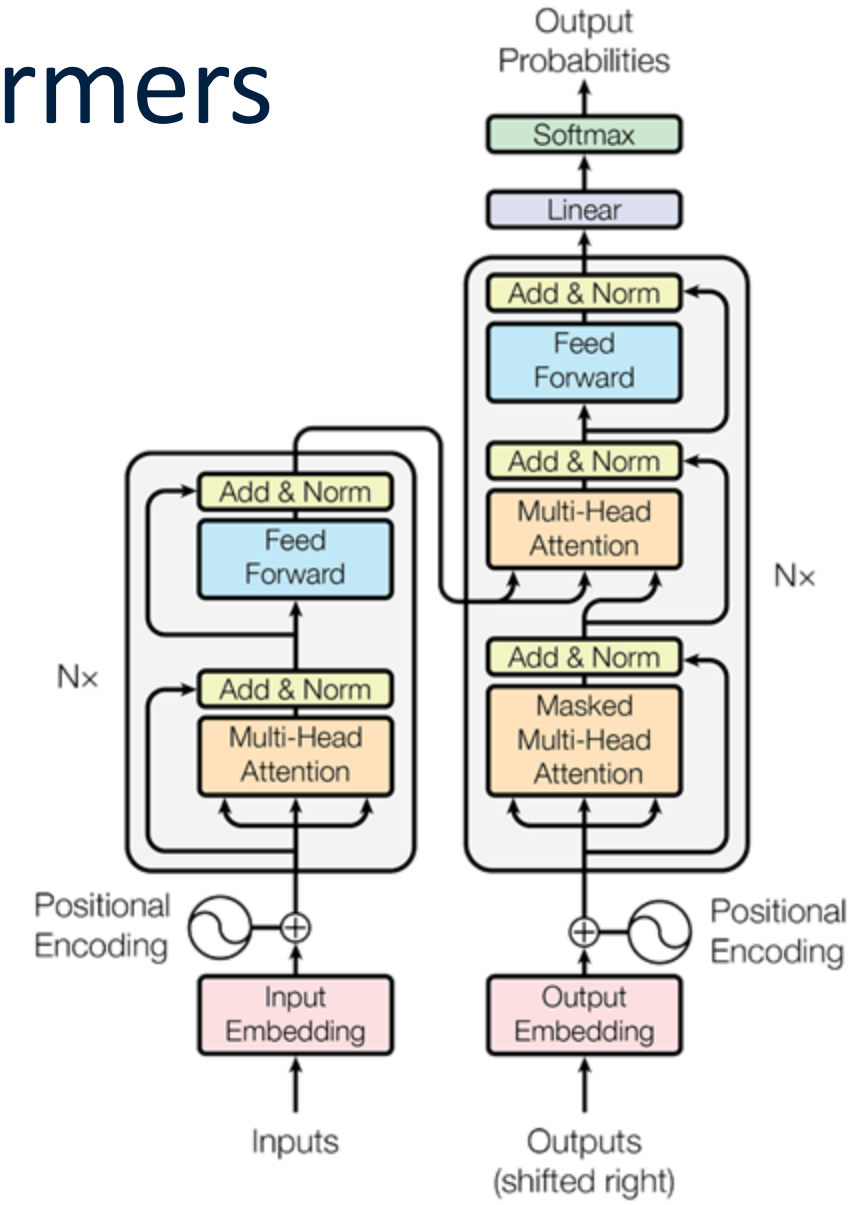
AI Agents

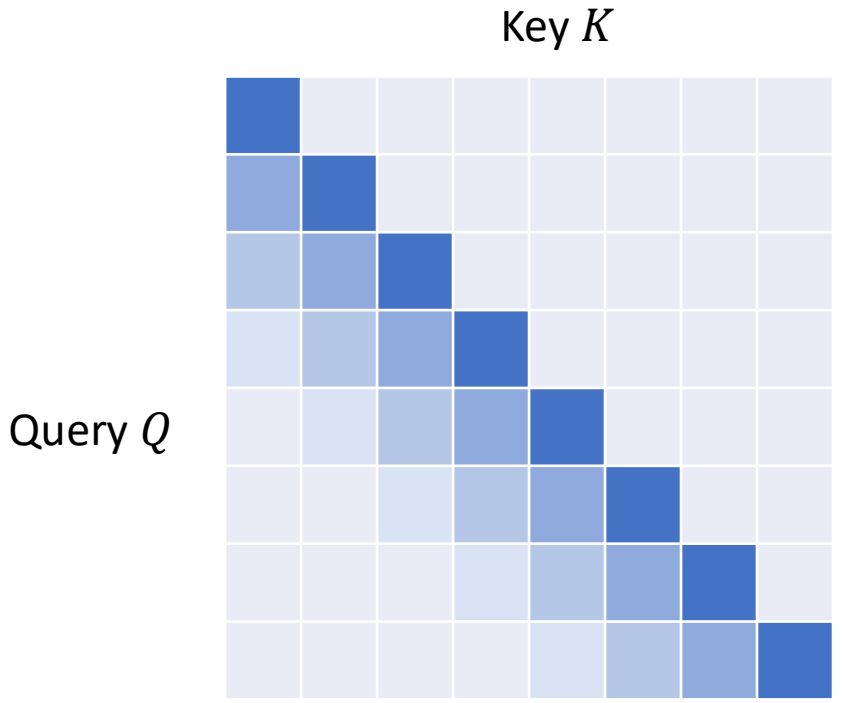


Reasoning

Planning

Transformers





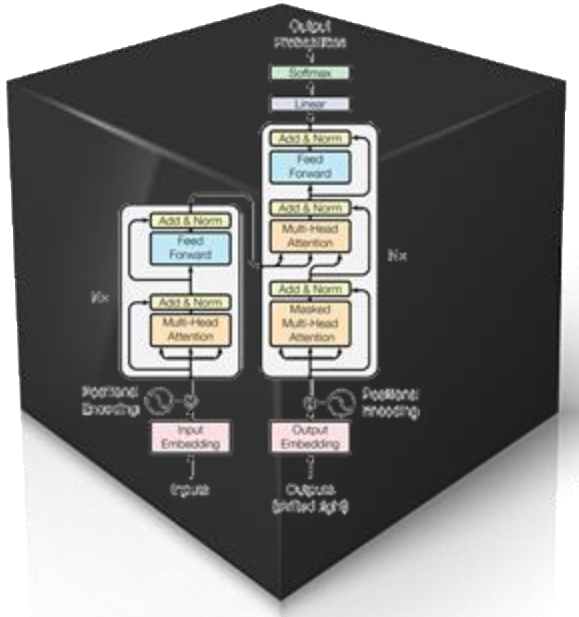
Attention mechanism

[A. Vaswani et al, Attention is all you need, NeurIPS'17]

How does Transformer work?

Input

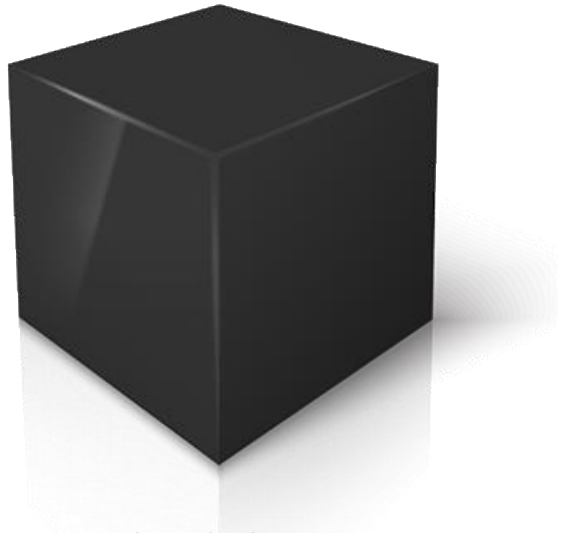
This is an apple



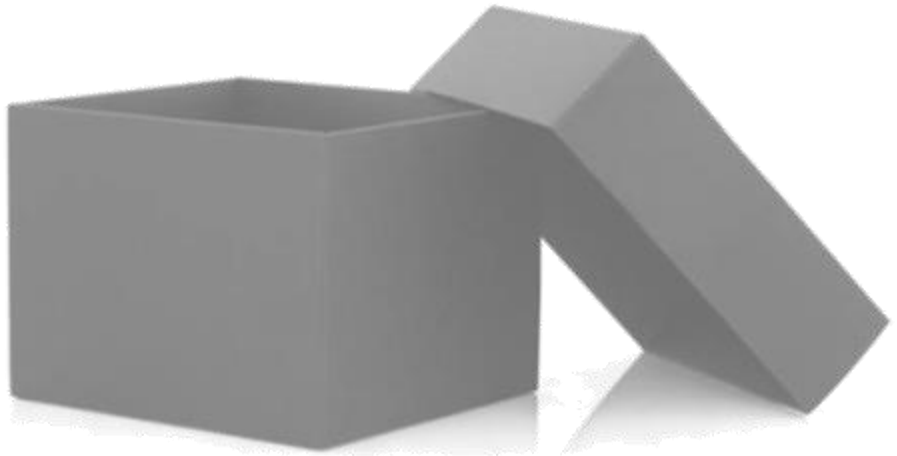
"Some Nonlinear Transformation"

Output

Black-box versus White-box



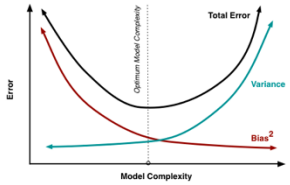
Black box



White box

What routes should we take?

Generalization



Architecture ✗

training dynamics ✗

Expressibility

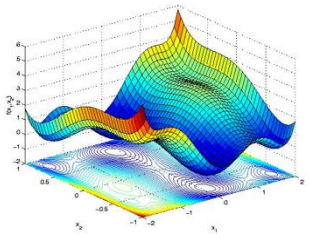
\+ −

− +

Architecture ✓

training dynamics ✗

Optimization



Architecture ✗

training dynamics ✓

How about

Architecture ✓

training dynamics ✓

Start From the First Principle

- Training follows Gradient and its variants (SGD, Adams, etc)

$$\dot{\boldsymbol{w}} := \frac{\mathrm{d}\boldsymbol{w}}{\mathrm{d}t} = -\nabla_{\boldsymbol{w}} J(\boldsymbol{w})$$

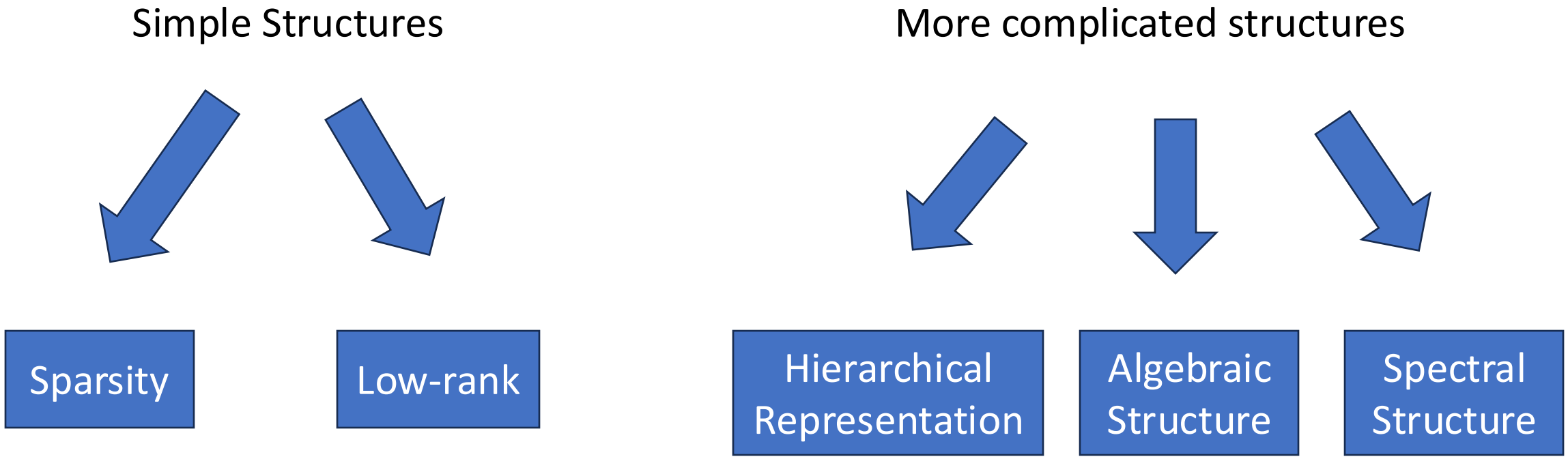
- **First principle** → Understand the behavior of the neural networks by checking the gradient **dynamics** induced by the neural **architectures**.

- Sounds complicated.. Is that possible? **Yes**

Architecture ✓

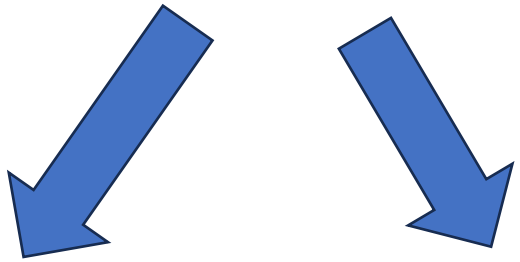
training dynamics ✓

What Gradient Descent gives us?



What Gradient Descent gives us?

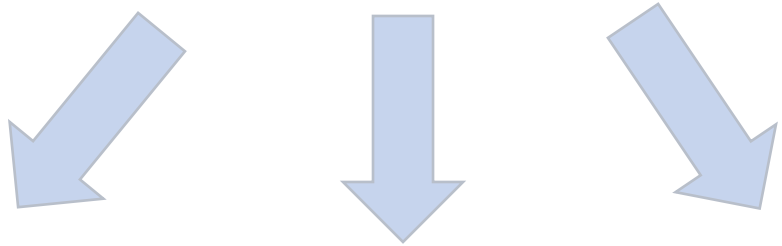
Simple Structures



Sparsity

Low-rank

More complicated structures



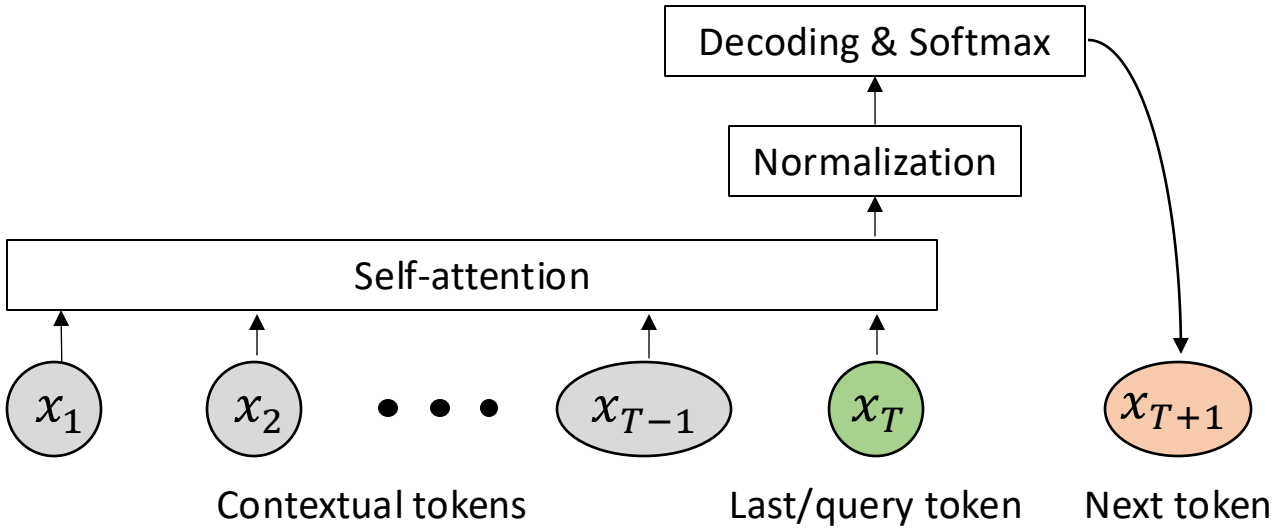
Hierarchical Representation

Algebraic Structure

Spectral Structure

Understanding Attention in 1-layer Setting

$U = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots \boldsymbol{u}_M]^T$: token embedding matrix



Self-attention

$$\hat{\boldsymbol{u}}_T = \sum_{t=1}^{T-1} b_{tT} \boldsymbol{u}_{x_t} = U^T X^T \boldsymbol{b}_T$$

$$b_{tT} := \frac{\exp(\boldsymbol{u}_{x_T}^\top W_Q W_K^\top \boldsymbol{u}_{x_t} / \sqrt{d})}{\sum_{t=1}^{T-1} \exp(\boldsymbol{u}_{x_T}^\top W_Q W_K^\top \boldsymbol{u}_{x_t} / \sqrt{d})}$$

Normalized version $\tilde{\boldsymbol{u}}_T = U^T \mathrm{LN}(X^T \boldsymbol{b}_T)$

Objective:

$$\max_{W_K, W_Q, W_V, U} J = \mathbb{E}_D \left[\boldsymbol{u}_{x_{T+1}}^T W_V \tilde{\boldsymbol{u}}_T - \log \sum_l \exp(\boldsymbol{u}_l^T W_V \tilde{\boldsymbol{u}}_T) \right]$$

[*Y. Tian et al*, Scan and Snap: Understanding Training Dynamics and Token Composition in 1-layer Transformer, *NeurIPS'23*]

Reparameterization

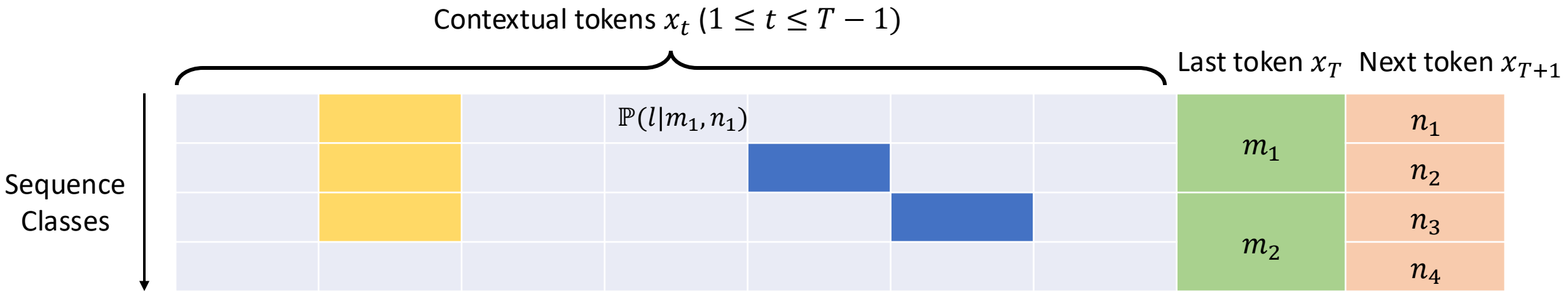
- Parameters W_K, W_Q, W_V, U makes the dynamics complicated.

- Reparameterize the problem with independent variable Y and Z
  - $Y = UW_V^T U^T$
  - $Z = UW_Q W_K^T U^T$ (pairwise logits of self-attention matrix)

- Then the dynamics becomes easier to analyze

Data Distribution

$x_t \in [M]$ for $1 \le t \le T$
$x_{T+1} \in [K]$
$K \ll M$



Distinct tokens: There exists unique n so that $\mathbb{P}(l|n) > 0$

Common tokens: There exists multiple n so that $\mathbb{P}(l|n) > 0$

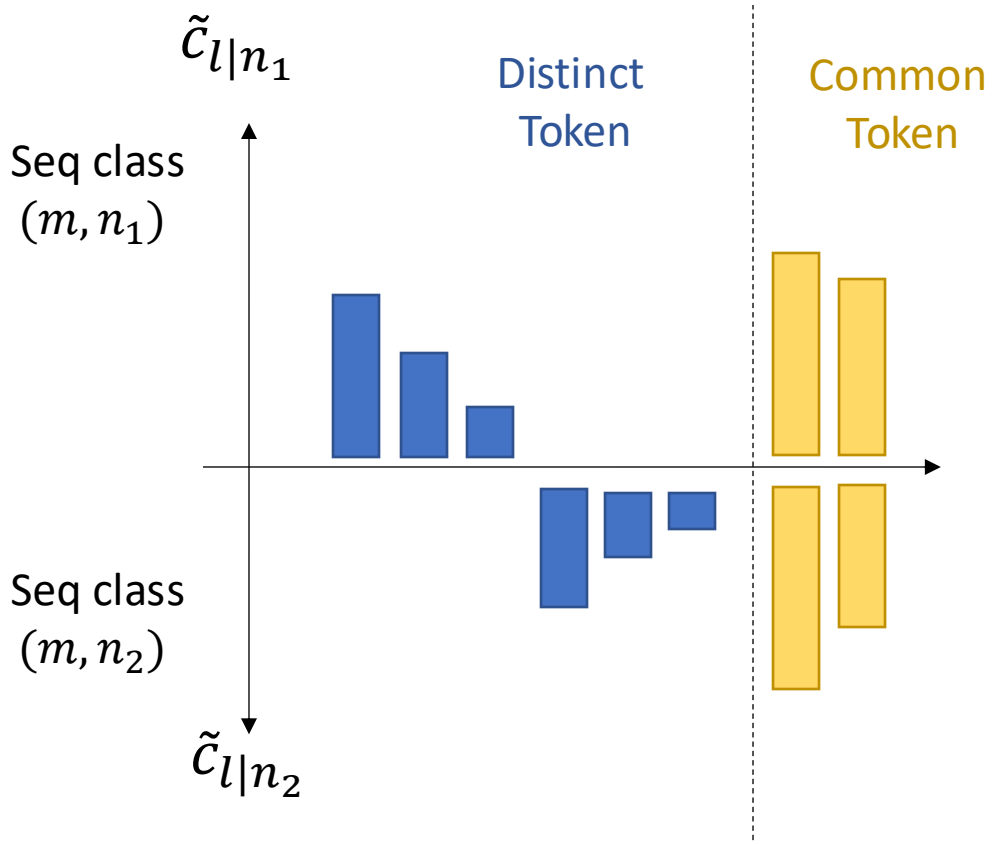
$\mathbb{P}(l|m,n) = \mathbb{P}(l|n)$ is the conditional probability of token l given last token $x_T = m$ and $x_{T+1} = n$

Assumption: $m = \psi(n)$, i.e., no next token shared among different last tokens

Question: Given the data distribution, how does the self-attention layer behave?

Overall Picture of the Training Dynamics

At initialization



Co-occurrence probability

$$\tilde{c}_{l|n_1} := \mathbb{P}(l|m, n_1) \exp(z_{ml})$$

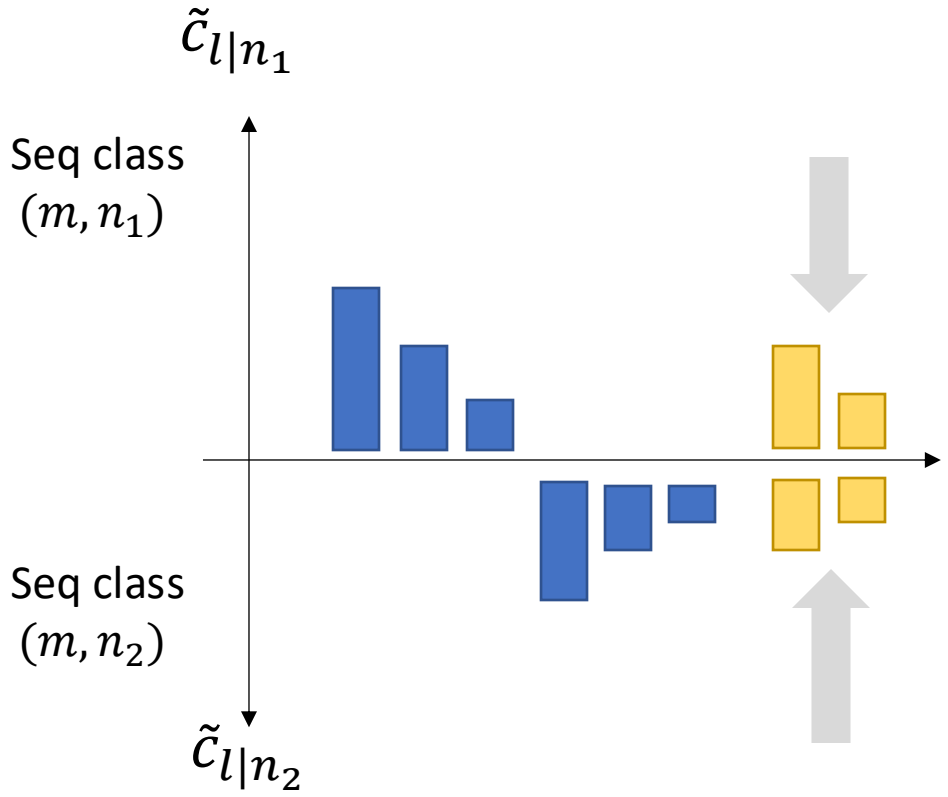
Initial condition: $z_{ml}(0) = 0$

$$Z = \begin{bmatrix} & & & \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ & & & \\ & & & \end{bmatrix} \mathbf{z}_m$$

$\mathbf{z}_m$: All logits of the contextual tokens when attending to last token $x_T = m$

Overall Picture of the Training Dynamics

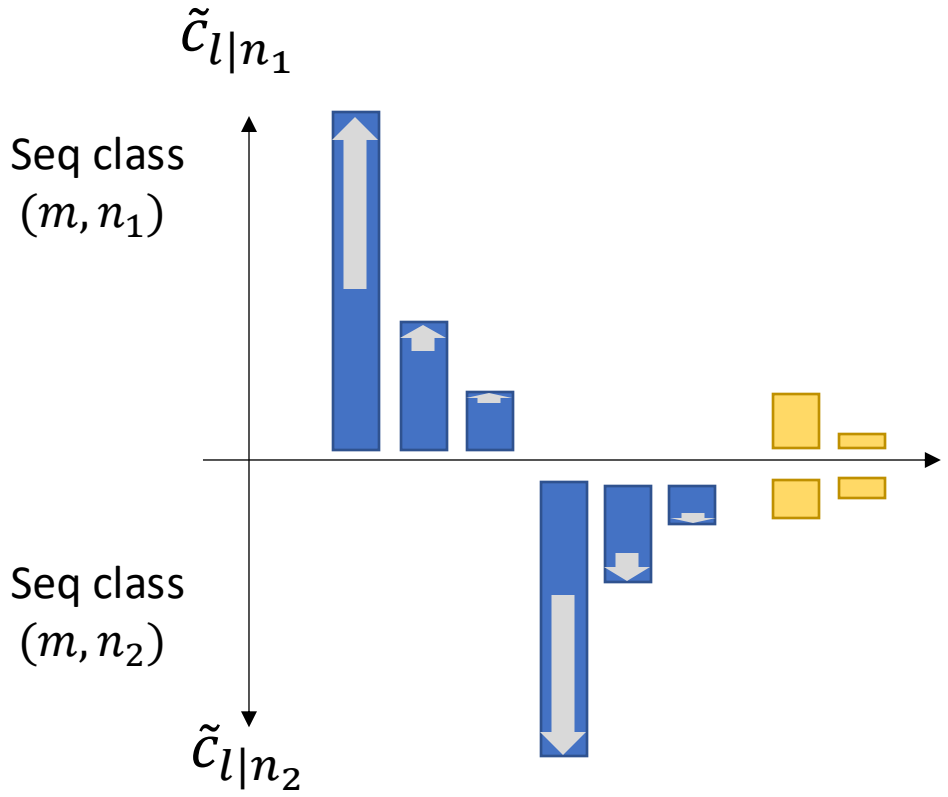
Common Token Suppression



(a) $\dot{z_{ml}} < 0$, for common token l

Overall Picture of the Training Dynamics

Winners-emergence



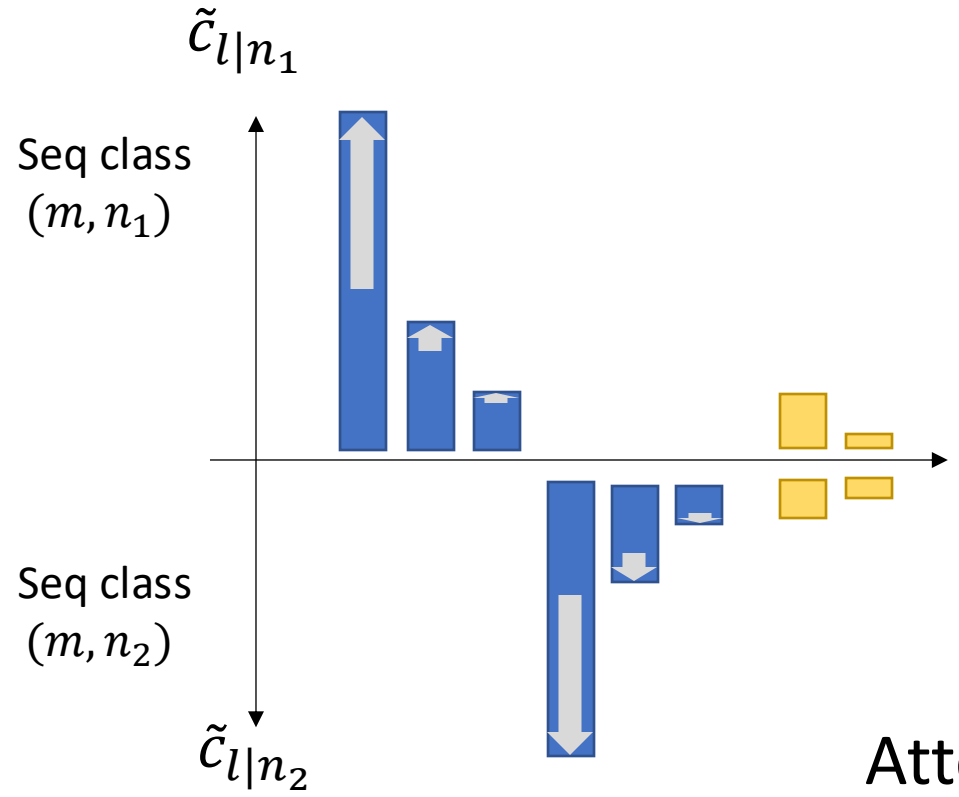
(a) $\dot{z_{ml}} < 0$, for common token l

(b) $\dot{z_{ml}} > 0$, for distinct token l

Learnable TF-IDF (Term Frequency, Inverse Document Frequency)

Overall Picture of the Training Dynamics

Winners-emergence



(a) $\dot{z_{ml}} < 0$, for common token l

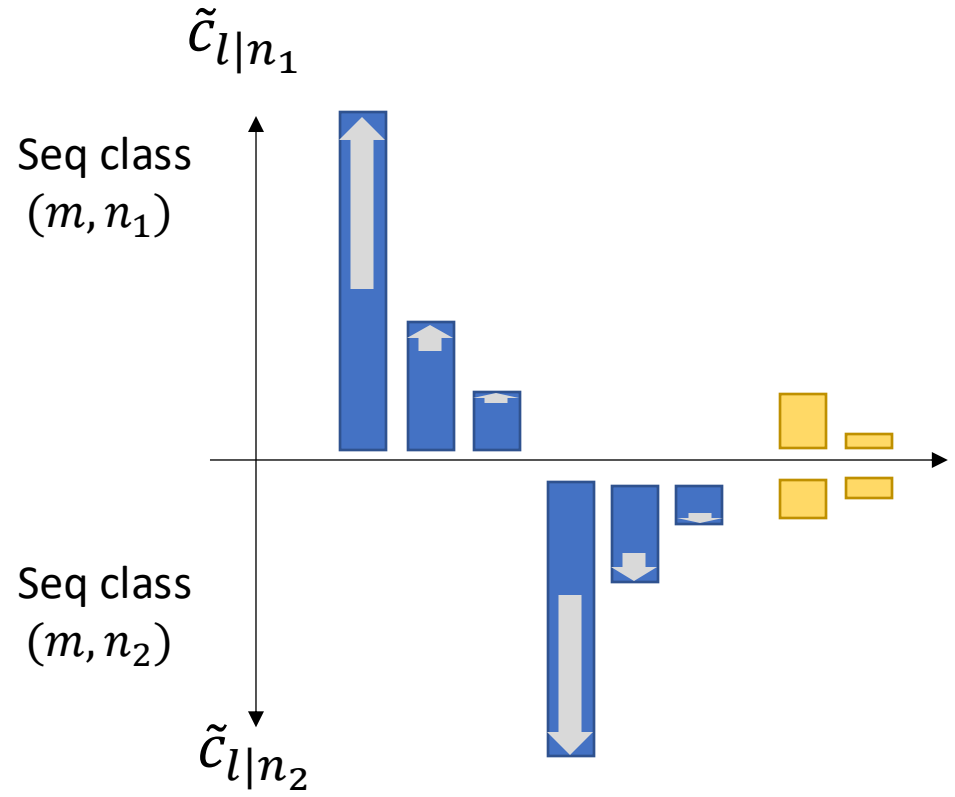
(b) $\dot{z_{ml}} > 0$, for distinct token l

(c) $z_{ml}(t)$ grows faster with larger $\mathbb{P}(l|m, n)$

Attention looks for **discriminative** tokens that **frequently co-occur** with the query.

Overall Picture of the Training Dynamics

Winners-emergence



(c) $z_{ml}(t)$ grows faster with larger $\mathbb{P}(l|m,n)$

Theorem 3 Relative gain $r_{l/l'|n}(t) := \frac{\tilde{c}^2_{l|n}(t)}{\tilde{c}^2_{l'|n}(t)} - 1$ has a close form:

$$r_{l/l'|n}(t) = r_{l/l'|n}(0)\chi_l(t)$$

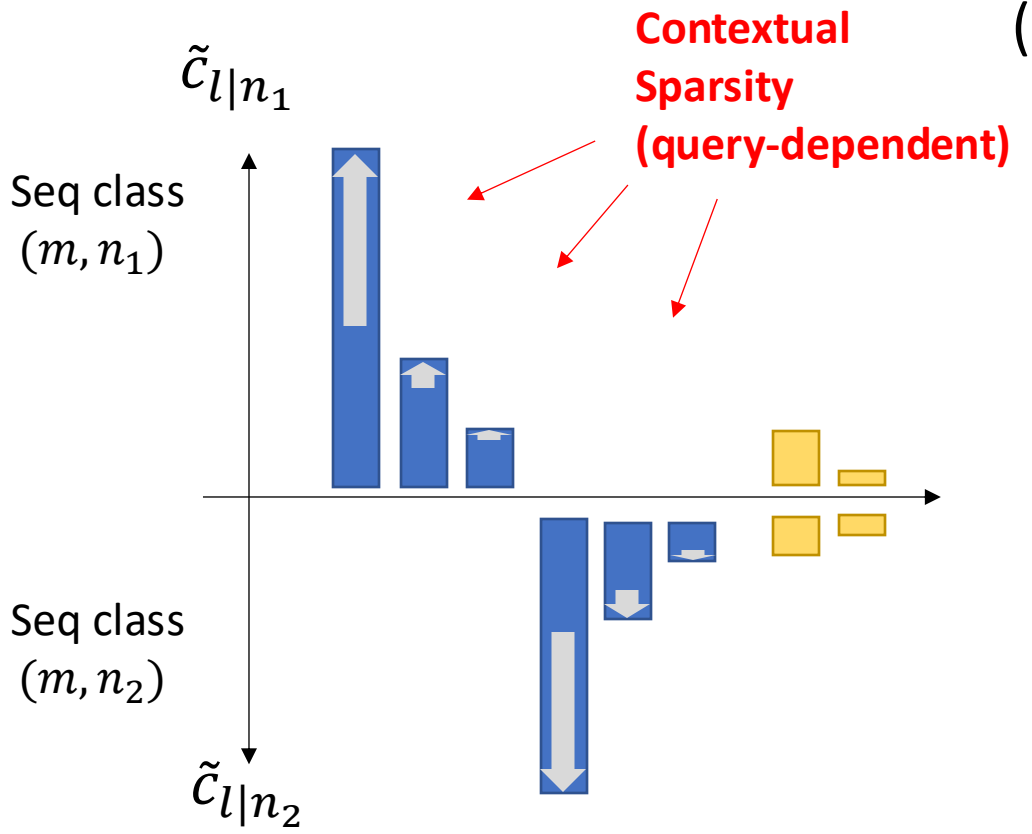
If l_0 is the dominant token: $r_{l_0/l|n}(0) > 0$ for all $l \neq l_0$ then

$$e^{2f^2_{nl_0}(0)B_n(t)} \leq \chi_{l_0}(t) \leq e^{2B_n(t)}$$

where $B_n(t) \geq 0$ monotonously increases, $B_n(0) = 0$

Overall Picture of the Training Dynamics

Winners-emergence



(c) $z_{ml}(t)$ grows faster with larger $\mathbb{P}(l|m,n)$

Theorem 3 Relative gain $r_{l/l'|n}(t) := \frac{\tilde{c}^2_{l|n}(t)}{\tilde{c}^2_{l'|n}(t)} - 1$ has a close form:

$$r_{l/l'|n}(t) = r_{l/l'|n}(0)\chi_l(t)$$

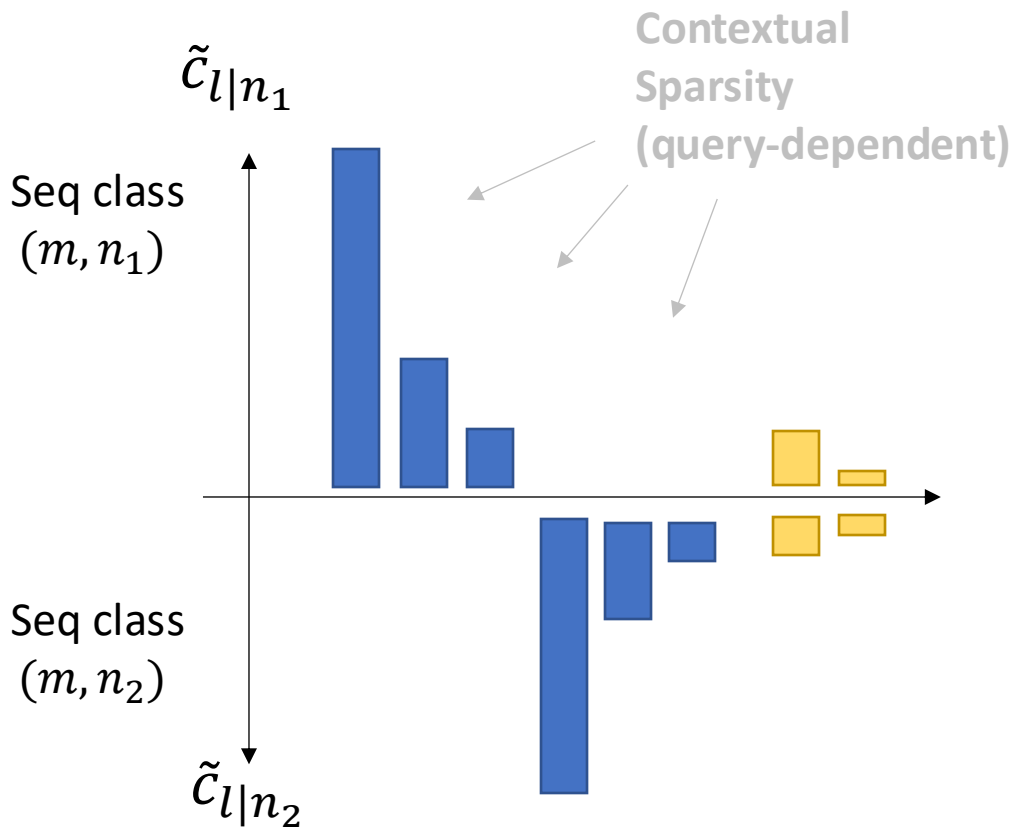
If l_0 is the dominant token: $r_{l_0/l|n}(0) > 0$ for all $l \neq l_0$ then

$$e^{2f^2_{nl_0}(0)B_n(t)} \leq \chi_{l_0}(t) \leq e^{2B_n(t)}$$

where $B_n(t) \geq 0$ monotonously increases, $B_n(0) = 0$

Overall Picture of the Training Dynamics

Attention frozen



Theorem 4 When $t \to +\infty$,

$$B_n(t) \sim \ln\left(C_0 + 2K\frac{\eta_z}{\eta_Y}\ln^2\left(\frac{M\eta_Y t}{K}\right)\right)$$

Attention **scanning**:

When training starts, $B_n(t) = O(\ln t)$

Attention **snapping**:

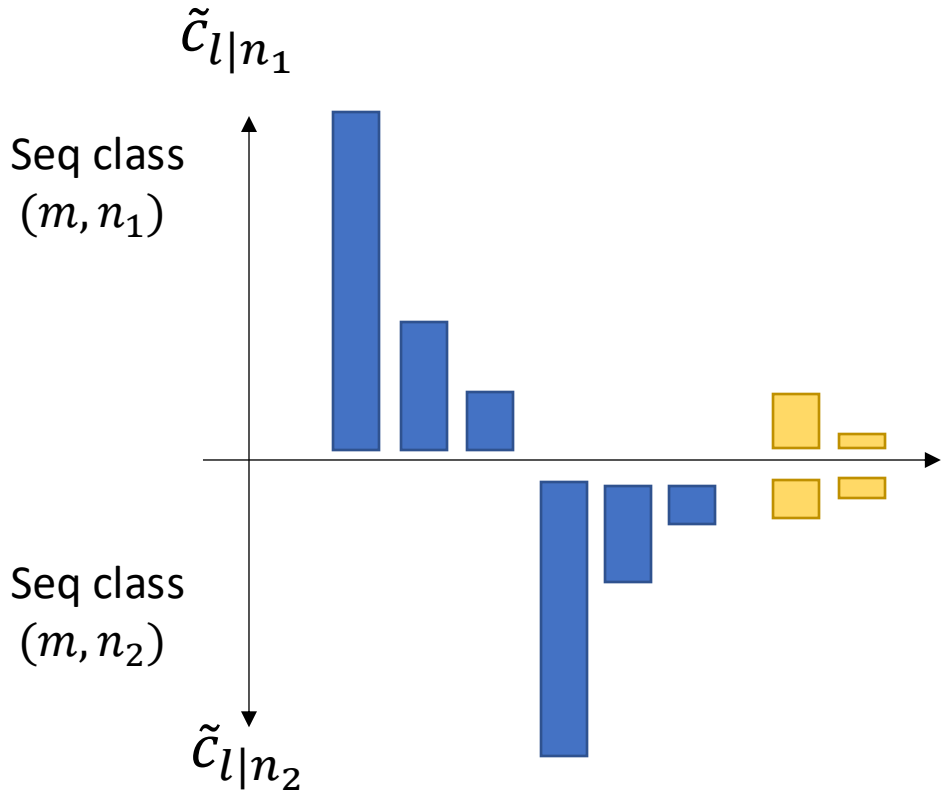
When $t \geq t_0 = O\left(\frac{2K \ln M}{\eta_Y}\right)$, $B_n(t) = O(\ln \ln t)$

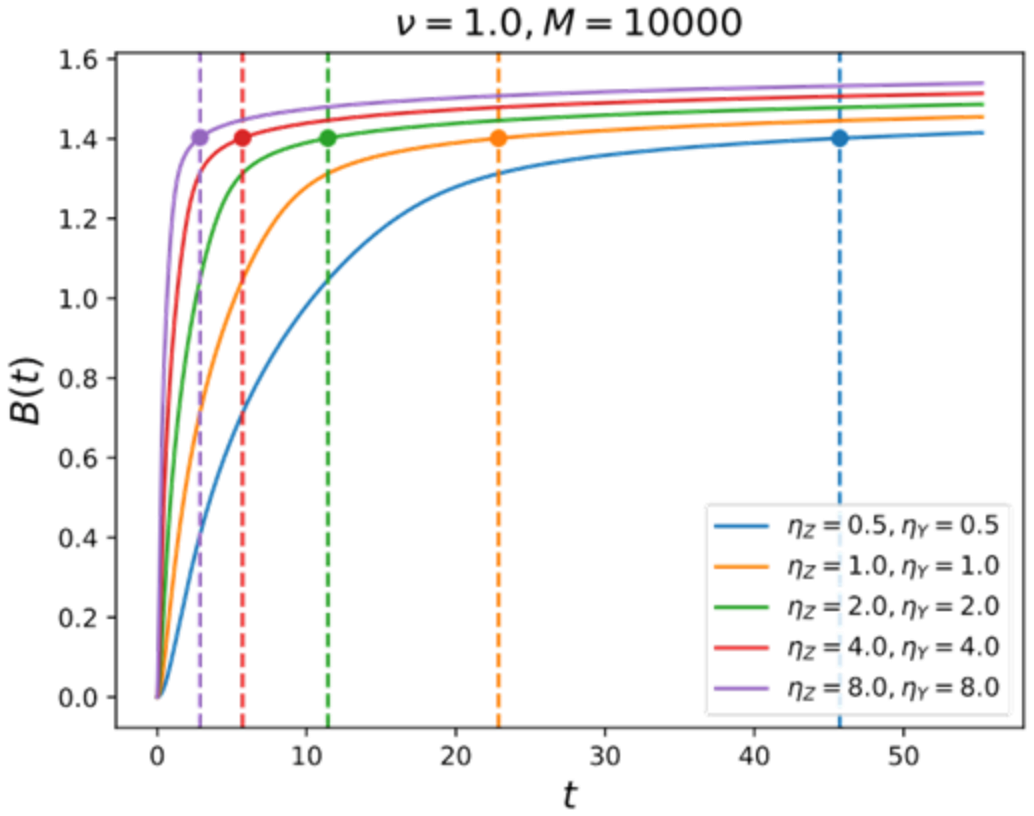
(1) η_z and η_Y are large, $B_n(t)$ is large and attention is sparse

(2) Fixing η_z, large η_Y leads to slightly small $B_n(t)$ and denser attention

Overall Picture of the Training Dynamics

Attention frozen

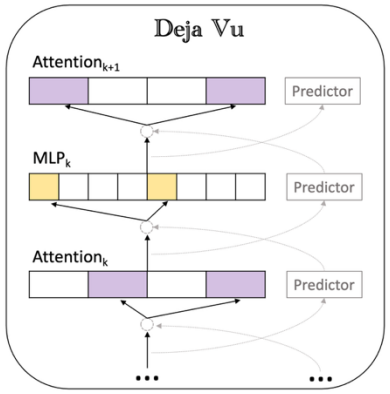




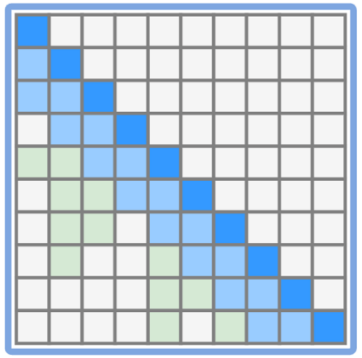
Larger learning rate η_z leads to faster phase transition

$$B_n(t) \sim \ln\left(C_0 + 2K\frac{\eta_z}{\eta_Y}\ln^2\left(\frac{M\eta_Y t}{K}\right)\right)$$

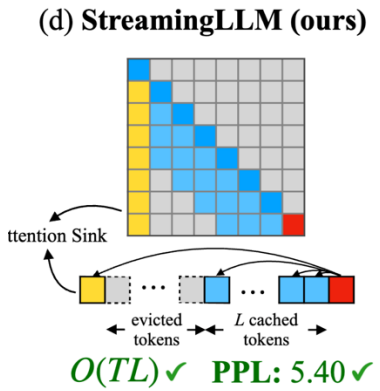
Further Study of Sparse Attention / Low Rank



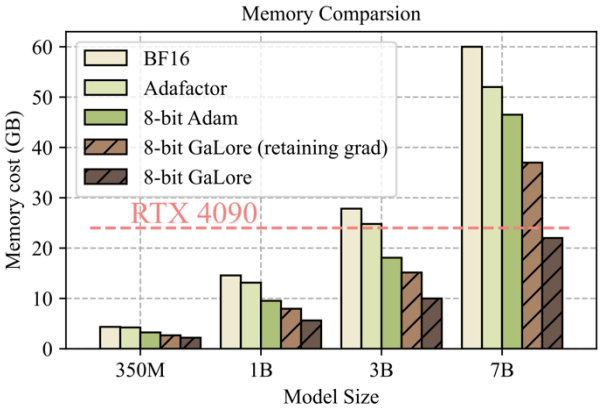
DejaVu [Z. Liu et al, ICML'23 (oral)]



H2O [Z. Zhang et al, NeurIPS'23]



Attention Sink [G. Xiao et al, ICLR'24]



GaLore [J. Zhao et al, , ICML'24 (Oral)]

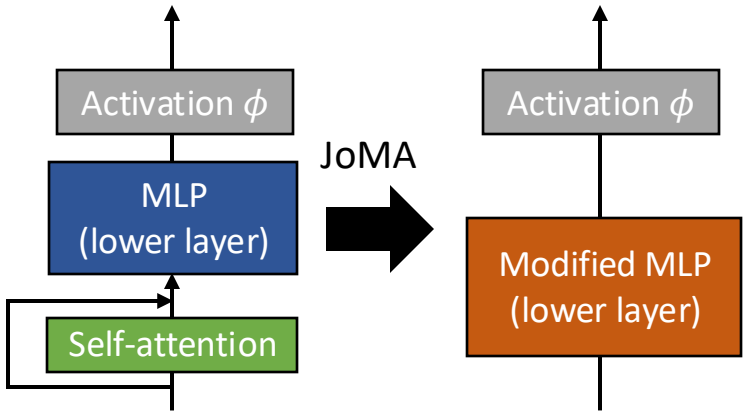
How to get rid of the assumptions?

- A few annoying assumptions in the analysis
  - No residual connections
  - No embedding vectors
  - The decoder needs to learn faster than the self-attention ($\eta_Y \gg \eta_Z$).
  - Single layer analysis

- How to get rid of them?

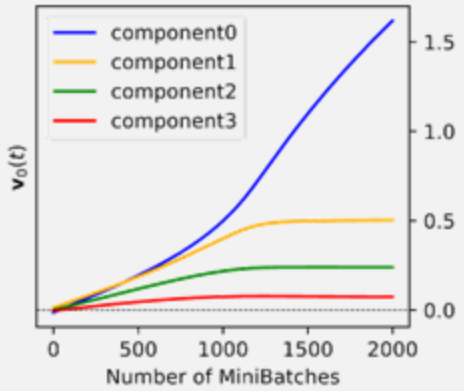
- Follow-up work: **JoMA** (Joint MLP/Attention dynamics)

Folding self-attention into MLP



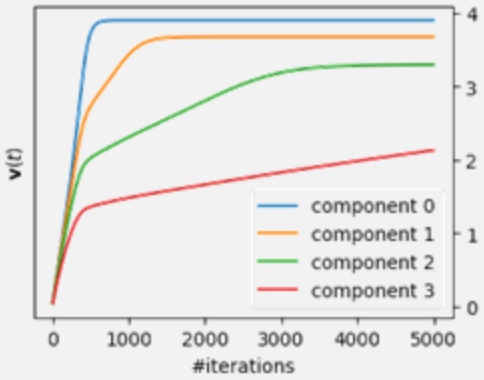
A Transformer block becomes a modified MLP
(due to nice property in gradient dynamics)

Linear case ($\phi = \mathrm{Id}, K = 1$)



Most salient feature takes all
(Attention becomes sparser)
(Consistent with Scan&Snap)

Nonlinear case (ϕ nonlinear, $K = 1$)



Most salient feature grows, and others catch up
(Attention becomes sparser and denser)

Saliency is defined as $\Delta_{lm} = \mathbb{E}[g|l,m] \cdot \mathbb{P}[l|m]$

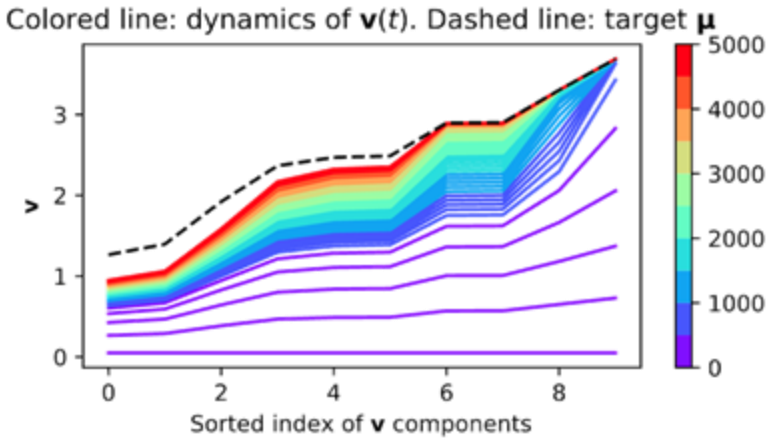
Discriminancy **CoOccurrence**

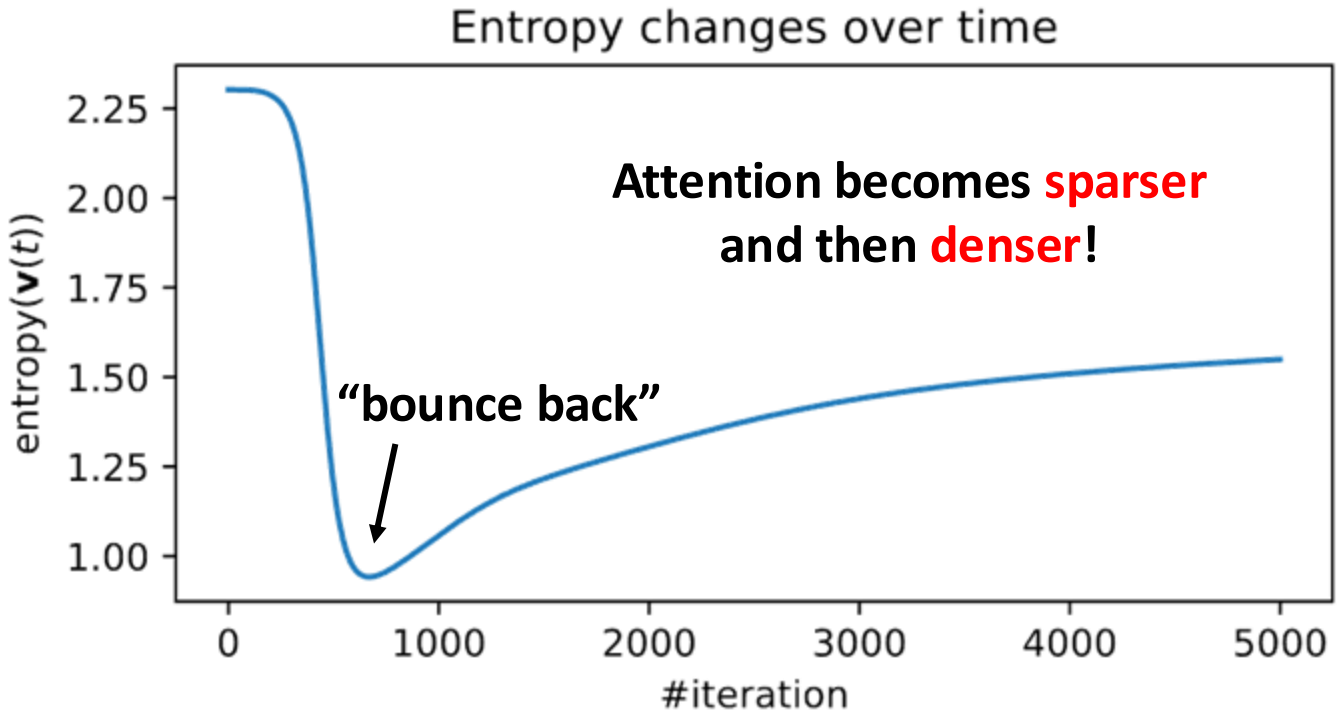
$\Delta_{lm} \approx 0$: **Common** tokens

$|\Delta_{lm}|$ large: **Distinct** tokens

Nonlinear case

	Nonlinear
$\dot{v} = (\mu - v) \circ \exp\left(\frac{v^2}{2}\right)$	Modified MLP (lower layer)

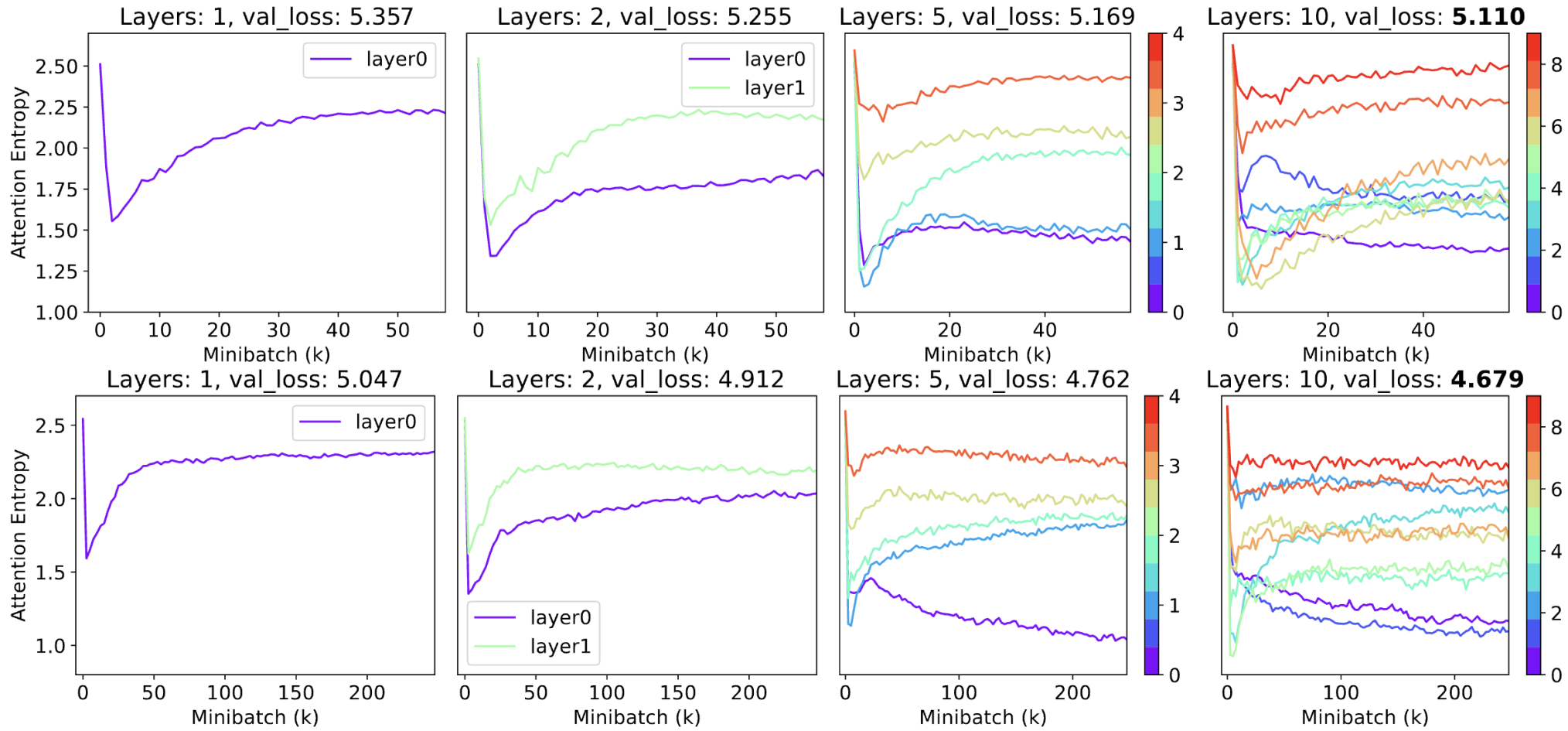




Real-world Experiments

Wikitext2

Wikitext103

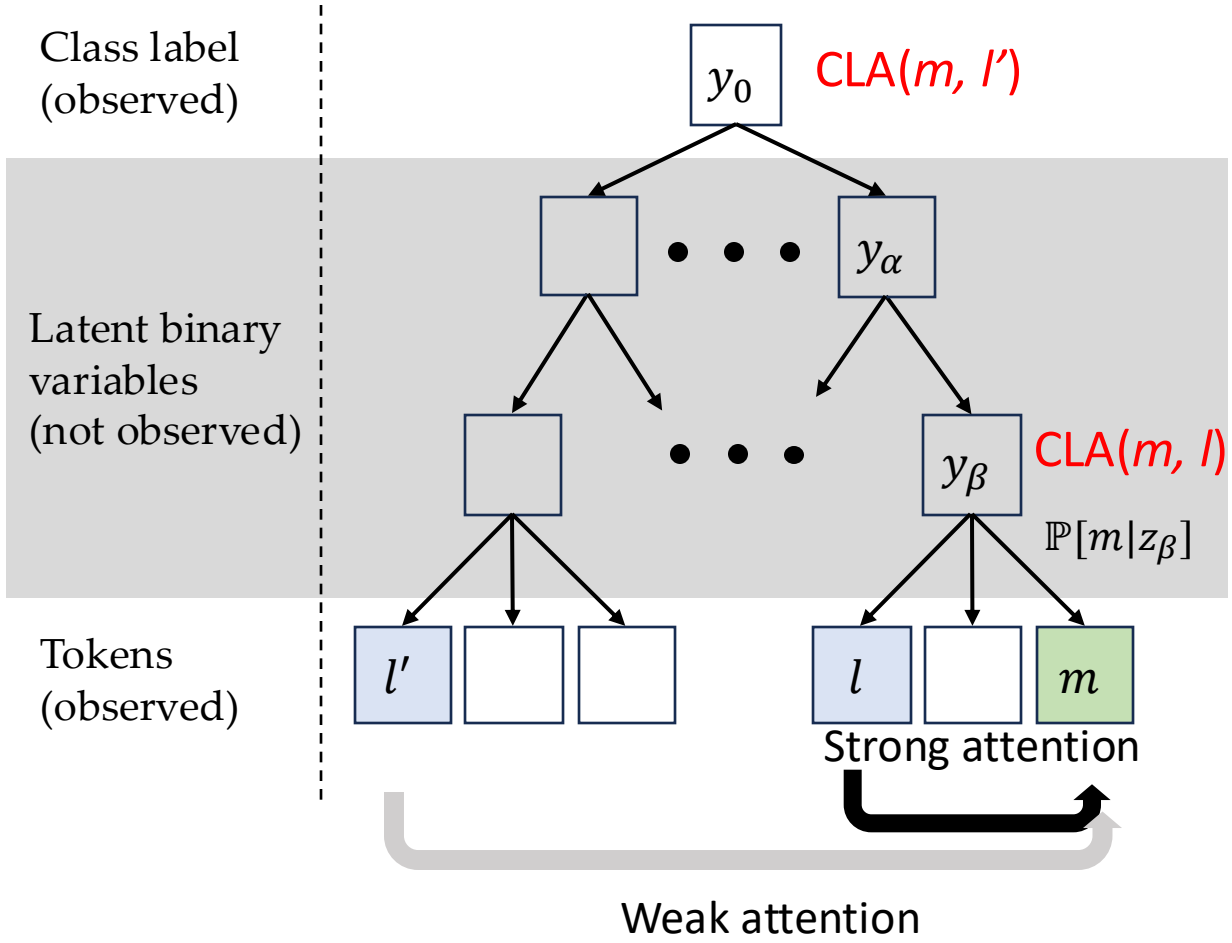


Why is this “bouncing back” property useful?

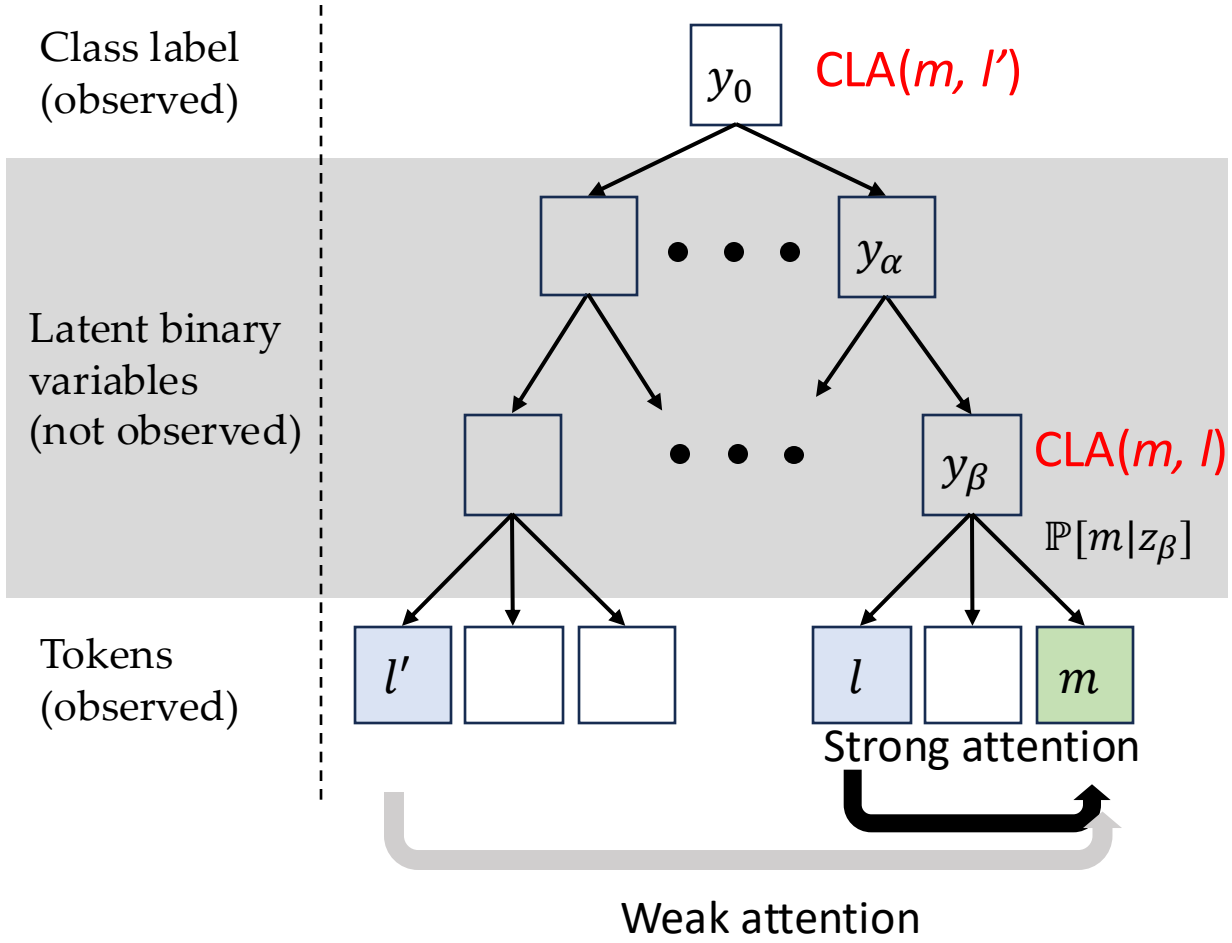
It seems that it only slows down the training??

Not useful in 1-layer, but useful in multiple Transformer layers!

Data Hierarchy & Multilayer Transformer



Data Hierarchy & Multilayer Transformer



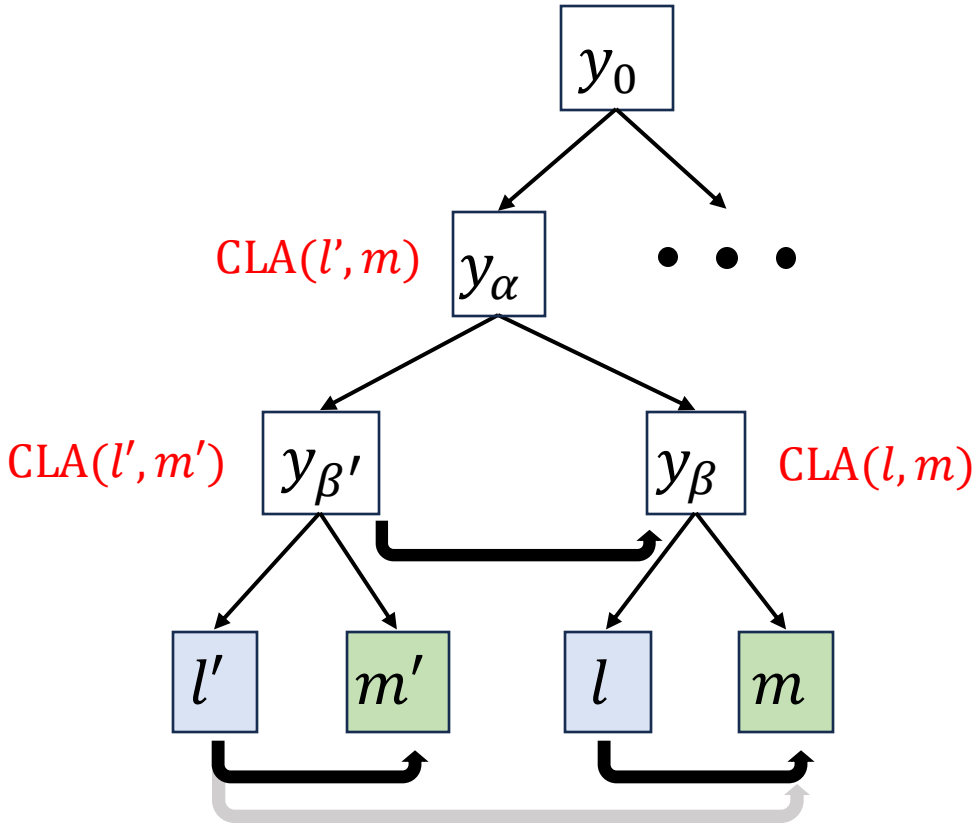
Theorem 5

$$\mathbb{P}[l|m] \approx 1 - \frac{H}{L}$$

H: height of the common latent ancestor (CLA) of l & m

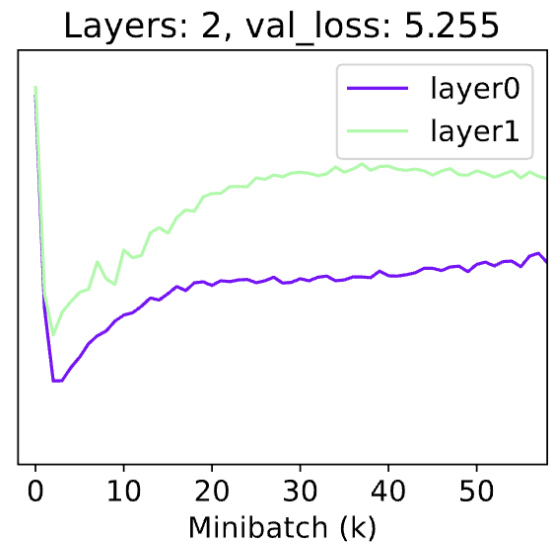
L: total height of the hierarchy

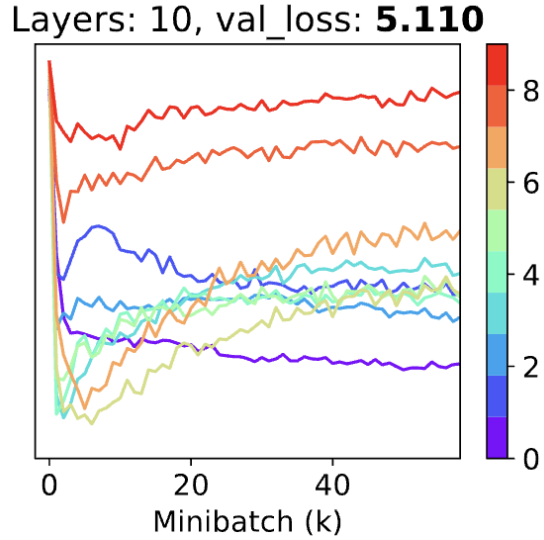
Deep Latent Distribution

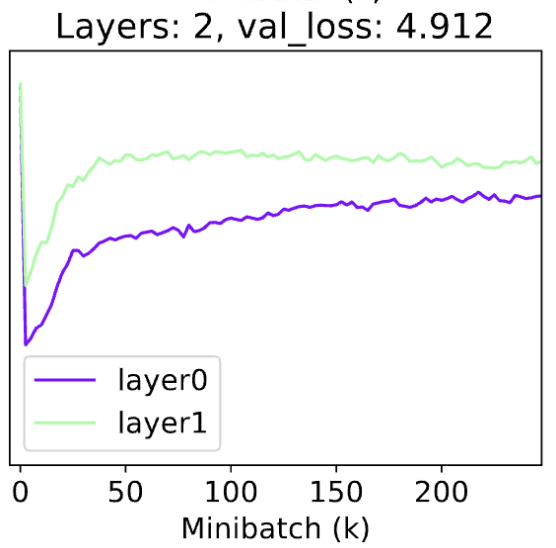


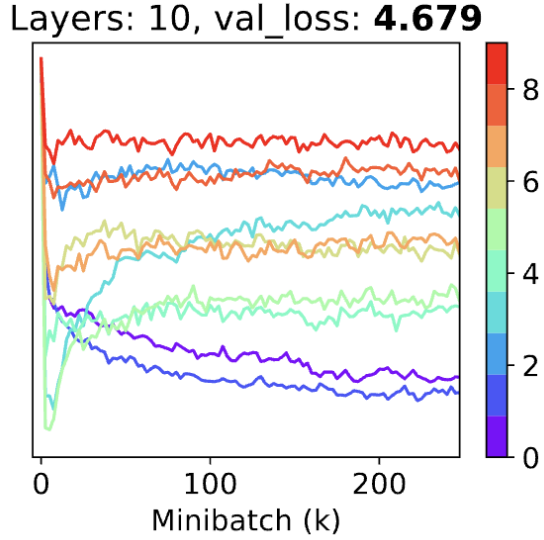
Strong Attention

Weak Attention







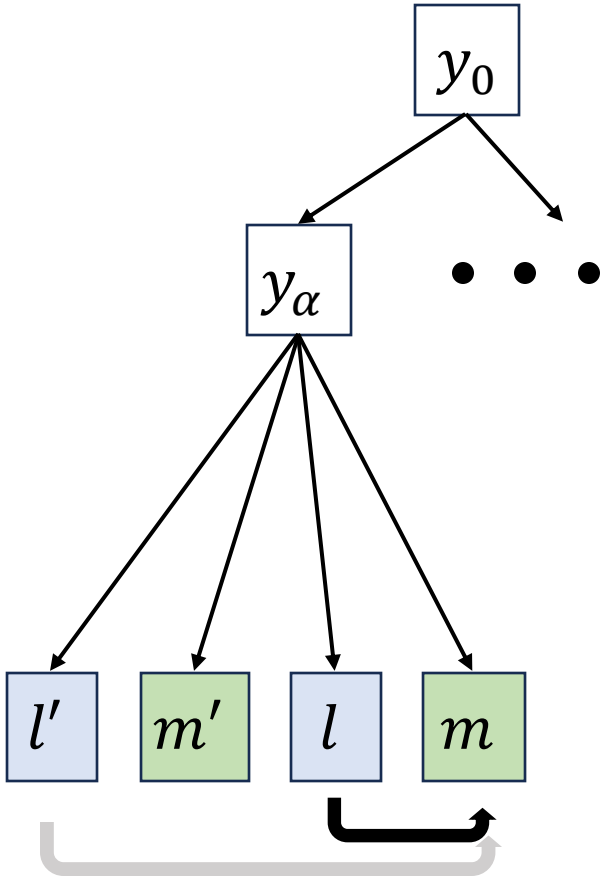


Learning the current hierarchical structure by
slowing down the association of tokens that are not directly correlated

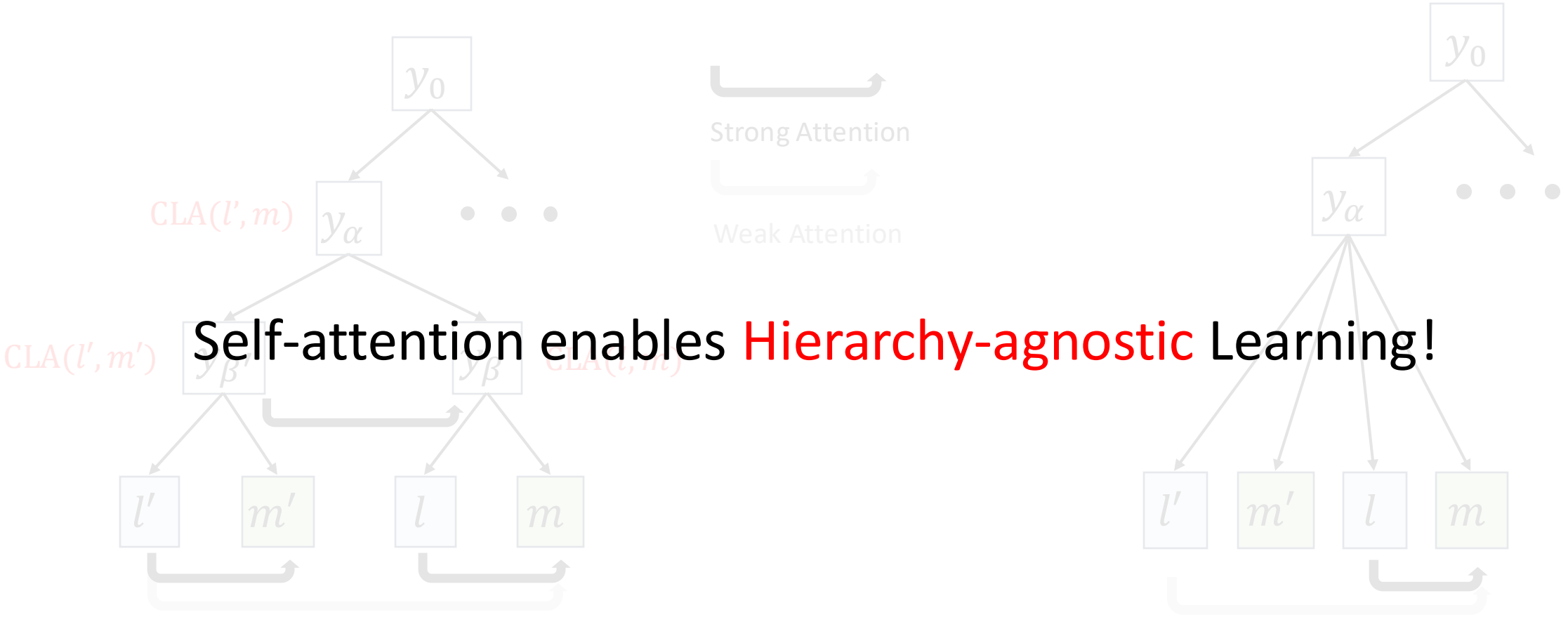
Shallow Latent Distribution

Strong Attention

Weak Attention

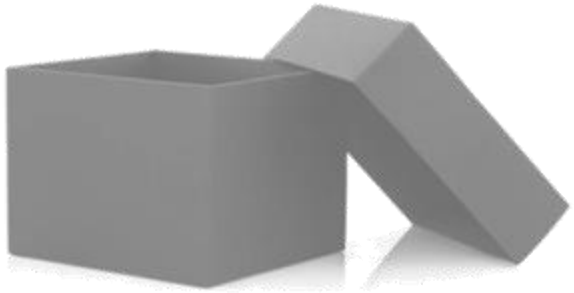


Hierarchy-agnostic Learning



Self-attention enables Hierarchy-agnostic Learning!

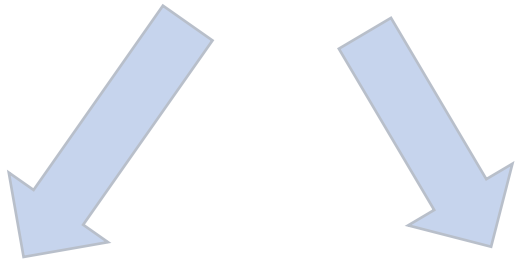
Take away messages



- Architecture ✓ training dynamics ✓
- Nonlinearity is not formidable!
  - Transformer can be analyzed following gradient descent rules
- Property of self-attention
  - Attention becomes sparse over training
  - Inductive bias
    - Favor the learning of strong co-occurred tokens
    - Deter the learning of weakly co-occurred tokens, avoiding spurious correlation.
- Key insights lead to broad applications

What Gradient Descent gives us?

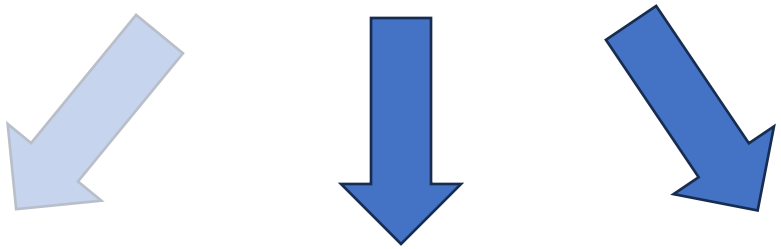
Simple Structures



Sparsity

Low-rank

More complicated structures



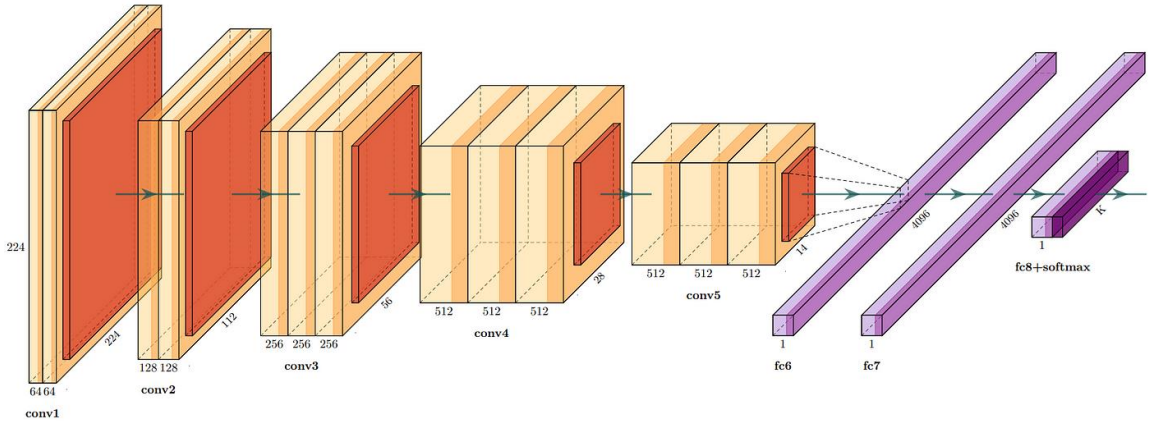
Hierarchical Representation

Algebraic Structure

Spectral Structure

Dichotomy: Symbolic and Neural Representation

Neural Representation



Symbolic Representation

$$\nabla \cdot \mathbf{E} = \frac{\rho_v}{\varepsilon} \qquad \text{(Gauss' Law)}$$

$$\nabla \cdot \mathbf{H} = 0 \qquad \text{(Gauss' Law for Magnetism)}$$

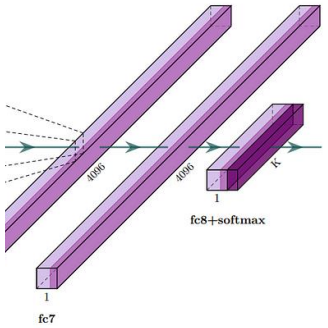
$$\nabla \times \mathbf{E} = -\mu \frac{\partial \mathbf{H}}{\partial t} \qquad \text{(Faraday's Law)}$$

$$\nabla \times \mathbf{H} = \mathbf{J} + \varepsilon \frac{\partial \mathbf{E}}{\partial t} \qquad \text{(Ampere's Law)}$$

Unification of Symbolic and Neural Representation

Emerging Symbolic Structure

Neural Repres



Symbo Repres

tism)

pere' s Law)

Deep Models

Concrete Example: **Modular Addition**

$$a + b = c \bmod d$$

Does neural network have an *implicit table* to do retrieval?

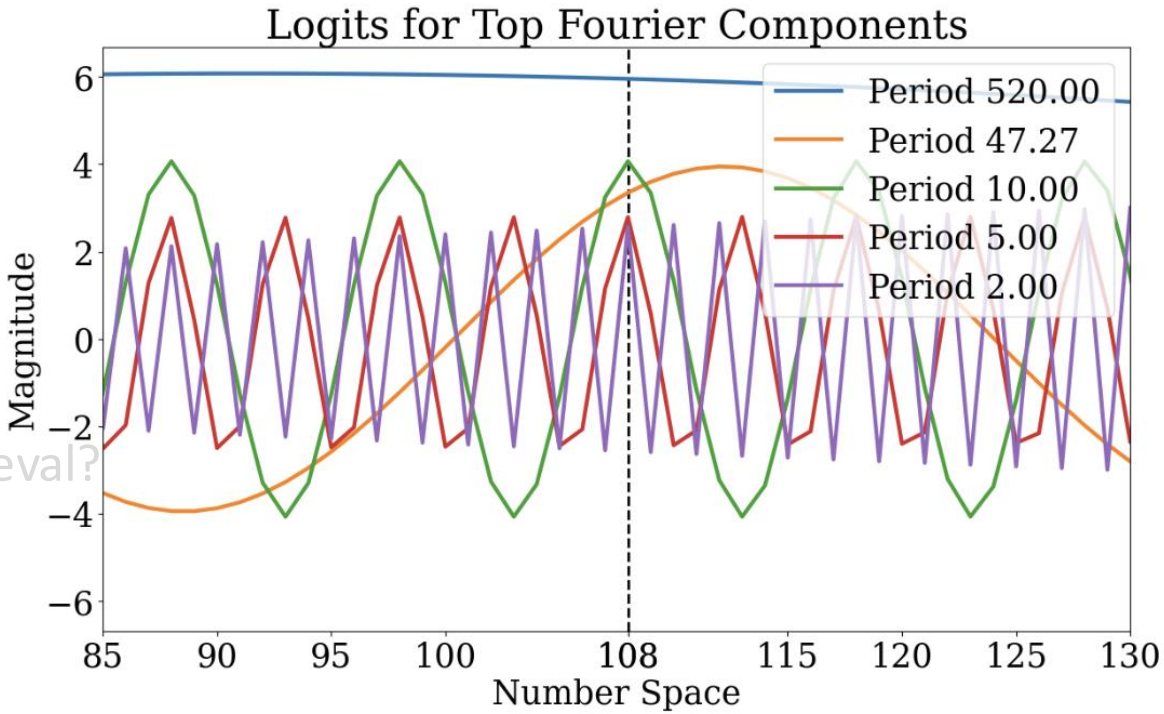
Concrete Example: **Modular Addition**

$$a + b = c \bmod d$$

Does neural network have an *implicit table* to do retrieval?

Learned representation = Fourier basis 😲

Why? 🤔

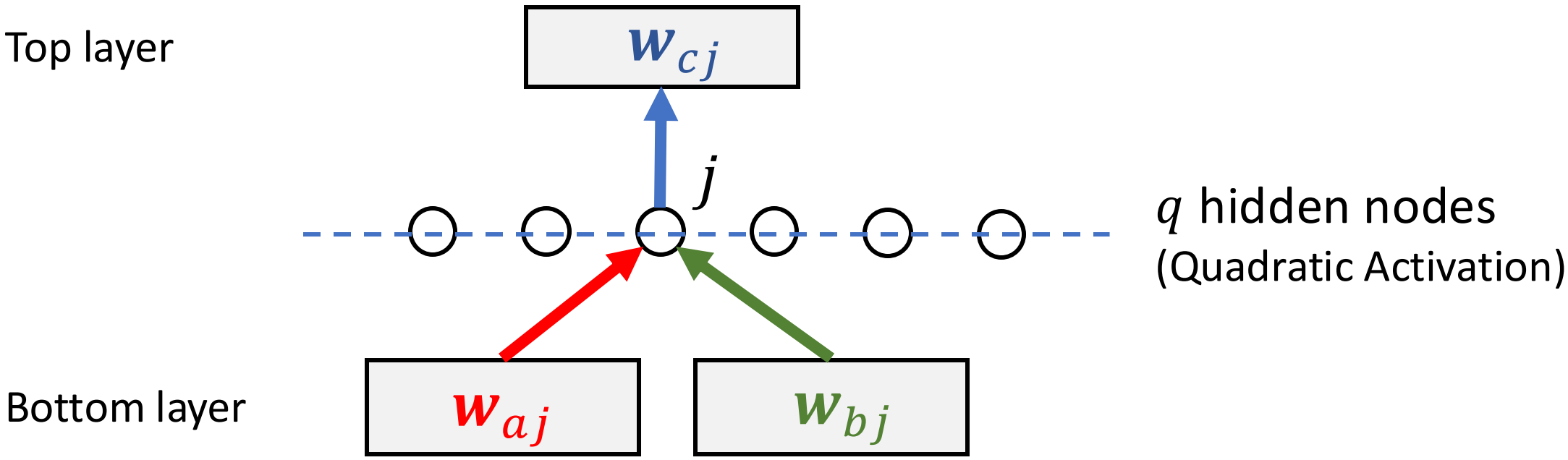


(a) Final logits for top Fourier components

[T. Zhou et al, *Pre-trained Large Language Models Use Fourier Features to Compute Addition*, NeurIPS'24]
[S. Kantamneni, *Language Models Use Trigonometry to Do Addition*, arXiv'25]

Minimal Setup

MSE Loss: $Min \ \|\text{Output} - \text{one-hot}(\boldsymbol{c})\|_2$

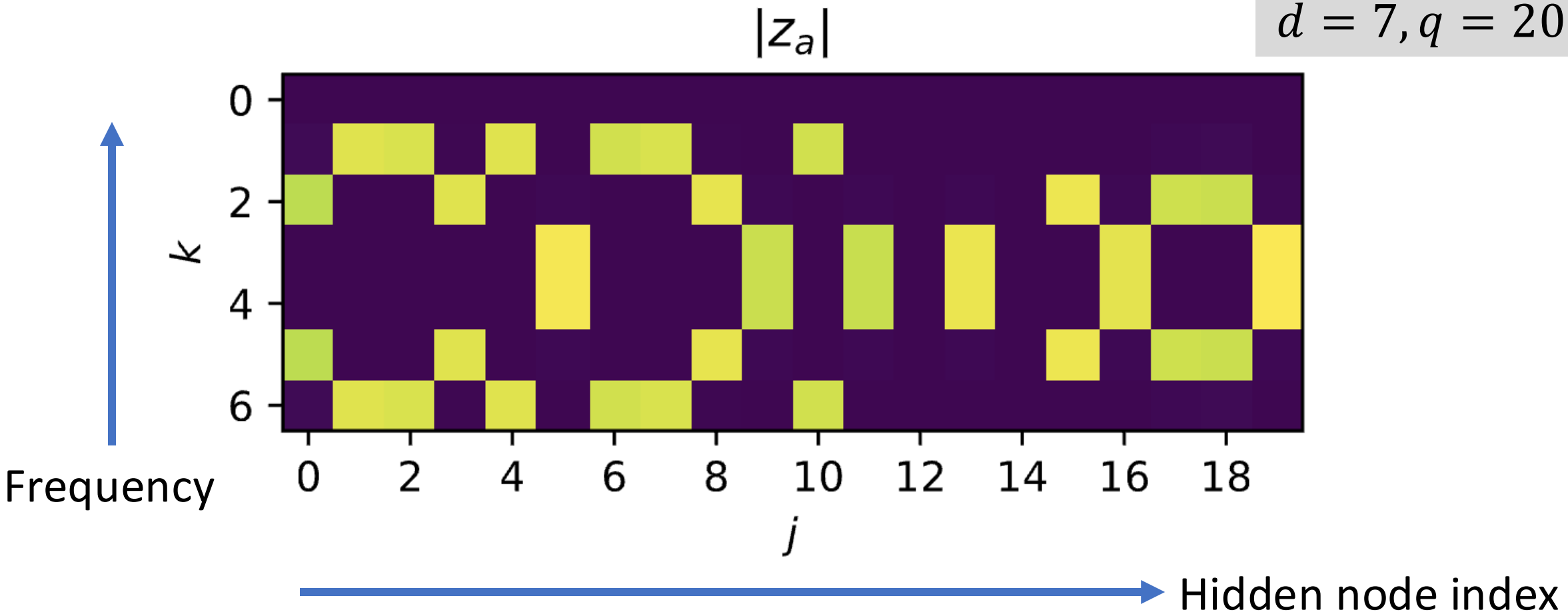


$$\boldsymbol{a} + \boldsymbol{b} = \boldsymbol{c} \bmod d$$

[Y. Tian, *Composing Global Optimizers to Reasoning Tasks via Algebraic Objects in Neural Nets*, arXiv'24]

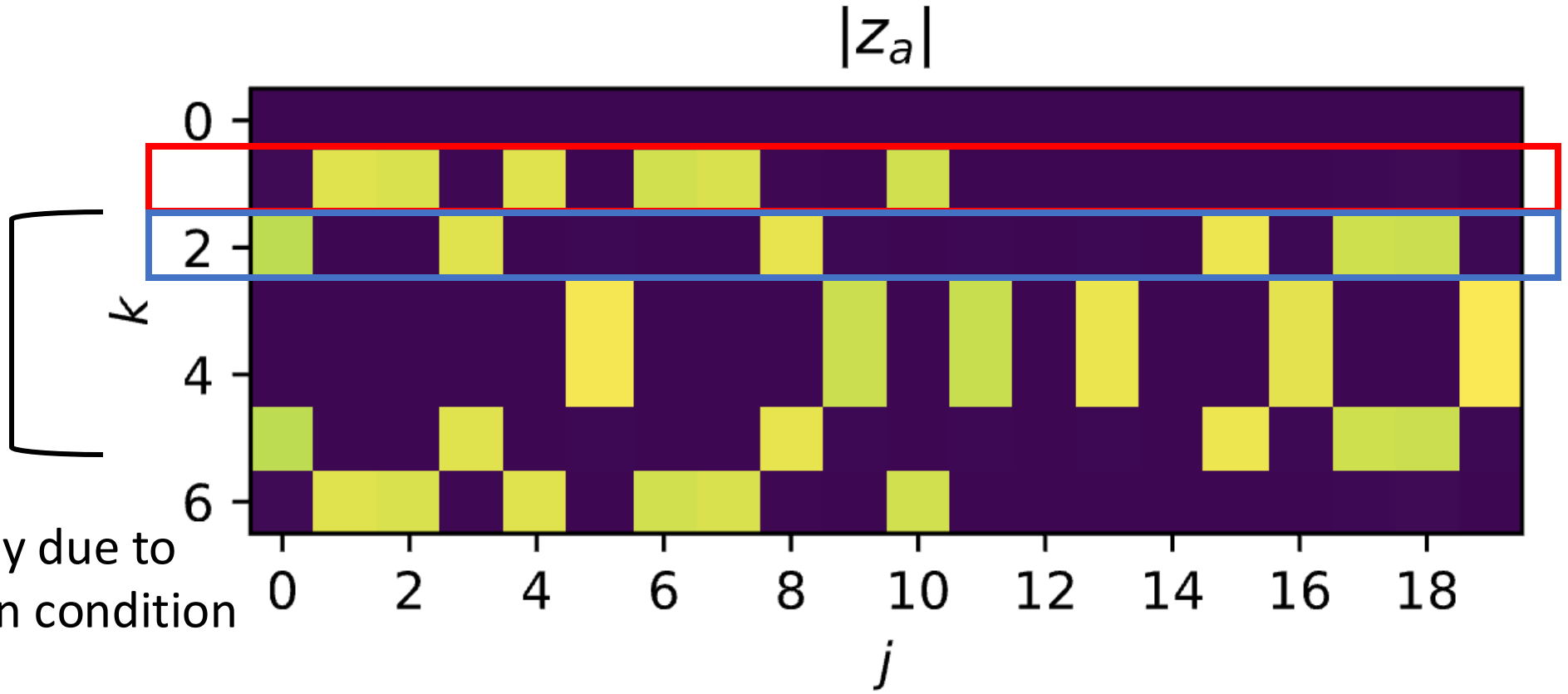
What a Gradient Descent Solution look like?

$d = 7, q = 20$



[Y. Tian, *Composing Global Optimizers to Reasoning Tasks via Algebraic Objects in Neural Nets*, arXiv'24]

What a Gradient Descent Solution look like?

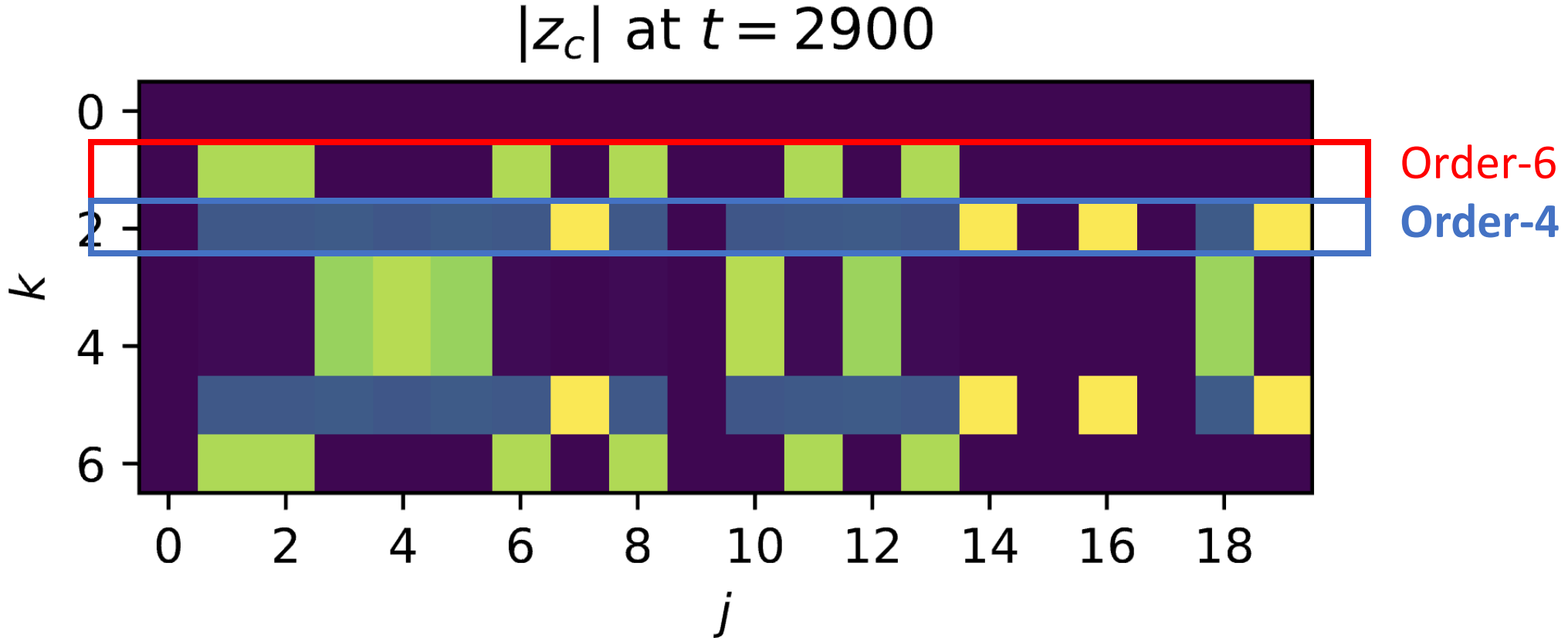


Order-6 solutions

Symmetry due to Hermitian condition

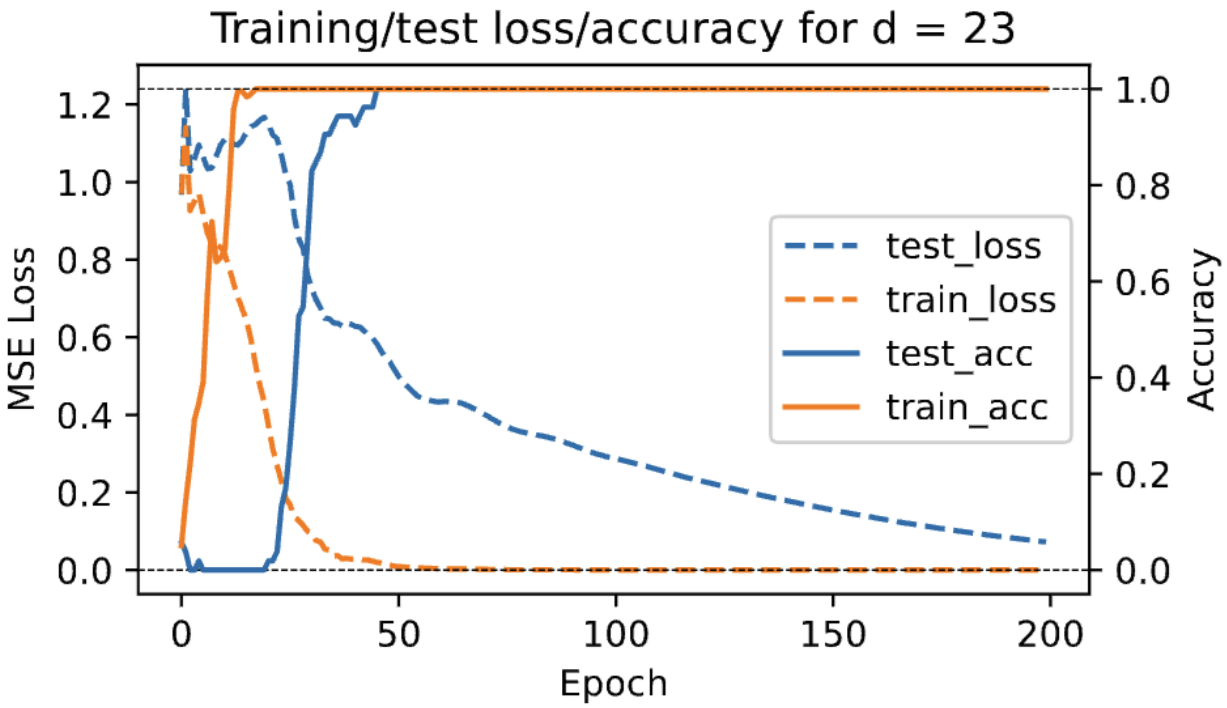
[Y. Tian, *Composing Global Optimizers to Reasoning Tasks via Algebraic Objects in Neural Nets*, arXiv'24]

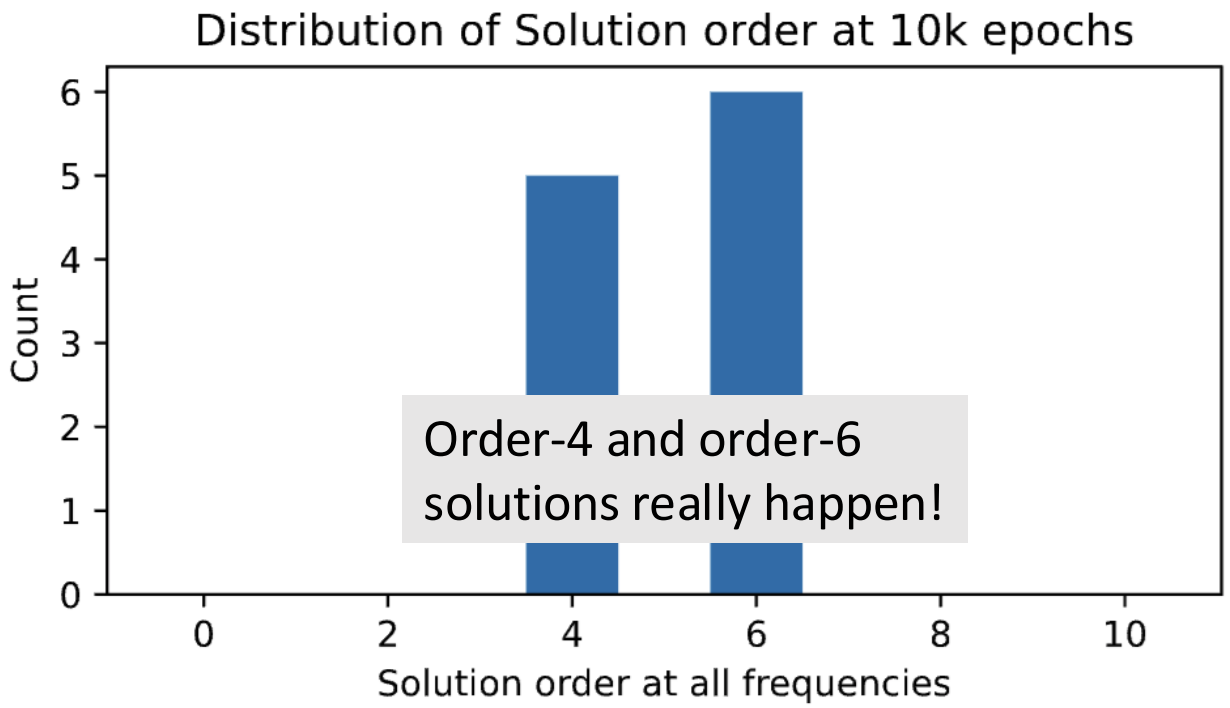
What a Gradient Descent Solution look like?



[Y. Tian, *Composing Global Optimizers to Reasoning Tasks via Algebraic Objects in Neural Nets*, arXiv'24]

More Statistics on Gradient Descent Solutions

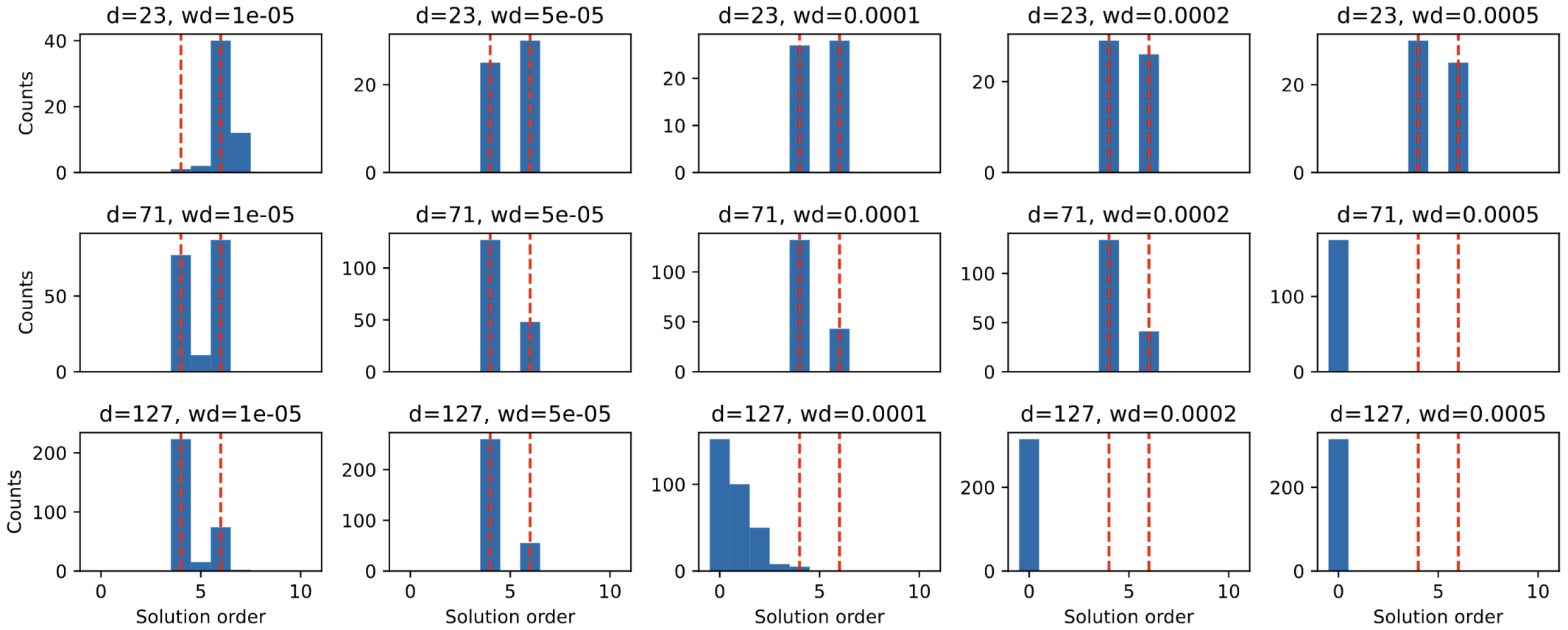




[Y. Tian, *Composing Global Optimizers to Reasoning Tasks via Algebraic Objects in Neural Nets*, arXiv'24]

Effect of Weight Decay

Stronger weight decay



[Y. Tian, *Composing Global Optimizers to Reasoning Tasks via Algebraic Objects in Neural Nets*, arXiv'24]

Why? 

How to Optimize?

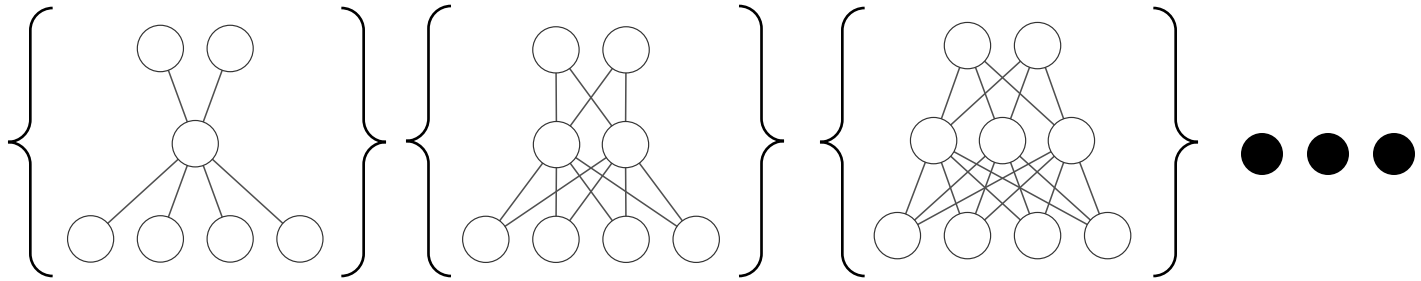
The objective is highly nonlinear !!

However, nice *algebraic structures* exist!

How to Optimize?

The objective is highly nonlinear !!

However, nice *algebraic structures* exist!



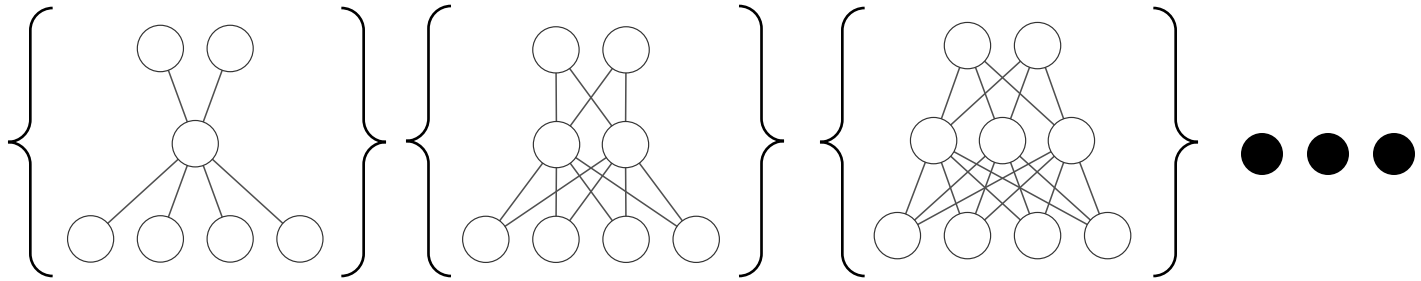
$\mathcal{Z}_1$ $\mathcal{Z}_2$ $\mathcal{Z}_3$

$\mathcal{Z} = \bigcup_{q \geq 0} \mathcal{Z}_q$: All 2-layer networks with different number of hidden nodes

How to Optimize?

The objective is highly nonlinear !!

However, nice *algebraic structures* exist!



$\mathcal{Z} = \bigcup_{q \geq 0} \mathcal{Z}_q$: All 2-layer networks with different number of hidden nodes

Ring addition $+$: Concatenate hidden nodes

Ring multiplication $*$: Kronecker production along the hidden dimensions

$\langle \mathcal{Z}, +, * \rangle$ is a ***semi-ring***

Composing Global Optimizers from Partial Ones

Partial solution #1

$\boldsymbol{z}_{\text{syn}}^{(k)} \in R_{\text{c}} \cap R_{\text{n}}$ but $\boldsymbol{z}_{\text{syn}}^{(k)} \notin R_{*}$

Partial solution #2

$\boldsymbol{z}_{\nu}^{(k)} \in R_{*}$

[Y. Tian, *Composing Global Optimizers to Reasoning Tasks via Algebraic Objects in Neural Nets*, arXiv'24]

Composing Global Optimizers from Partial Ones

Compositing solutions using ***ring multiplication*** $*$

Partial solution #1

$\boldsymbol{z}_{\text{syn}}^{(k)} \in R_{\text{c}} \cap R_{\text{n}}$ but $\boldsymbol{z}_{\text{syn}}^{(k)} \notin R_*$

Partial solution #2

$\boldsymbol{z}_{\nu}^{(k)} \in R_*$

Better solution

$\boldsymbol{z}_{\text{syn}}^{(k)} * \boldsymbol{z}_{\nu}^{(k)} \in R_{\text{c}} \cap R_{\text{n}} \cap R_*$

[Y. Tian, *Composing Global Optimizers to Reasoning Tasks via Algebraic Objects in Neural Nets*, arXiv'24]

Composing Global Optimizers from Partial Ones

Compositing solutions using ***ring multiplication*** $*$

Compositing solutions using ***ring addition*** $+$

Partial solution #1

$\boldsymbol{z}_{\text{syn}}^{(k)} \in R_{\text{c}} \cap R_{\text{n}}$ but $\boldsymbol{z}_{\text{syn}}^{(k)} \notin R_*$

Partial solution #2

$\boldsymbol{z}_{\nu}^{(k)} \in R_*$

Better solution

$\boldsymbol{z}_{\text{syn}}^{(k)} * \boldsymbol{z}_{\nu}^{(k)} \in R_{\text{c}} \cap R_{\text{n}} \cap R_*$

Global Optimizer to MSE loss $\ell(z)$!

$$\boldsymbol{z}_{F6} = \frac{\mathbf{1}}{\sqrt[3]{6}} \sum_k \boldsymbol{z}_{\text{syn}}^{(k)} * \boldsymbol{z}_{\nu}^{(k)}$$

[Y. Tian, *Composing Global Optimizers to Reasoning Tasks via Algebraic Objects in Neural Nets*, arXiv'24]

Exemplar constructed global optimizers

Order-6 $\boldsymbol{z}_{F6}$ (2*3)

$$\boldsymbol{z}_{F6} = \frac{1}{\sqrt[3]{6}} \sum_{k=1}^{(d-1)/2} \boldsymbol{z}_{\mathrm{syn}}^{(k)} * \boldsymbol{z}_{\nu}^{(k)} * \boldsymbol{y}_k$$

Exemplar constructed global optimizers

Order-6 $\boldsymbol{z}_{F6}$ (2*3)

$$\boldsymbol{z}_{F6} = \frac{1}{\sqrt[3]{6}} \sum_{k=1}^{(d-1)/2} \boldsymbol{z}_{\text{syn}}^{(k)} * \boldsymbol{z}_{\nu}^{(k)} * \boldsymbol{y}_k$$

Order-4 $\boldsymbol{z}_{F4/6}$ (2*2)
(mixed with order-6)

$$\boldsymbol{z}_{F4/6} = \frac{1}{\sqrt[3]{6}} \hat{\boldsymbol{z}}_{F6}^{(k_0)} + \frac{1}{\sqrt[3]{4}} \sum_{k=1, k\neq k_0}^{(d-1)/2} \boldsymbol{z}_{F4}^{(k)}$$

Exemplar constructed global optimizers

Order-6 $\boldsymbol{z}_{F6}$ (2*3)

$$\boldsymbol{z}_{F6} = \frac{1}{\sqrt[3]{6}} \sum_{k=1}^{(d-1)/2} \boldsymbol{z}_{\text{syn}}^{(k)} * \boldsymbol{z}_{\nu}^{(k)} * \boldsymbol{y}_k$$

Order-4 $\boldsymbol{z}_{F4/6}$ (2*2)
(mixed with order-6)

$$\boldsymbol{z}_{F4/6} = \frac{1}{\sqrt[3]{6}} \hat{\boldsymbol{z}}_{F6}^{(k_0)} + \frac{1}{\sqrt[3]{4}} \sum_{k=1, k\neq k_0}^{(d-1)/2} \boldsymbol{z}_{F4}^{(k)}$$

Perfect memorization
(order-d per frequency)

$$\boldsymbol{z}_a = \sum_{j=0}^{d-1} \boldsymbol{u}_a^j, \qquad \boldsymbol{z}_b = \sum_{j=0}^{d-1} \boldsymbol{u}_b^j$$

$$\boldsymbol{z}_M = d^{-2/3} \boldsymbol{z}_a * \boldsymbol{z}_b$$

Gradient Descent solutions matches with construction

d	%not order-4/6	%non-factorable order-4	%non-factorable order-6	error ($\times 10^{-2}$) order-4	error ($\times 10^{-2}$) order-6	solution distribution (%) in factorable ones $z^{(k)}_{\nu=\mathrm{i}} * z^{(k)}_{\xi}$	$z^{(k)}_{\nu=\mathrm{i}} * z^{(k)}_{\mathrm{syn},\alpha\beta}$	$z^{(k)}_{\nu} * z^{(k)}_{\mathrm{syn}}$	others
23	$0.0_{\pm 0.0}$	$0.00_{\pm 0.00}$	$5.71_{\pm 5.71}$	$0.05_{\pm 0.01}$	$4.80_{\pm 0.96}$	$47.07_{\pm 1.88}$	$11.31_{\pm 1.76}$	$39.80_{\pm 2.11}$	$1.82_{\pm 1.82}$
71	$0.0_{\pm 0.0}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.03_{\pm 0.00}$	$5.02_{\pm 0.25}$	$72.57_{\pm 0.70}$	$4.00_{\pm 1.14}$	$21.14_{\pm 2.14}$	$2.29_{\pm 1.07}$
127	$0.0_{\pm 0.0}$	$1.50_{\pm 0.92}$	$0.00_{\pm 0.00}$	$0.26_{\pm 0.14}$	$0.93_{\pm 0.18}$	$82.96_{\pm 0.39}$	$2.25_{\pm 0.64}$	$14.13_{\pm 0.87}$	$0.66_{\pm 0.66}$

$$q = 512, wd = 5 \cdot 10^{-5}$$

Gradient Descent solutions matches with construction

d	%not order-4/6	%non-factorable order-4	%non-factorable order-6	error ($\times 10^{-2}$) order-4	error ($\times 10^{-2}$) order-6	solution distribution (%) in factorable ones $z^{(k)}_{\nu=\mathrm{i}} * z^{(k)}_{\xi}$	$z^{(k)}_{\nu=\mathrm{i}} * z^{(k)}_{\mathrm{syn},\alpha\beta}$	$z^{(k)}_{\nu} * z^{(k)}_{\mathrm{syn}}$	others
23	$0.0_{\pm 0.0}$	$0.00_{\pm 0.00}$	$5.71_{\pm 5.71}$	$0.05_{\pm 0.01}$	$4.80_{\pm 0.96}$	$47.07_{\pm 1.88}$	$11.31_{\pm 1.76}$	$39.80_{\pm 2.11}$	$1.82_{\pm 1.82}$
71	$0.0_{\pm 0.0}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.03_{\pm 0.00}$	$5.02_{\pm 0.25}$	$72.57_{\pm 0.70}$	$4.00_{\pm 1.14}$	$21.14_{\pm 2.14}$	$2.29_{\pm 1.07}$
127	$0.0_{\pm 0.0}$	$1.50_{\pm 0.92}$	$0.00_{\pm 0.00}$	$0.26_{\pm 0.14}$	$0.93_{\pm 0.18}$	$82.96_{\pm 0.39}$	$2.25_{\pm 0.64}$	$14.13_{\pm 0.87}$	$0.66_{\pm 0.66}$

100% of the per-freq solutions are order-4/6

Gradient Descent solutions matches with construction

d	%not order-4/6	%non-factorable order-4	%non-factorable order-6	error ($\times 10^{-2}$) order-4	error ($\times 10^{-2}$) order-6	solution distribution (%) in factorable ones $z^{(k)}_{\nu=\mathrm{i}} * z^{(k)}_{\xi}$	$z^{(k)}_{\nu=\mathrm{i}} * z^{(k)}_{\mathrm{syn},\alpha\beta}$	$z^{(k)}_{\nu} * z^{(k)}_{\mathrm{syn}}$	others
23	$0.0_{\pm 0.0}$	$0.00_{\pm 0.00}$	$5.71_{\pm 5.71}$	$0.05_{\pm 0.01}$	$4.80_{\pm 0.96}$	$47.07_{\pm 1.88}$	$11.31_{\pm 1.76}$	$39.80_{\pm 2.11}$	$1.82_{\pm 1.82}$
71	$0.0_{\pm 0.0}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.03_{\pm 0.00}$	$5.02_{\pm 0.25}$	$72.57_{\pm 0.70}$	$4.00_{\pm 1.14}$	$21.14_{\pm 2.14}$	$2.29_{\pm 1.07}$
127	$0.0_{\pm 0.0}$	$1.50_{\pm 0.92}$	$0.00_{\pm 0.00}$	$0.26_{\pm 0.14}$	$0.93_{\pm 0.18}$	$82.96_{\pm 0.39}$	$2.25_{\pm 0.64}$	$14.13_{\pm 0.87}$	$0.66_{\pm 0.66}$

95% of the solutions are factorizable into “2*3” or “2*2”

Gradient Descent solutions matches with construction

d	%not order-4/6	%non-factorable order-4	%non-factorable order-6	error ($\times 10^{-2}$) order-4	error ($\times 10^{-2}$) order-6	solution distribution (%) in factorable ones $\boldsymbol{z}^{(k)}_{\nu=\mathrm{i}} * \boldsymbol{z}^{(k)}_{\xi}$	solution distribution (%) in factorable ones $\boldsymbol{z}^{(k)}_{\nu=\mathrm{i}} * \boldsymbol{z}^{(k)}_{\mathrm{syn},\alpha\beta}$	solution distribution (%) in factorable ones $\boldsymbol{z}^{(k)}_{\nu} * \boldsymbol{z}^{(k)}_{\mathrm{syn}}$	solution distribution (%) in factorable ones others
23	$0.0_{\pm 0.0}$	$0.00_{\pm 0.00}$	$5.71_{\pm 5.71}$	$0.05_{\pm 0.01}$	$4.80_{\pm 0.96}$	$47.07_{\pm 1.88}$	$11.31_{\pm 1.76}$	$39.80_{\pm 2.11}$	$1.82_{\pm 1.82}$
71	$0.0_{\pm 0.0}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.03_{\pm 0.00}$	$5.02_{\pm 0.25}$	$72.57_{\pm 0.70}$	$4.00_{\pm 1.14}$	$21.14_{\pm 2.14}$	$2.29_{\pm 1.07}$
127	$0.0_{\pm 0.0}$	$1.50_{\pm 0.92}$	$0.00_{\pm 0.00}$	$0.26_{\pm 0.14}$	$0.93_{\pm 0.18}$	$82.96_{\pm 0.39}$	$2.25_{\pm 0.64}$	$14.13_{\pm 0.87}$	$0.66_{\pm 0.66}$

Factorization error is very small

Gradient Descent solutions matches with construction

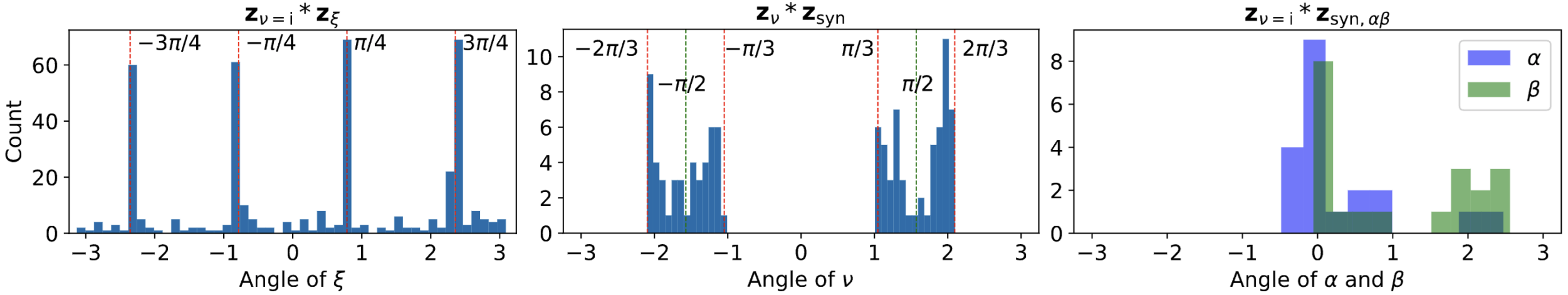
d	%not order-4/6	%non-factorable order-4	%non-factorable order-6	error ($\times 10^{-2}$) order-4	error ($\times 10^{-2}$) order-6	solution distribution (%) in factorable ones $\boldsymbol{z}^{(k)}_{\nu=\mathrm{i}} * \boldsymbol{z}^{(k)}_{\xi}$	solution distribution (%) in factorable ones $\boldsymbol{z}^{(k)}_{\nu=\mathrm{i}} * \boldsymbol{z}^{(k)}_{\mathrm{syn},\alpha\beta}$	solution distribution (%) in factorable ones $\boldsymbol{z}^{(k)}_{\nu} * \boldsymbol{z}^{(k)}_{\mathrm{syn}}$	solution distribution (%) in factorable ones others
23	$0.0_{\pm 0.0}$	$0.00_{\pm 0.00}$	$5.71_{\pm 5.71}$	$0.05_{\pm 0.01}$	$4.80_{\pm 0.96}$	$47.07_{\pm 1.88}$	$11.31_{\pm 1.76}$	$39.80_{\pm 2.11}$	$1.82_{\pm 1.82}$
71	$0.0_{\pm 0.0}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.03_{\pm 0.00}$	$5.02_{\pm 0.25}$	$72.57_{\pm 0.70}$	$4.00_{\pm 1.14}$	$21.14_{\pm 2.14}$	$2.29_{\pm 1.07}$
127	$0.0_{\pm 0.0}$	$1.50_{\pm 0.92}$	$0.00_{\pm 0.00}$	$0.26_{\pm 0.14}$	$0.93_{\pm 0.18}$	$82.96_{\pm 0.39}$	$2.25_{\pm 0.64}$	$14.13_{\pm 0.87}$	$0.66_{\pm 0.66}$

98% of the solutions can be factorizable into the constructed forms

Gradient Descent solutions matches with construction

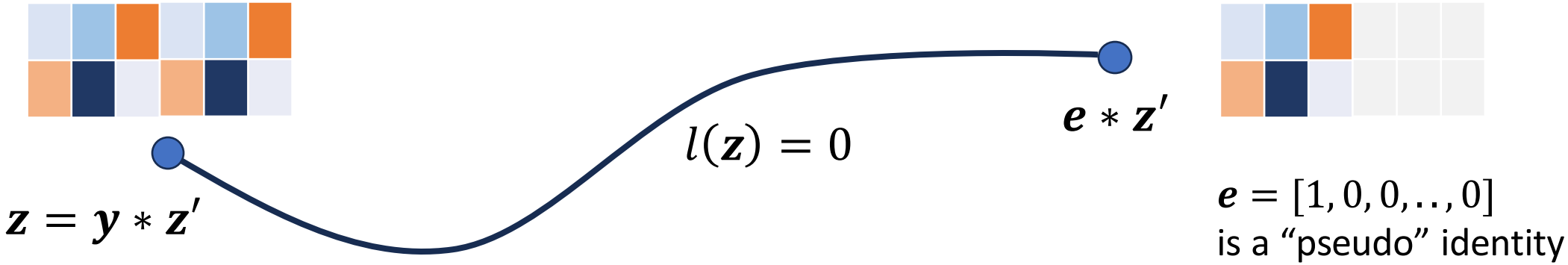
d	%not order-4/6	%non-factorable order-4	%non-factorable order-6	error ($\times 10^{-2}$) order-4	error ($\times 10^{-2}$) order-6	solution distribution (%) in factorable ones $z_{\nu=\mathrm{i}}^{(k)} * z_{\xi}^{(k)}$	$z_{\nu=\mathrm{i}}^{(k)} * z_{\mathrm{syn},\alpha\beta}^{(k)}$	$z_{\nu}^{(k)} * z_{\mathrm{syn}}^{(k)}$	others
23	$0.0_{\pm 0.0}$	$0.00_{\pm 0.00}$	$5.71_{\pm 5.71}$	$0.05_{\pm 0.01}$	$4.80_{\pm 0.96}$	$47.07_{\pm 1.88}$	$11.31_{\pm 1.76}$	$39.80_{\pm 2.11}$	$1.82_{\pm 1.82}$
5						$72.57_{\pm 0.70}$	$4.00_{\pm 1.14}$	$21.14_{\pm 2.14}$	$2.29_{\pm 1.07}$
8						$82.96_{\pm 0.39}$	$2.25_{\pm 0.64}$	$14.13_{\pm 0.87}$	$0.66_{\pm 0.66}$

Distribution of the parameters in the solutions



Gradient Dynamics

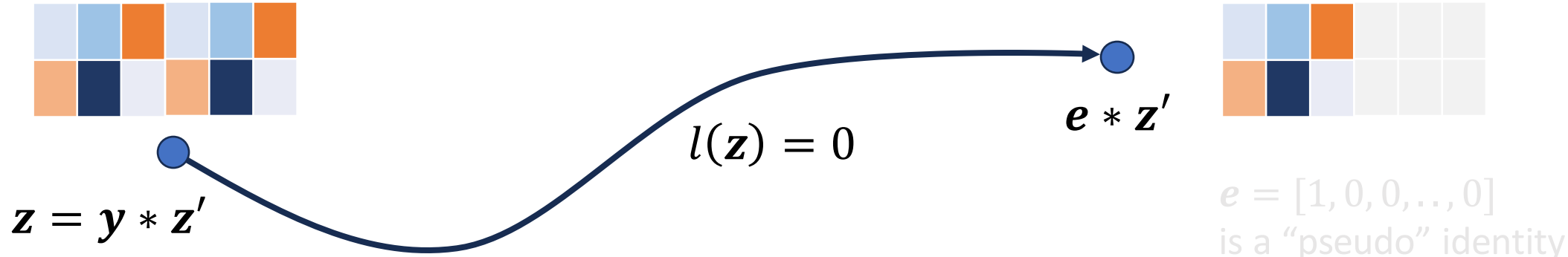
Theorem [**The Occam's Razer**] If $\boldsymbol{z} = \boldsymbol{y} * \boldsymbol{z}'$ and both $\boldsymbol{z}$ and $\boldsymbol{z}'$ are global optimal, then there exists a path of zero loss connecting $\boldsymbol{z}$ and $\boldsymbol{z}'$.



$\boldsymbol{e} = [1, 0, 0, .., 0]$
is a "pseudo" identity

Gradient Dynamics

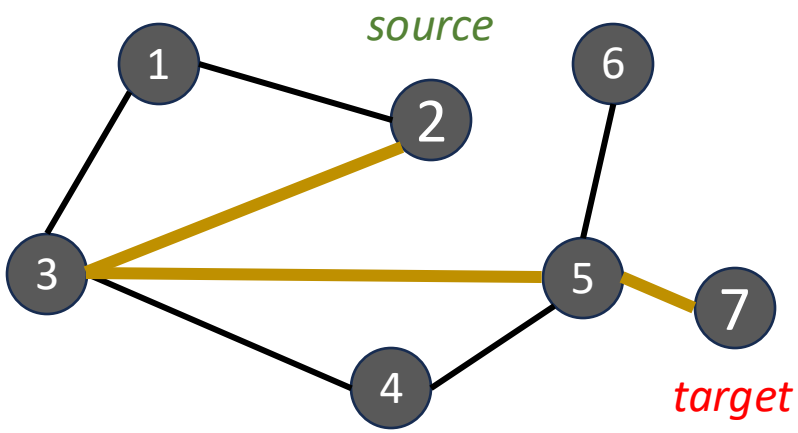
Theorem [**The Occam's Razer**] If $\boldsymbol{z} = \boldsymbol{y} * \boldsymbol{z}'$ and both $\boldsymbol{z}$ and $\boldsymbol{z}'$ are global optimal, then there exists a path of zero loss connecting $\boldsymbol{z}$ and $\boldsymbol{z}'$.



L2 regularization will push the solution to $\boldsymbol{e} * \boldsymbol{z}'$ (simpler solutions), since $\|\boldsymbol{e} * \boldsymbol{z}'\|_2 \leq \|\boldsymbol{y} * \boldsymbol{z}'\|_2$

Another Example: Symbolic from Neural Representation

Task: Learn a 2-layer Transformer for predicting **shortest path** in the graph



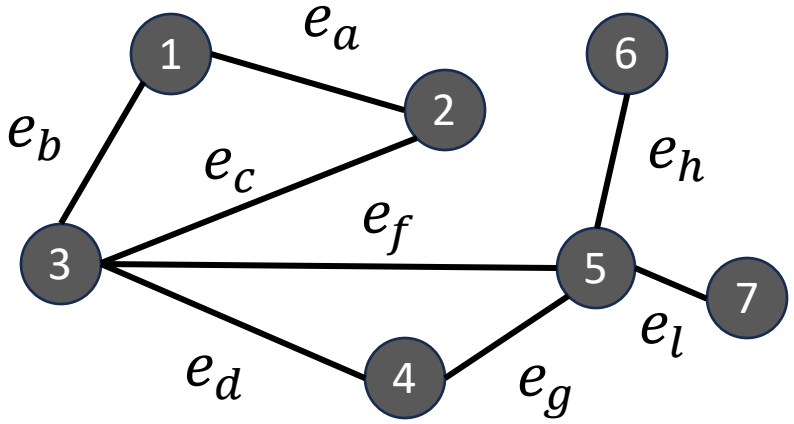
<bos> 1 2 <e> … <q> [source] [target] <p> [source] [node 1] [node 2] … [target]

Context

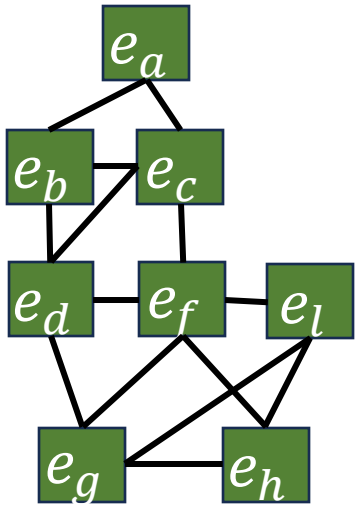
Predicted Shortest path

[A. Cohen et al, *Spectral Journey: How Transformers Predict the Shortest Path,* ICLR Workshop 2025]

What representations it learns?



Line graph



Normalized Graph Laplacian

$$L = I - D^{-1/2} A D^{-1/2}$$

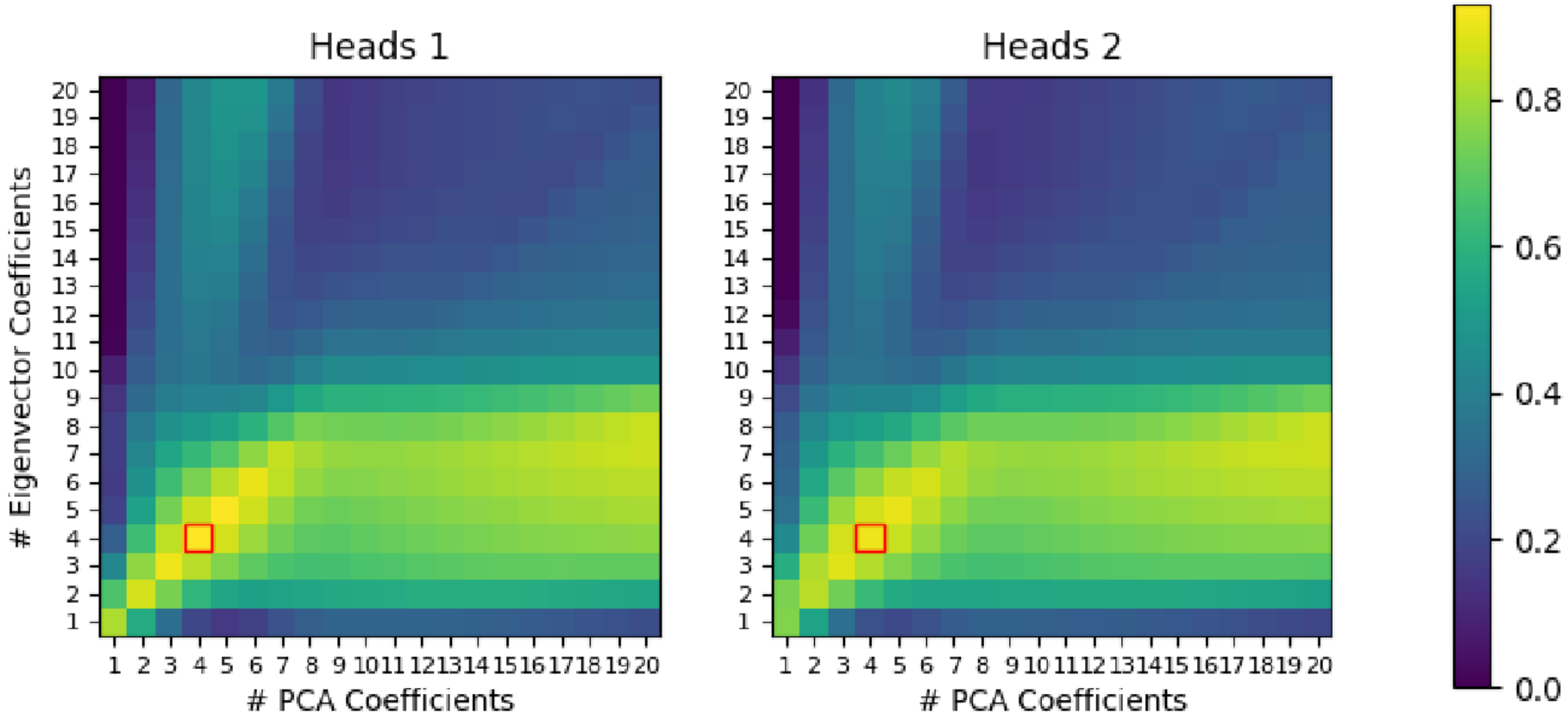
Edge Embedding

Representation after the first Transformer layer (averaged over random edge order)

<bos> 1 2 **<e>** … <q> [source] [target] <p> [source] [node 1] [node 2] … [target]

What representations it learns?

Graph Edge Embedding of various dimensions



Computed edge embedding with trained Transformers

Normalized Correlation > 0.9

Spectral Line Navigator (SLN)

Simple Algorithms of Graph Shortest Path

1. Compute Line Graph $\tilde{G}$ of existing graph G
2. Compute eigenvectors of normalized Laplacian $L(\tilde{G})$
3. $i = source$
4. While $i \neq target$ do

   $distance(j,k;i) := \left\| v_{ij} - v_{k,target} \right\|_2$

   Find $j = \text{argmin}_{j,k}\, distance(j,k;i)$

   Let $i = j$

>99% optimal for small random graph (size < 10)

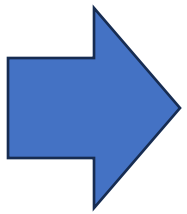
o3-mini-high implementation: https://chatgpt.com/share/67b027f9-fb28-8012-aa64-a1f7479134b7

Possible Implications

Do neural networks end up learning more efficient **symbolic representations** that we don't know?

Does gradient descent lead to a solution that can be reached by **advanced algebraic operations**?

Will gradient descent become **obsolete**, eventually?



Thanks!

Thanks!