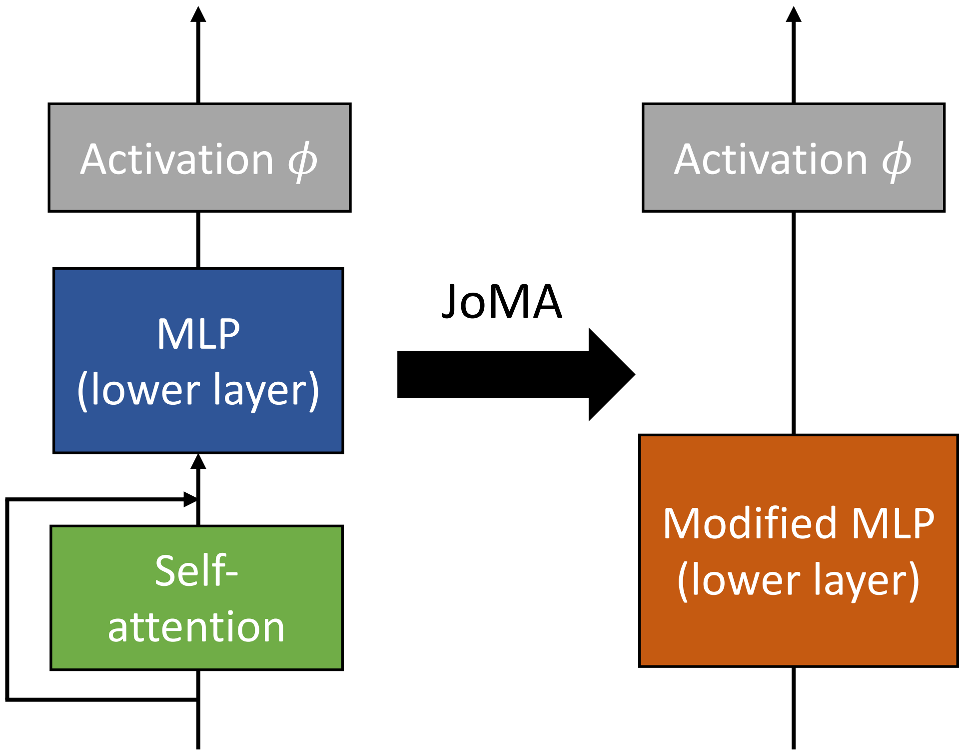
JoMA: Demystifying Multilayer Transformers via JOint Dynamics of MLP and Attention

Yuandong Tian[1], Yiping Wang[2], Zhenyu Zhang[3], Beidi Chen[1,4], Simon Du[2]

[1]Meta AI (FAIR)
[2]University of Washington
[3]University of Texas, Austin
[4]Carnegie Mellon University

Published in International Conference in Learning Representation (ICLR) 2024

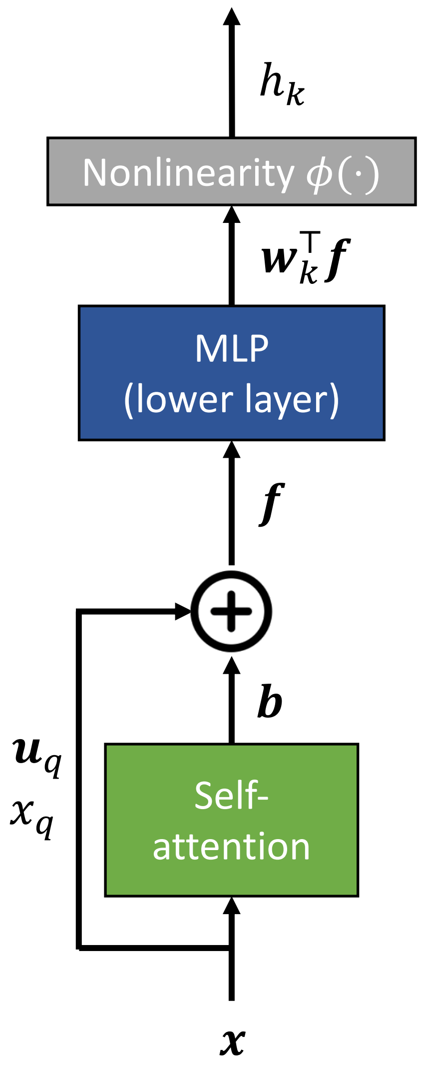
JoMA: JOint Dynamics of MLP/Attention layers



Main Contributions:

1. Find a joint dynamics that connects MLP with self-attention.
2. Understand self-attention behaviors for linear/nonlinear activations.
3. Explain how data hierarchy is learned in multi-layer Transformers.

JoMA Settings



"This is an apple"

$$h_k = \phi(\boldsymbol{w}_k^\top \boldsymbol{f})$$

$$\boldsymbol{f} = U_C \boldsymbol{b} + \boldsymbol{u}_q$$

U_C and $\boldsymbol{u}_q$ are embeddings

$$\boldsymbol{b} = \sigma(\boldsymbol{z}_q) \circ \boldsymbol{x}/A$$

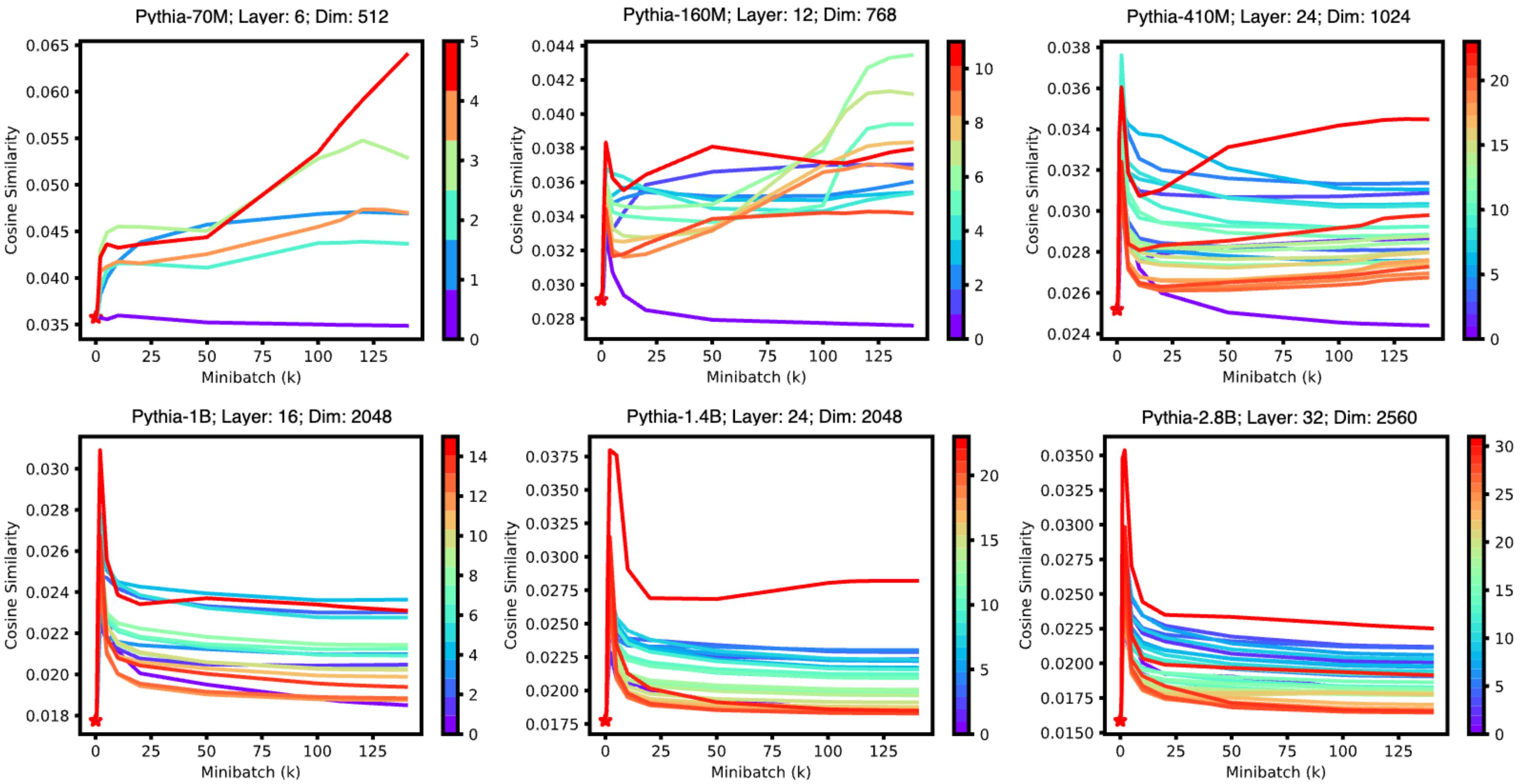
SoftmaxAttn: $b_l = \frac{x_l e^{z_{ql}}}{\sum_l x_l e^{z_{ql}}}$

ExpAttn: $b_l = x_l e^{z_{ql}}$

LinearAttn: $b_l = x_l z_{ql}$

Assumption (Orthogonal Embeddings $[U_C, u_q]$)

Cosine similarity between embedding vectors at different layers.



JoMA Dynamics

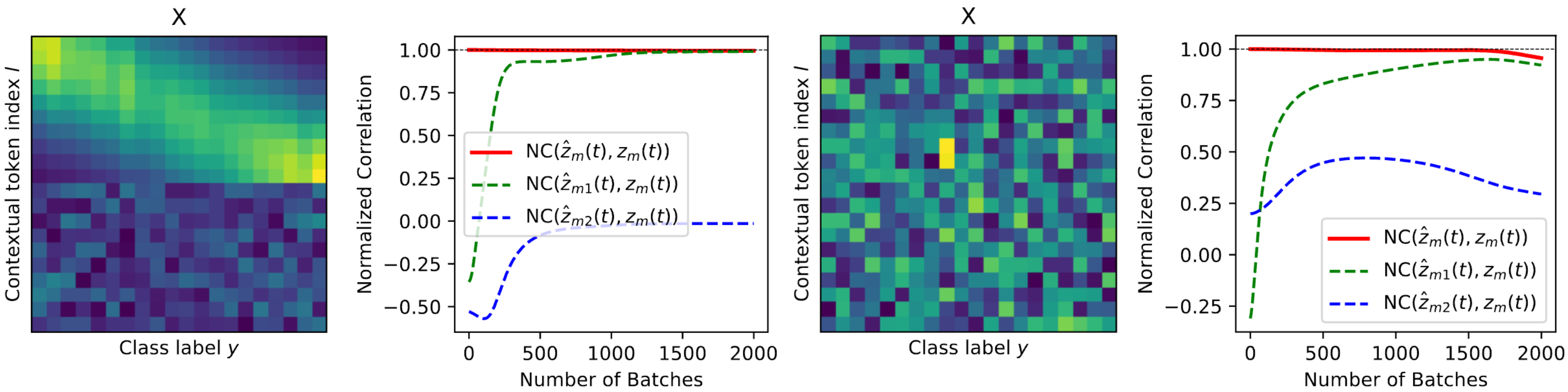
Theorem 1 (JoMA). *Let $\boldsymbol{v}_k := U_C^\top \boldsymbol{w}_k$, then the dynamics of Eqn. 3 satisfies the invariants:*

- *Linear attention. The dynamics satisfies $\boldsymbol{z}_m^2(t) = \sum_k \boldsymbol{v}_k^2(t) + \boldsymbol{c}$.*
- *Exp attention. The dynamics satisfies $\boldsymbol{z}_m(t) = \frac{1}{2}\sum_k \boldsymbol{v}_k^2(t) + \boldsymbol{c}$.*
- *Softmax attention. If $\bar{\boldsymbol{b}}_m := \mathbb{E}_{q=m}[\boldsymbol{b}]$ is a constant over time and $\mathbb{E}_{q=m}\left[\sum_k g_{h_k} h'_k \boldsymbol{b}\boldsymbol{b}^\top\right] = \bar{\boldsymbol{b}}_m \mathbb{E}_{q=m}\left[\sum_k g_{h_k} h'_k \boldsymbol{b}\right]$, then the dynamics satisfies $\boldsymbol{z}_m(t) = \frac{1}{2}\sum_k \boldsymbol{v}_k^2(t) - \|\boldsymbol{v}_k(t)\|_2^2 \bar{\boldsymbol{b}}_m + \boldsymbol{c}$.*

Under zero-initialization ($\boldsymbol{w}_k(0) = 0$, $\boldsymbol{z}_m(0) = 0$), then the time-independent constant $\boldsymbol{c} = 0$.

There is residual connection.
Joint dynamics works for any learning rates between self-attention and MLP layer.
No assumption on the data distribution.

Verification of JoMA dynamics



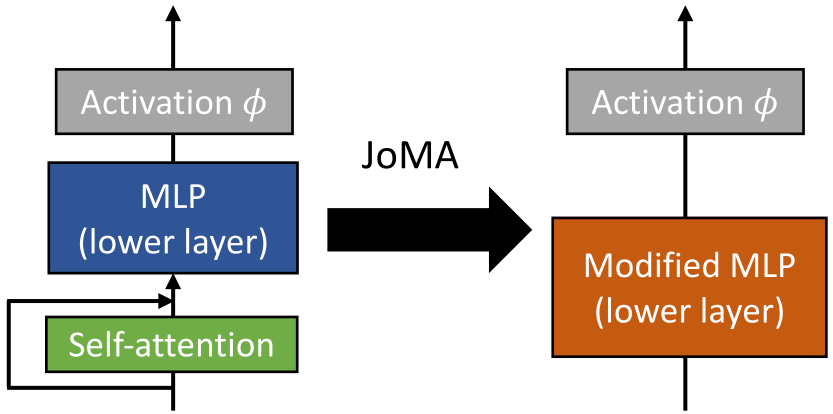
$\boldsymbol{z}_m(t)$: Real attention logits

$\hat{\boldsymbol{z}}_m(t)$: Estimated attention logits by JoMA

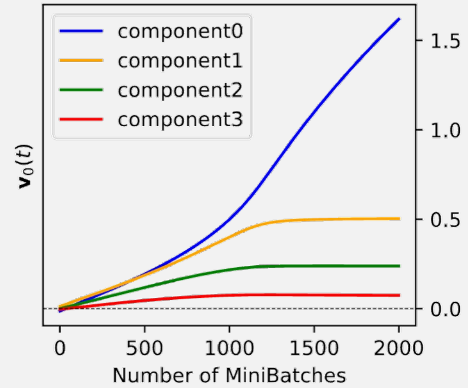
$$\hat{\boldsymbol{z}}_m(t) = \underbrace{\frac{1}{2}\sum_k \boldsymbol{v}_k^2(t)}_{\hat{\boldsymbol{z}}_{m1}(t)} - \underbrace{\|\boldsymbol{v}_k(t)\|_2^2\overline{\boldsymbol{b}}_m}_{\hat{\boldsymbol{z}}_{m2}(t)} + \boldsymbol{c}$$

Implication of Theorem 1

Key idea: folding self-attention into MLP
→ A Transformer block becomes a modified MLP

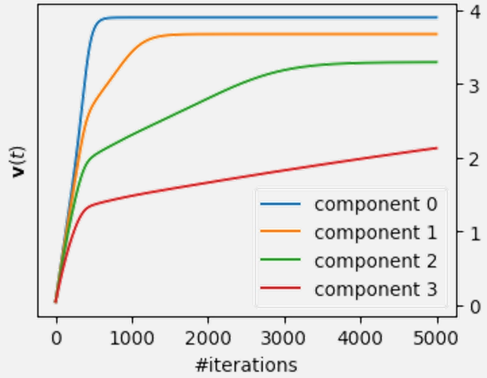


Linear case ($\phi = \mathrm{Id}, K = 1$)



Most salient feature takes all
(Attention becomes sparser)

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

facebook Artificial Intelligence

JoMA for Linear Activation

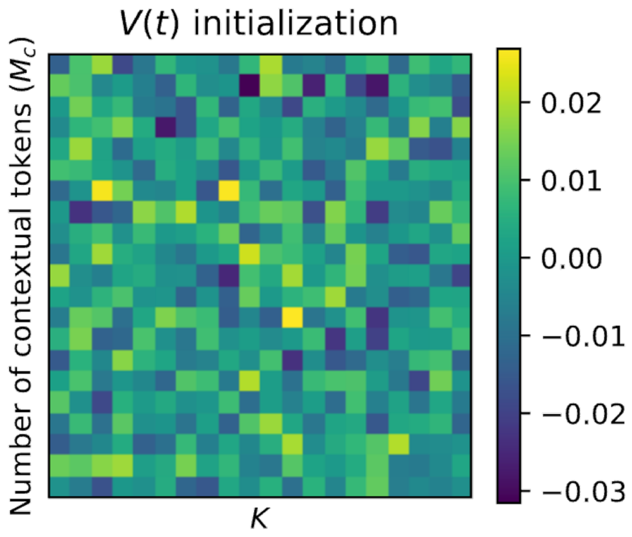
	Linear
$\dot{\boldsymbol{v}} = \boldsymbol{\Delta}_m \circ \exp\left(\frac{\boldsymbol{v}^2}{2}\right)$	Modified MLP (lower layer)

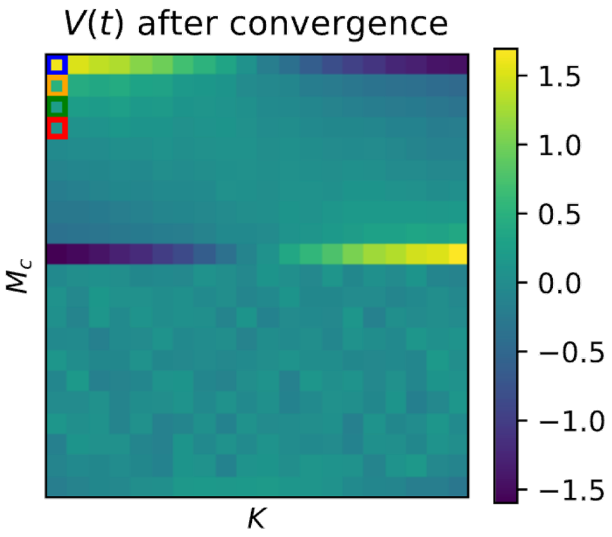
Theorem 2

We can prove $\frac{\mathrm{erf}(v_l(t)/2)}{\Delta_{lm}} = \frac{\mathrm{erf}(v_{l'}(t)/2)}{\Delta_{l'm}}$

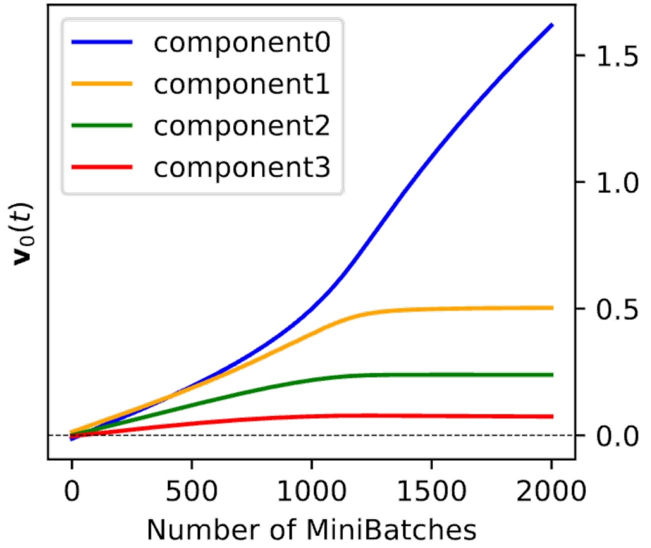
$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} \mathrm{d}t \in [-1,1]$$

Only the most salient token $l^* = \mathrm{argmax}\, |\Delta_{lm}|$ of $\boldsymbol{v}$ goes to $+\infty$
other components stay finite.





Attention becomes sparser
(Consistent with Scan&Snap)



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

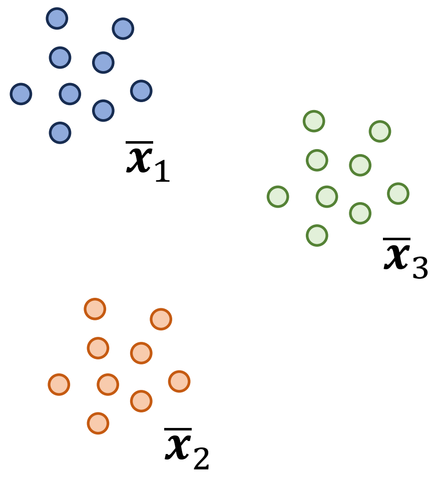
JoMA for Nonlinear Activation

Theorem 3

If $\boldsymbol{x}$ is sampled from a mixture of C isotropic distributions,
(i.e., “local salient/nonsalient map”), then

$$\dot{\boldsymbol{v}} = \frac{1}{\|\boldsymbol{v}\|_2}\sum_c a_c\theta_1(r_c)\bar{\boldsymbol{x}}_c + \frac{1}{\|\boldsymbol{v}\|_2^3}\sum_c a_c\theta_2(r_c)\boldsymbol{v}$$

Here $a_c \coloneqq \mathbb{E}_{q=m,c}\left[g_{h_k}\right]\mathbb{P}[c]$, $r_c = \boldsymbol{v}^\top\bar{\boldsymbol{x}}_c + \int_0^t \mathbb{E}_{q=m}\left[g_{h_k}h_k'\right]\mathrm{d}t$,
and θ_1 and θ_2 depends on nonlinearity



What does the dynamics look like?

$$\dot{\boldsymbol{v}} = (\boldsymbol{\mu} - \boldsymbol{v}) \circ \exp\left(\frac{\boldsymbol{v}^2}{2}\right)$$

$\boldsymbol{\mu} \sim \bar{\boldsymbol{x}}_c$: Critical point due to nonlinearity
(one of the cluster centers)

JoMA for Nonlinear activation

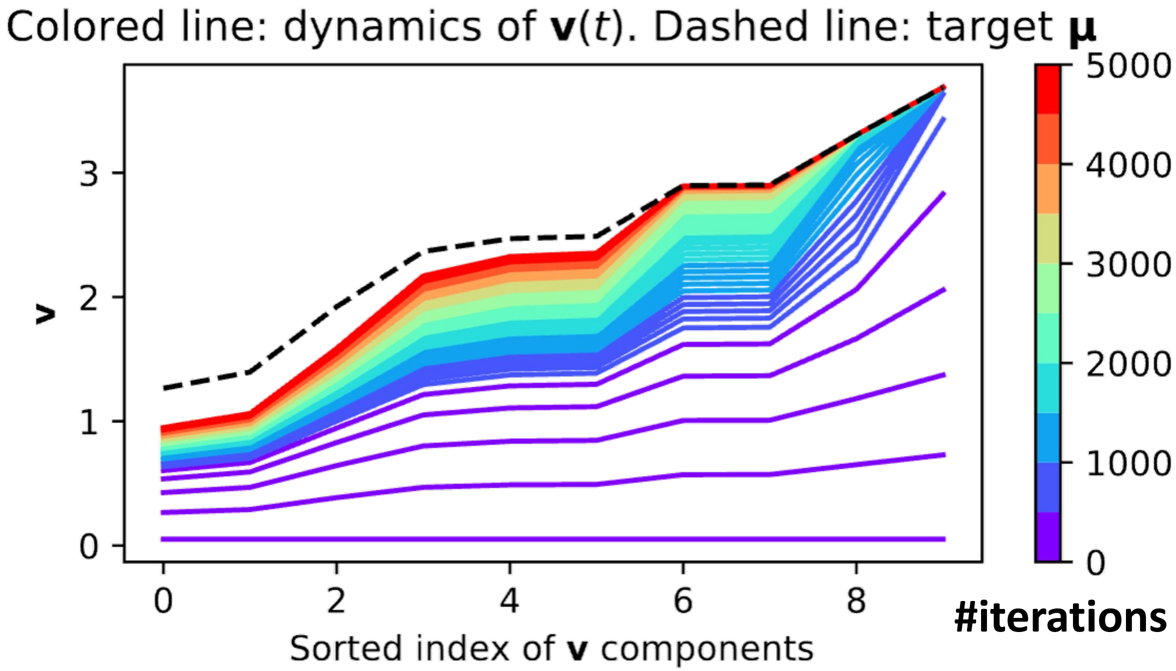
Nonlinear	
$\dot{\boldsymbol{v}} = (\boldsymbol{\mu} - \boldsymbol{v}) \circ \exp\left(\frac{\boldsymbol{v}^2}{2}\right)$	Modified MLP (lower layer)

Theorem 4

Salient components grow much faster than non-salient ones:

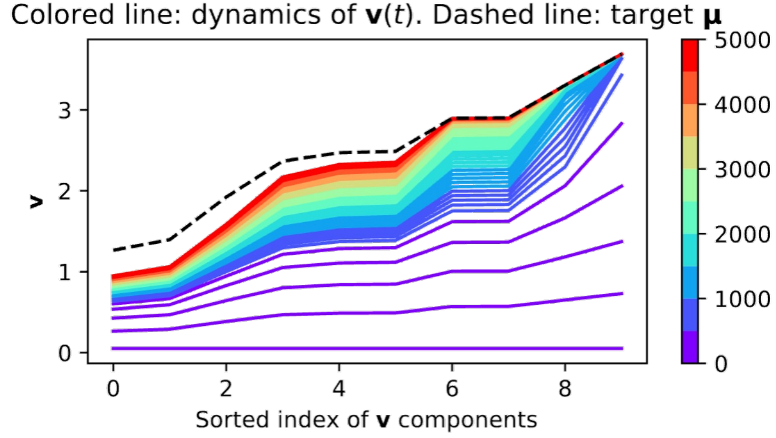
$$\frac{\mathrm{ConvergenceRate}(j)}{\mathrm{ConvergenceRate}(k)} \sim \frac{\exp(\mu_j^2/2)}{\exp(\mu_k^2/2)}$$

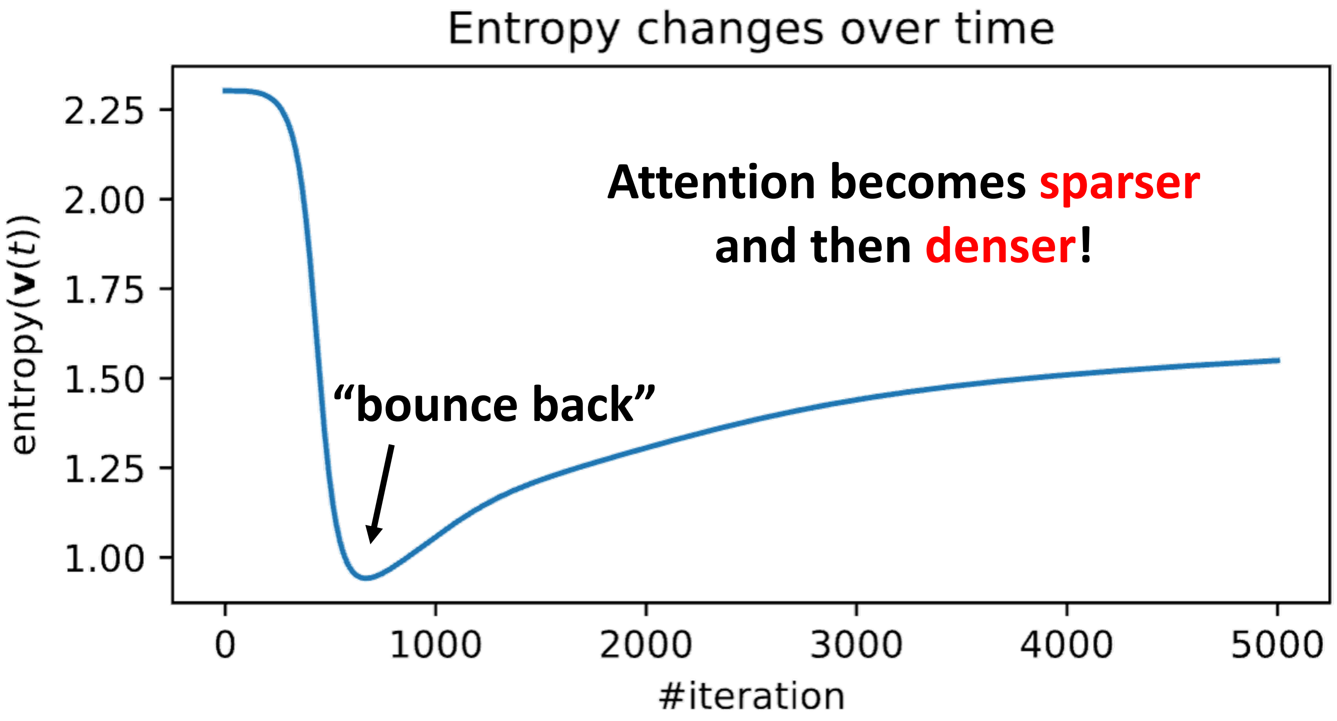
$$\mathrm{ConvergenceRate}(j) \coloneqq \ln 1/\delta_j(t)$$
$$\delta_j(t) \coloneqq 1 - v_j(t)/\mu_j$$



JoMA for Nonlinear activation

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

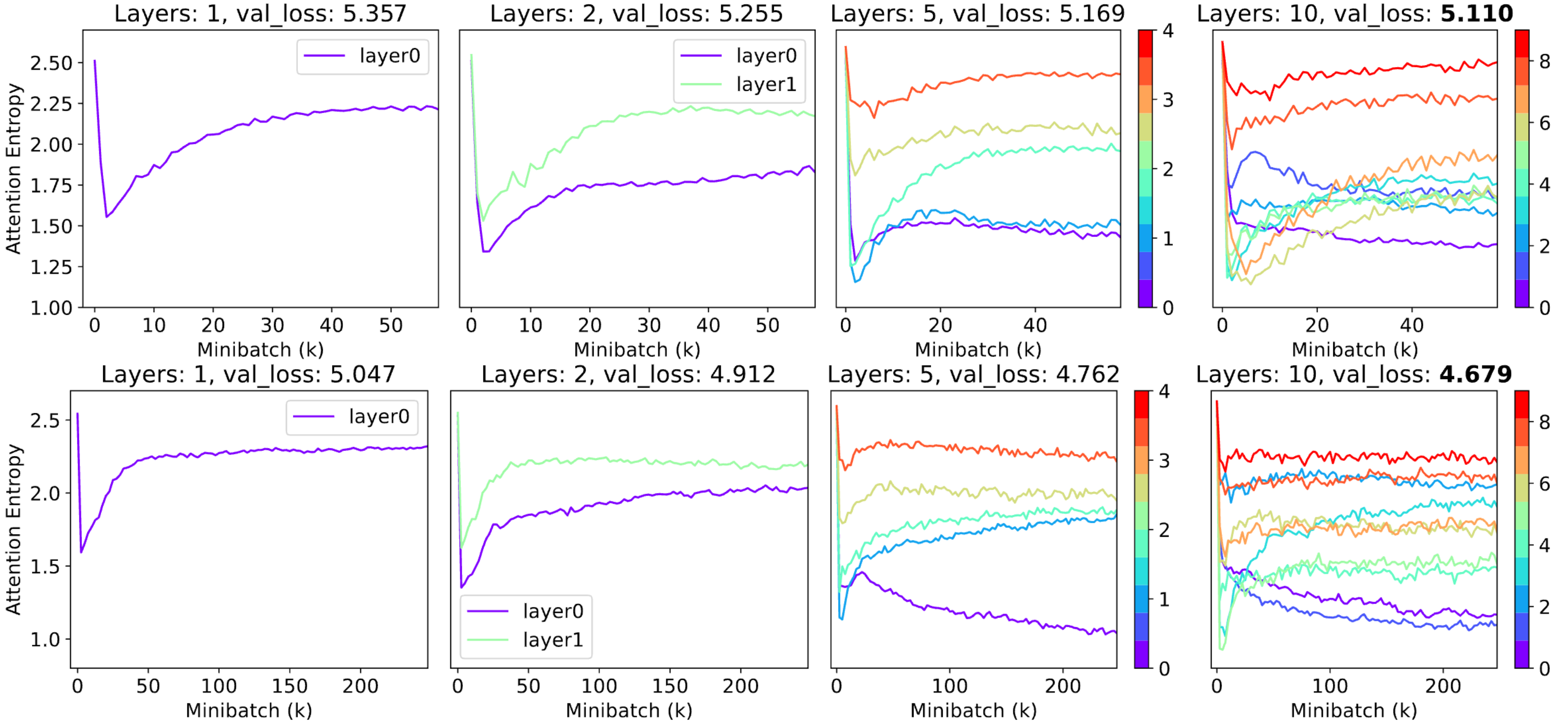




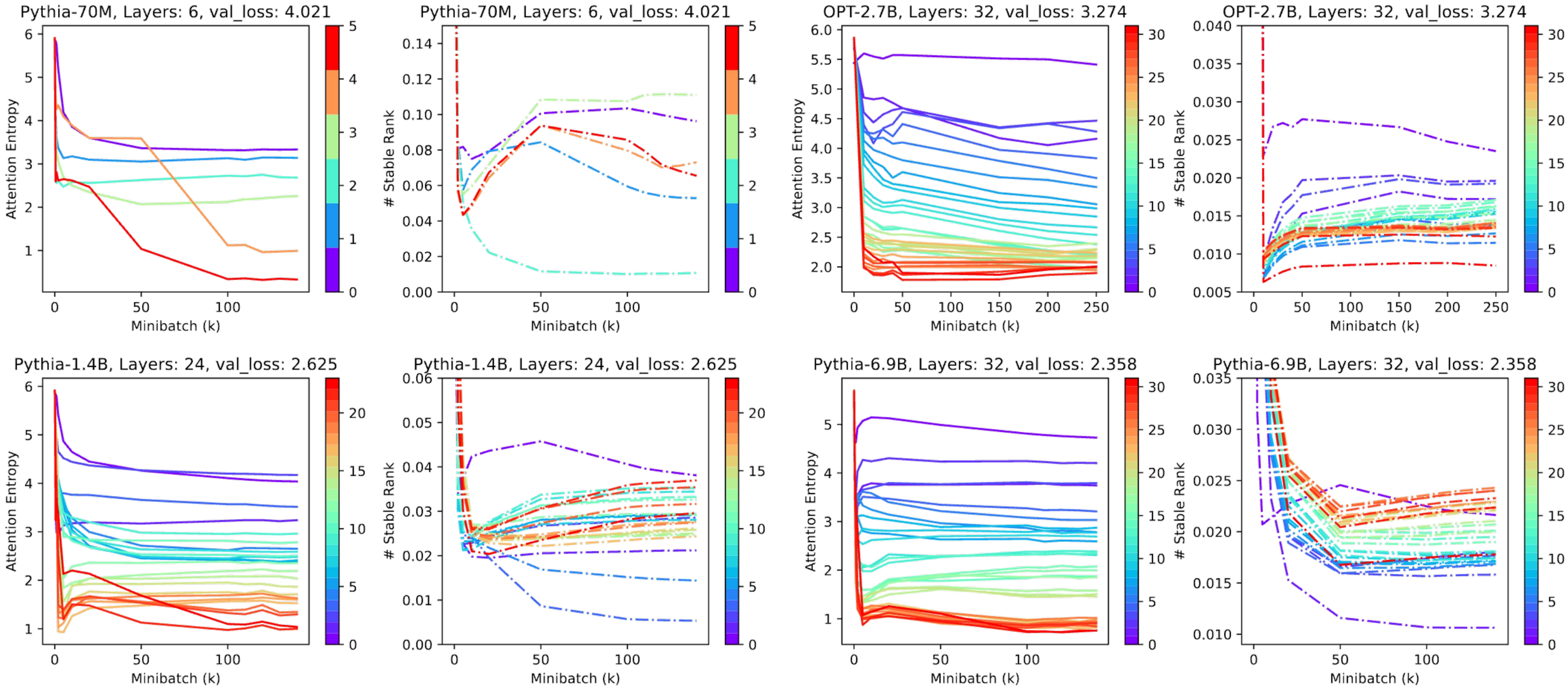
Real-world Experiments

Wikitext2

Wikitext103



Real-world Experiments



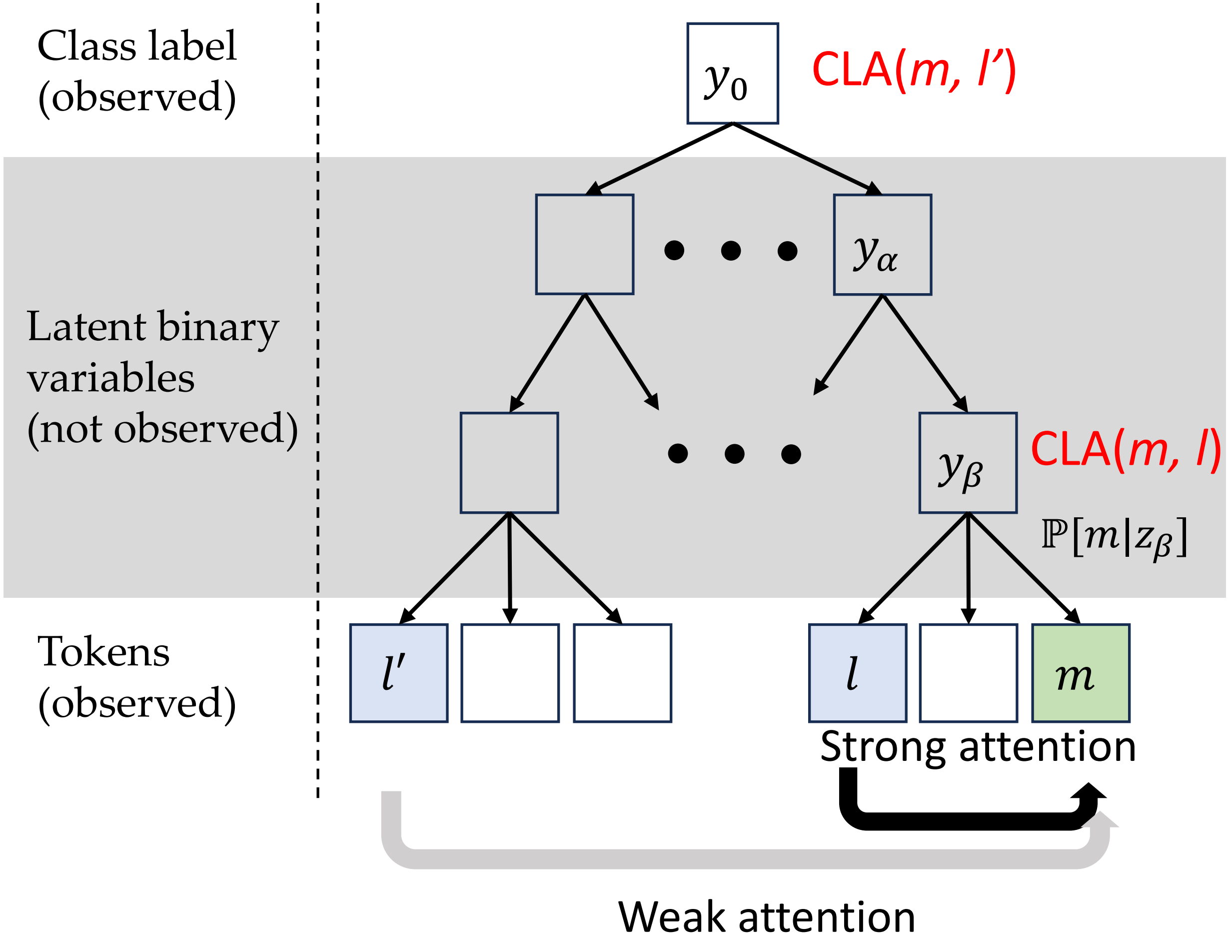
Stable Rank of the lower layer of MLP shows the “bouncing back” effects as well.

Why is this “bouncing back” property useful?

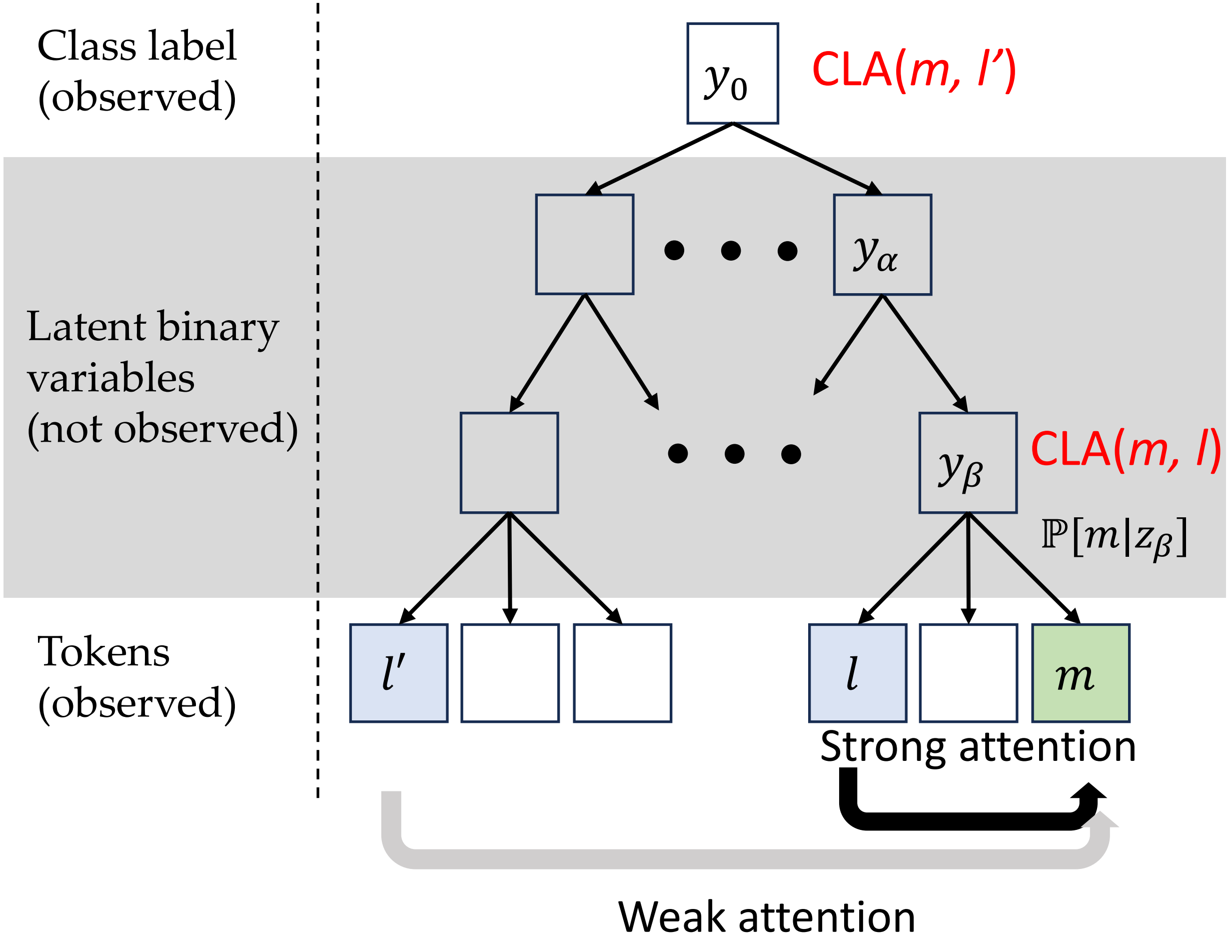
It seems that it only slows down the training??

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

Data Hierarchy & Multilayer Transformer



Data Hierarchy & Multilayer Transformer



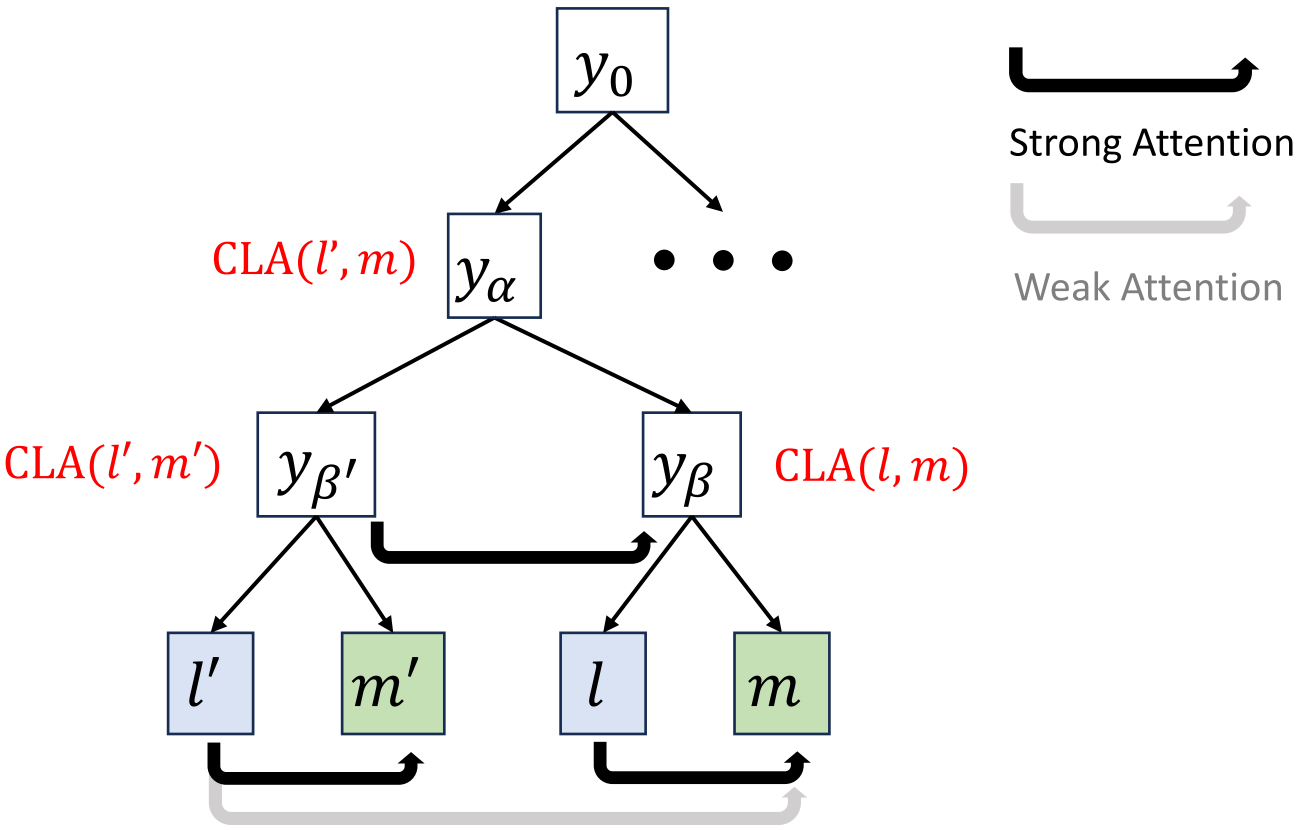
Theorem 5

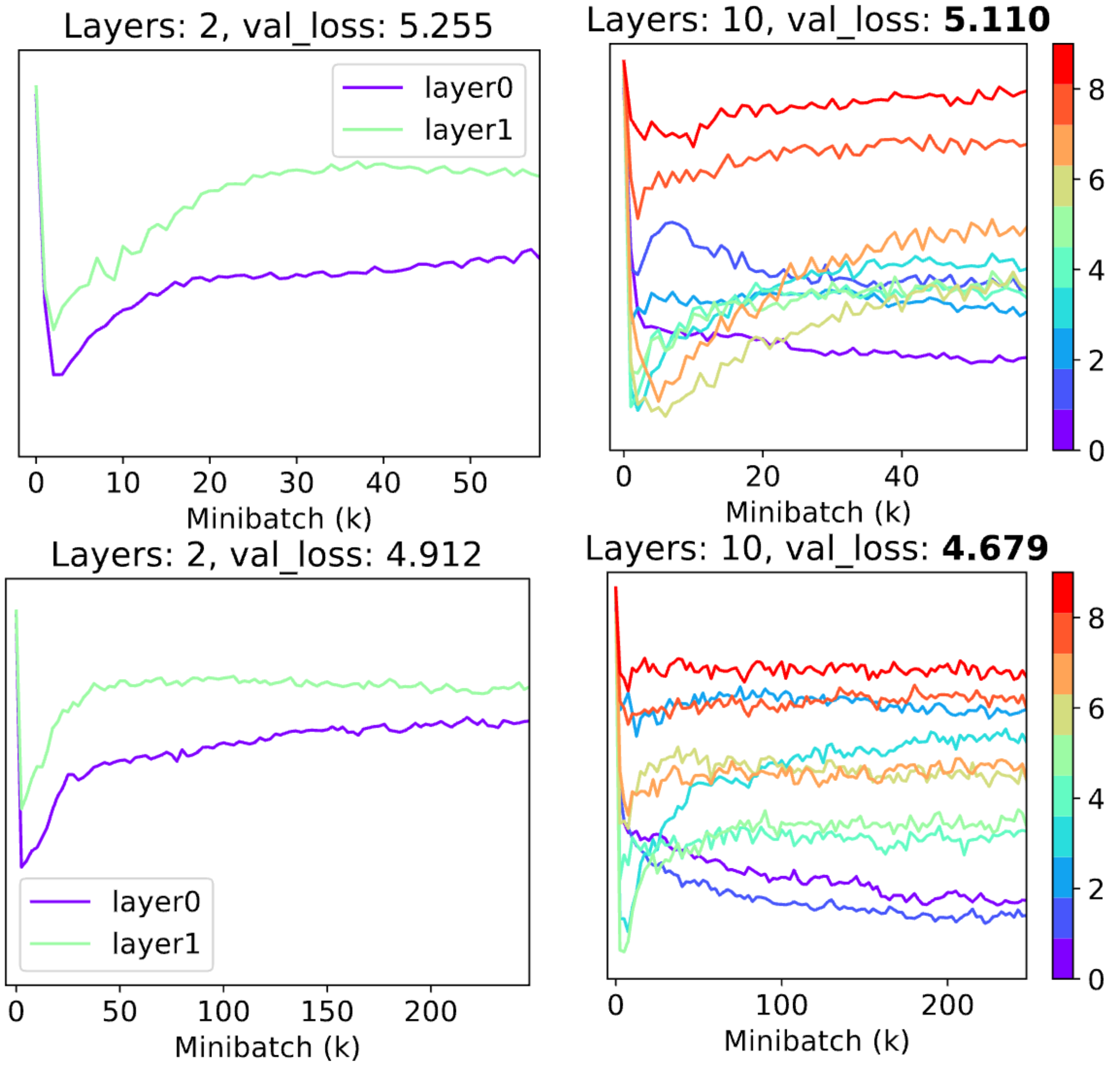
$$\mathbb{P}[l|m] \approx 1 - \frac{H}{L}$$

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

L: total height of the hierarchy

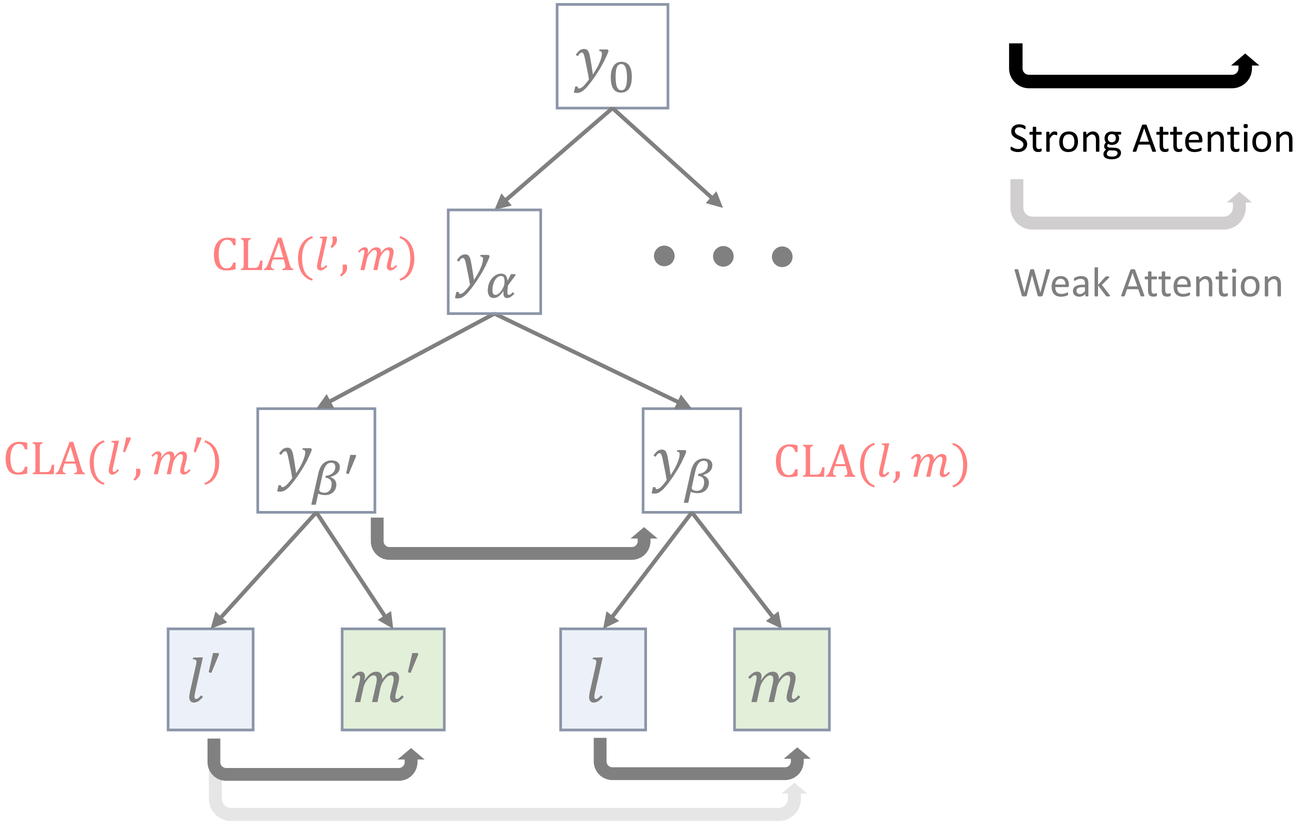
Deep Latent Distribution

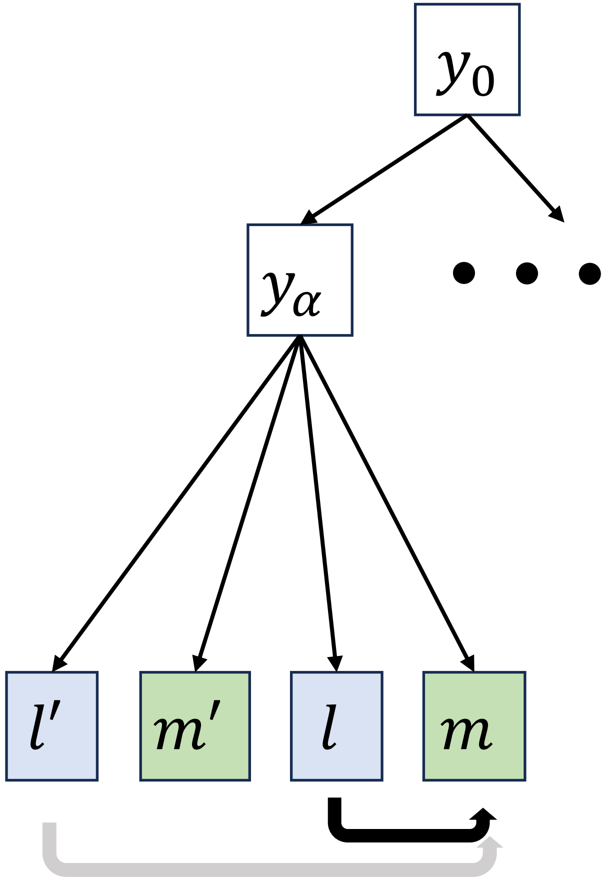




Learning the current hierarchical structure by
slowing down the association of tokens that are not directly correlated

Shallow Latent Distribution





Self-attention enables **Hierarchy-agnostic** Learning!

Future Work

- How embedding vectors are learned?
  - In both Scan&Snap and JoMA, we assume embeddings are constant.
- Positional Encoding
- Formulate the dynamics of Multi-layer Transformers
  - How intermediate latent concept gets learned during training?
  - Why we need over-parameterization?

Thanks!