Phase transitions between mechanisms in small models of transformers

A **sample complexity** and an **architectural perspective**

Hugo Cui

Luca Biggio

Florent Krzakala

Lenka Zdeborová

Freya Behrens, **SPOC** group, EPFL

Cargèse 15.08.2025

```
Three months after November is [prompt]
```

Not All Language Model Features Are Linear [Engels et al 2024]

`Three months after November is [prompt]`   99 % accuracy   **Llama 3 8B**

Not All Language Model Features Are Linear [Engels et al 2024]

```
Three months after November is [prompt]
(3 + 11) % 12 = [prompt]
```

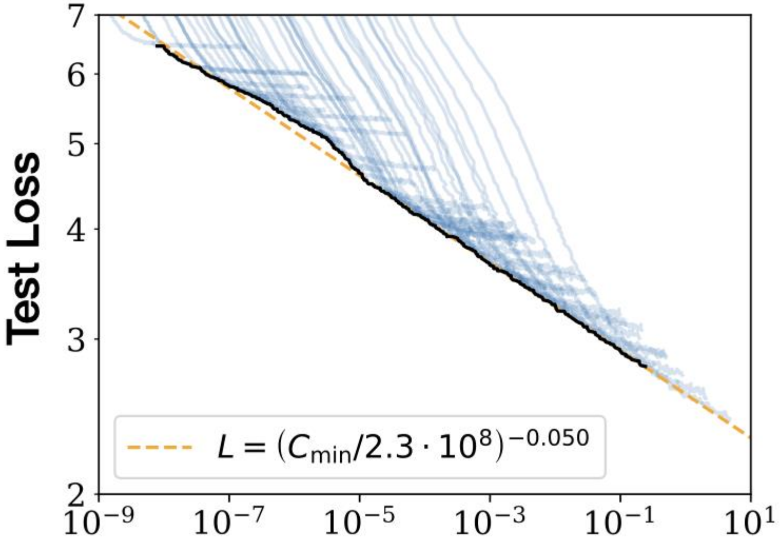
99 % accuracy **Llama 3 8B**

Not All Language Model Features Are Linear [Engels et al 2024]

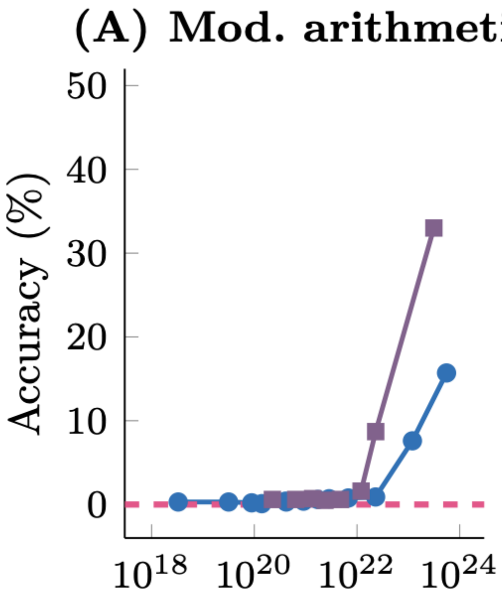
`Three months after November is [prompt]` 99 % accuracy **Llama 3 8B**

`(3 + 11) % 12 = [prompt]` ~8 % accuracy

Not All Language Model Features Are Linear [Engels et al 2024]



Model Scale



Model Scale

Scaling Laws for Neural Language Models
[Kaplan et al ’22]

Emergent Abilities of Large Language Models
[Wei et al ’22]

LLMs exhibit as many failure modes as capabilities.

LLMs exhibit as many failure modes as capabilities.

How do they fail? (When do their capabilities emerge?)

LLMs exhibit as many failure modes as capabilities.

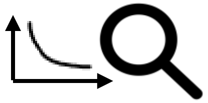
How do they fail? (When do their capabilities emerge?)

- Architecture: Capacity too small?
- Training: Memorizing the data instead of generalizing?
- Data: Too few samples available to generalize?

LLMs exhibit as many failure modes as capabilities.

How do they fail? (When do their capabilities emerge?)

- Architecture: Capacity too small?
- Training: Memorizing the data instead of generalizing?
- Data: Too few samples available to generalize?



What is the performance of the learned model?

How is the performance influenced by external factors?

LLMs exhibit as many failure modes as capabilities.

LLMs exhibit as many failure modes as capabilities.

Debug: Inspect and understand model internals

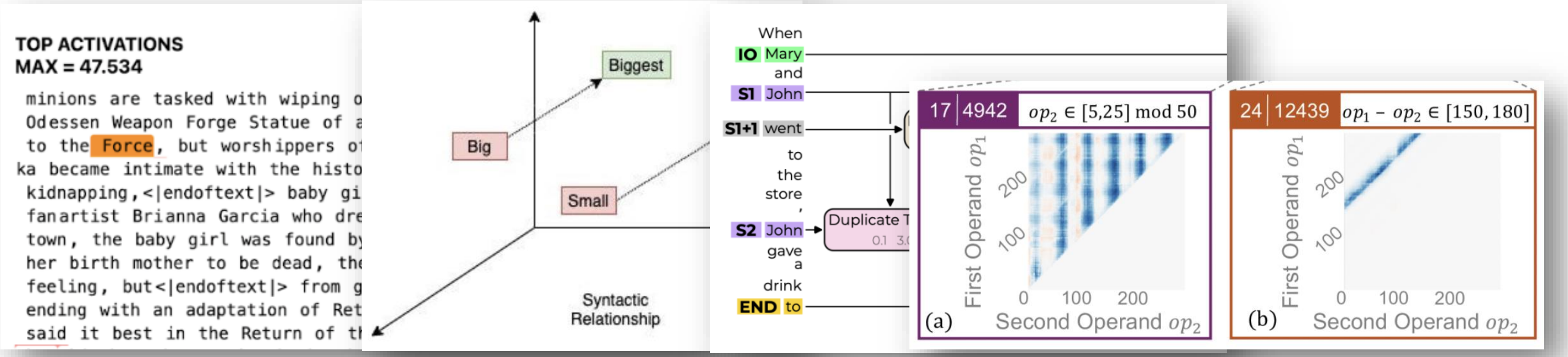
Which features or mechanisms did the model learn?

LLMs exhibit as many failure modes as capabilities.

Debug: Inspect and understand model internals

Which features or mechanisms did the model learn?

Examples:



[Bricken et al ‘23] [Miklov et al ‘15] [Wang et al ‘22] [Nitakin et al ‘24]

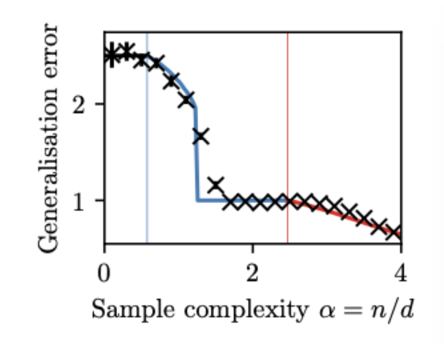
LLMs exhibit as many failure modes as capabilities.

To fix pre-training: How does this depend on external factors?

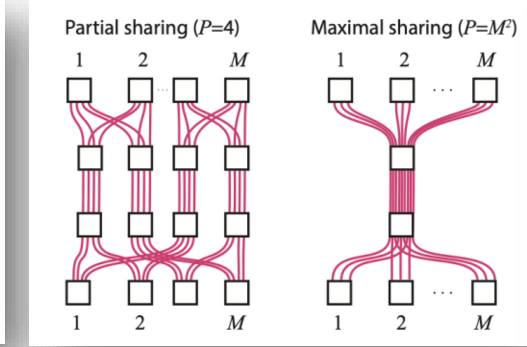
LLMs exhibit as many failure modes as capabilities.

To fix pre-training: How does this depend on external factors?

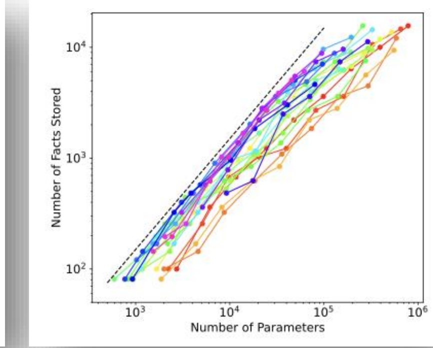
Examples:



[Troiani et al '24]



[Saxe et al '22]



[Nichani et al '24]

Part 1 : A Phase Transition Between Semantic and Positional Learning

arXiv:2402.03902 – Hugo Cui, Freya Behrens, Florent Krzakala, Lenka Zdeborová

How is the learned algorithm determined by the sample complexity?
Do different algorithms emerge spontaneously?

Part 2 : The Interplay between Attention and Feed-Forward Layers

arXiv:2407.11542 – Freya Behrens, Luca Biggio, Lenka Zdeborová

How is the learned algorithm determined by architectural choices?
Which functions are executed by which parts?

Part 1 : A Phase Transition Between Semantic and Positional Learning

arXiv:2402.03902 – Hugo Cui, Freya Behrens, Florent Krzakala, Lenka Zdeborová

How is the learned algorithm determined by the sample complexity?
Do different algorithms emerge spontaneously?

Part 2 : The Interplay between Attention and Feed-Forward Layers

arXiv:2407.11542 – Freya Behrens, Luca Biggio, Lenka Zdeborová

How is the learned algorithm determined by architectural choices?
Which functions are executed by which parts?

Algorithms use the information encoded in a sentence ….

We analyze a phase transition between positional and semantic meaning

Algorithms use the information encoded in a sentence ….

We analyze a phase transition between positional and semantic meaning

In the **meaning** of the tokens *(semantics)*

We sanitize a face ambition between rational and acrylic baking

Algorithms use the information encoded in a sentence ….

We analyze a phase transition between positional and semantic meaning

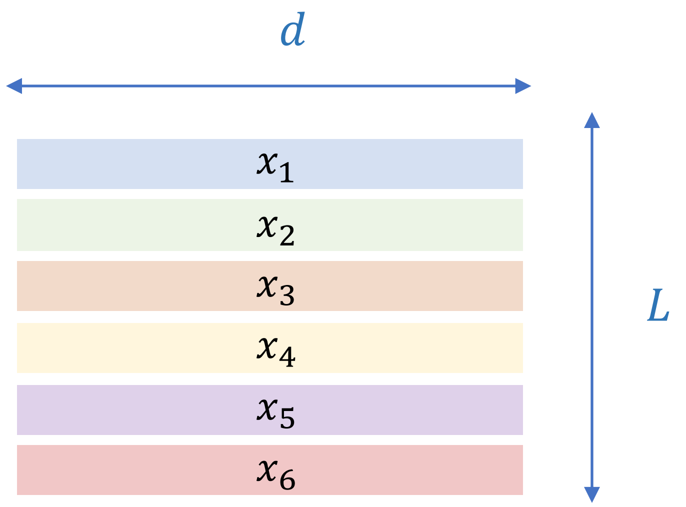
In the **meaning** of the tokens *(semantics)*

We sanitize a face ambition between rational and acrylic baking

And their **ordering** in the sentence *(positions)*

A between a phase semantic learning and positional analyze transition

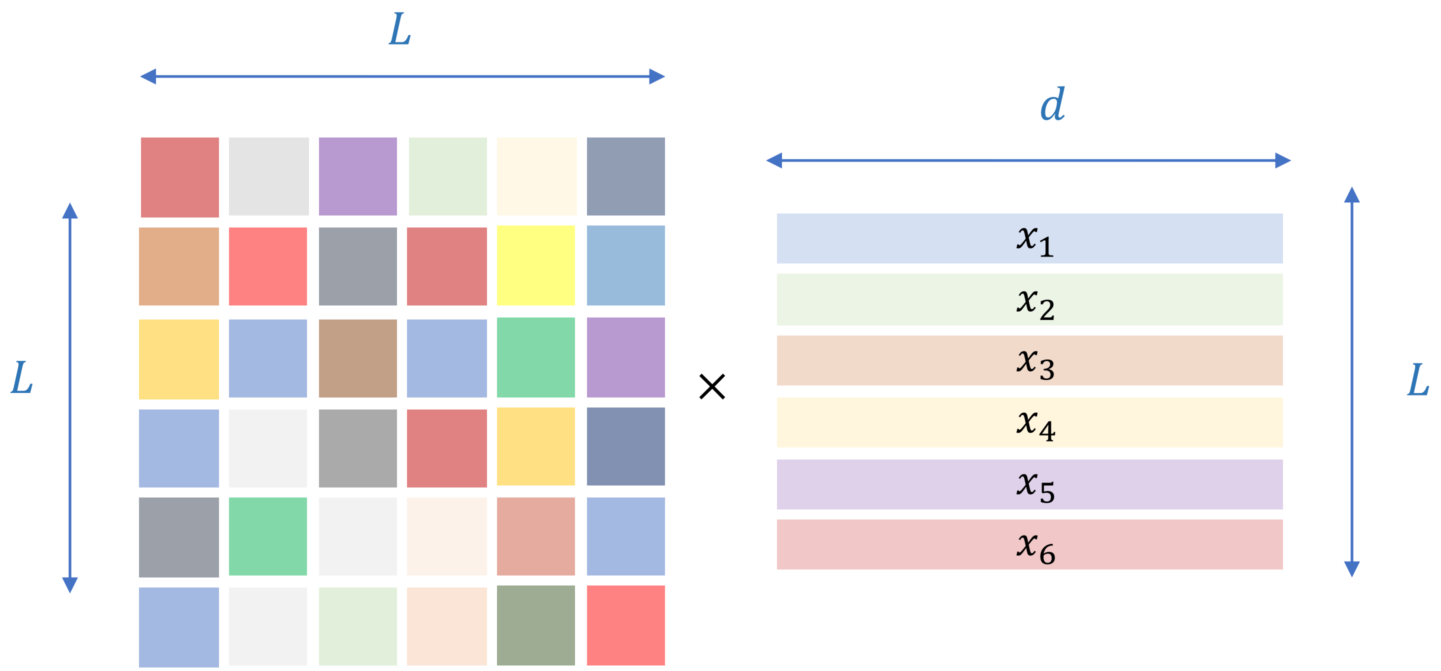
Input sentence
(embedded)



$\boldsymbol{x} \in \mathbb{R}^{L \times d}$

Attention matrix:
mixes tokens together with a matrix

Input sentence
(embedded)



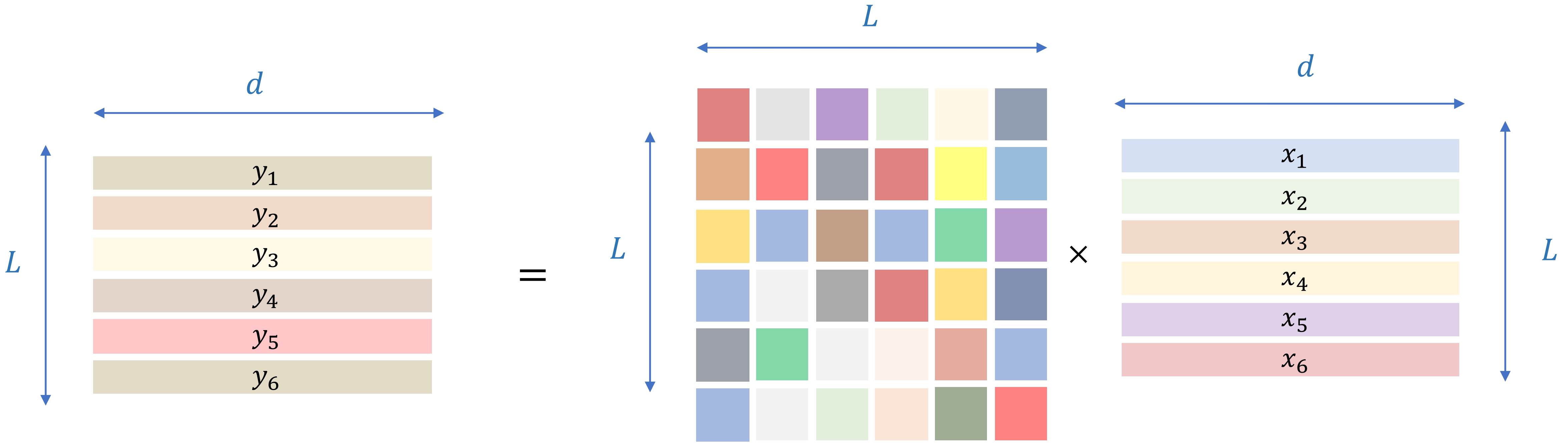
$\boldsymbol{S}(\boldsymbol{x}) \in \mathbb{R}^{L \times L}$

$\boldsymbol{x} \in \mathbb{R}^{L \times d}$

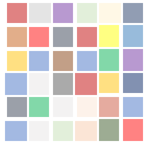
Context vector: fed to a feed-forward architecture for further feature extraction

Attention matrix: mixes tokens together with a matrix

Input sentence (embedded)

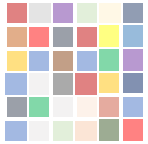


$\boldsymbol{y} \in \mathbb{R}^{L \times d}$ $\quad$ $\boldsymbol{S}(\boldsymbol{x}) \in \mathbb{R}^{L \times L}$ $\quad$ $\boldsymbol{x} \in \mathbb{R}^{L \times d}$



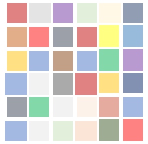
$S(x)_{ij} = S(x_i, x_j, i, j)$

Dependence on the **positions** i, j and the **tokens**, $x_i x_j$



$S(x)_{ij} = S(x_i, x_j, i, j)$ Dependence on the **positions** i, j and the **tokens**, $x_i x_j$

$S(x)_{ij} = S(x_i, x_j, i, j)$ *Purely semantic attention mechanism*



$S(x)_{ij} = S(x_i, x_j, i, j)$ Dependence on the **positions** i, j and the **tokens**, $x_i x_j$

$S(x)_{ij} = S(x_i, x_j, i, j)$ *Purely semantic attention mechanism*

$S(x)_{ij} = S(x_i, x_j, i, j)$ *Purely positional attention mechanism*

$S(x)_{ij} = S(x_i, x_j, i, j)$ Dependence on the **positions** i, j and the **tokens**, $x_i x_j$

$S(x)_{ij} = S(x_i, x_j, i, j)$ *Purely semantic attention mechanism*

$S(x)_{ij} = S(x_i, x_j, i, j)$ *Purely positional attention mechanism*

When does attention learn to implement positional/semantic mechanisms?

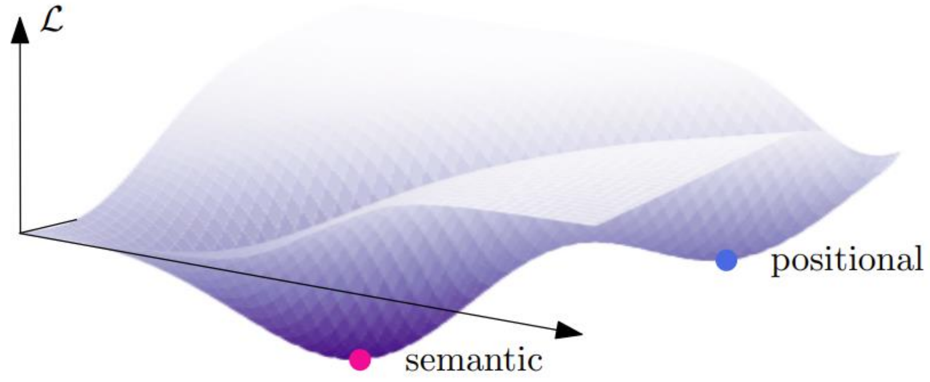
Histogram task : for each token, output the number of identical tokens in the sequence

input $x = (a, b, b, c, c, a, c, c, b)$

Histogram task : for each token, output the number of identical tokens in the sequence

input $x = (a, b, b, c, c, a, c, c, b)$

target $y = (2, 3, 3, 4, 4, 2, 4\ 4, 3)$



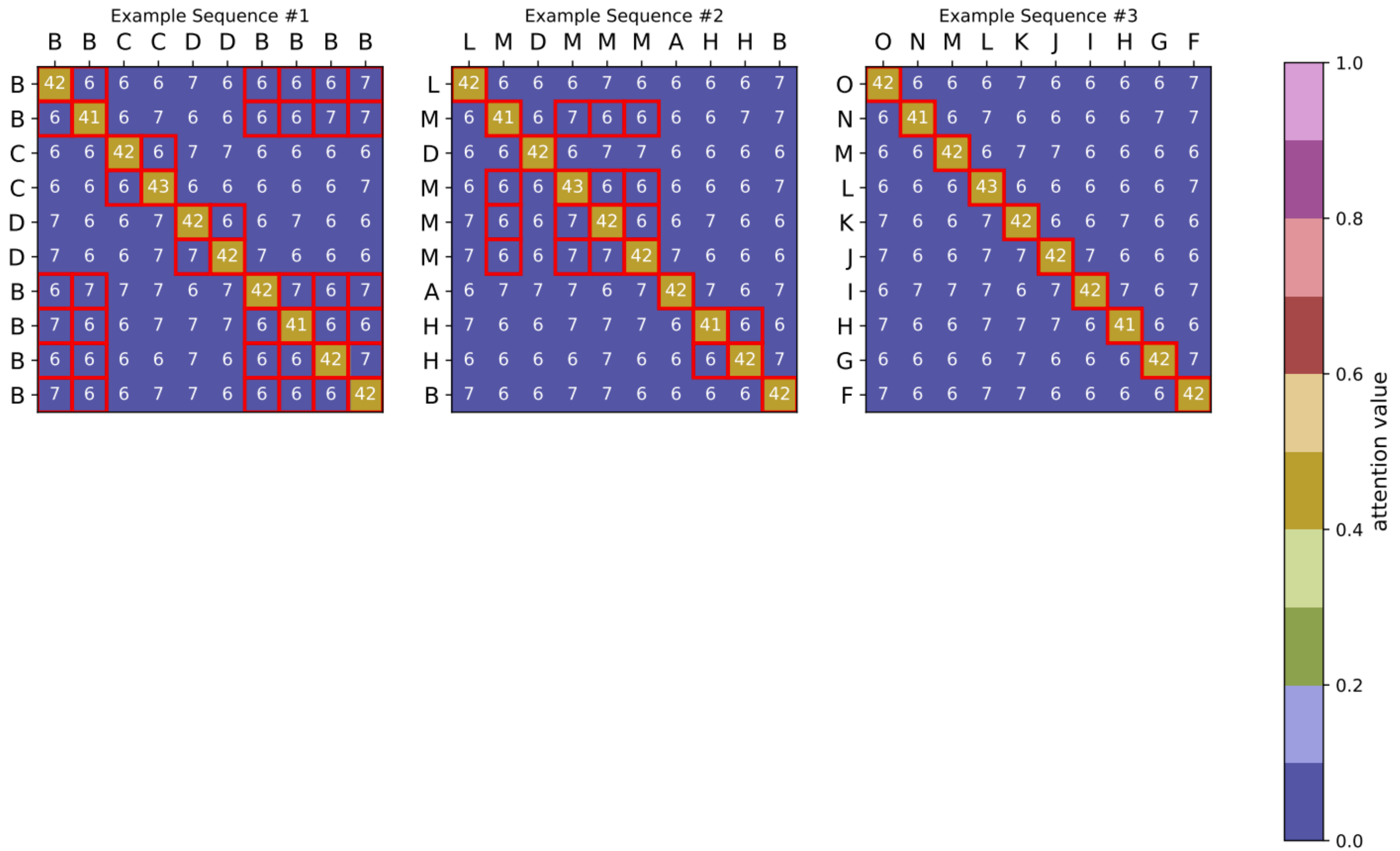
Histogram task : for each token, output the number of identical tokens in the sequence

input $x = (a, b, b, c, c, a, c, c, b)$

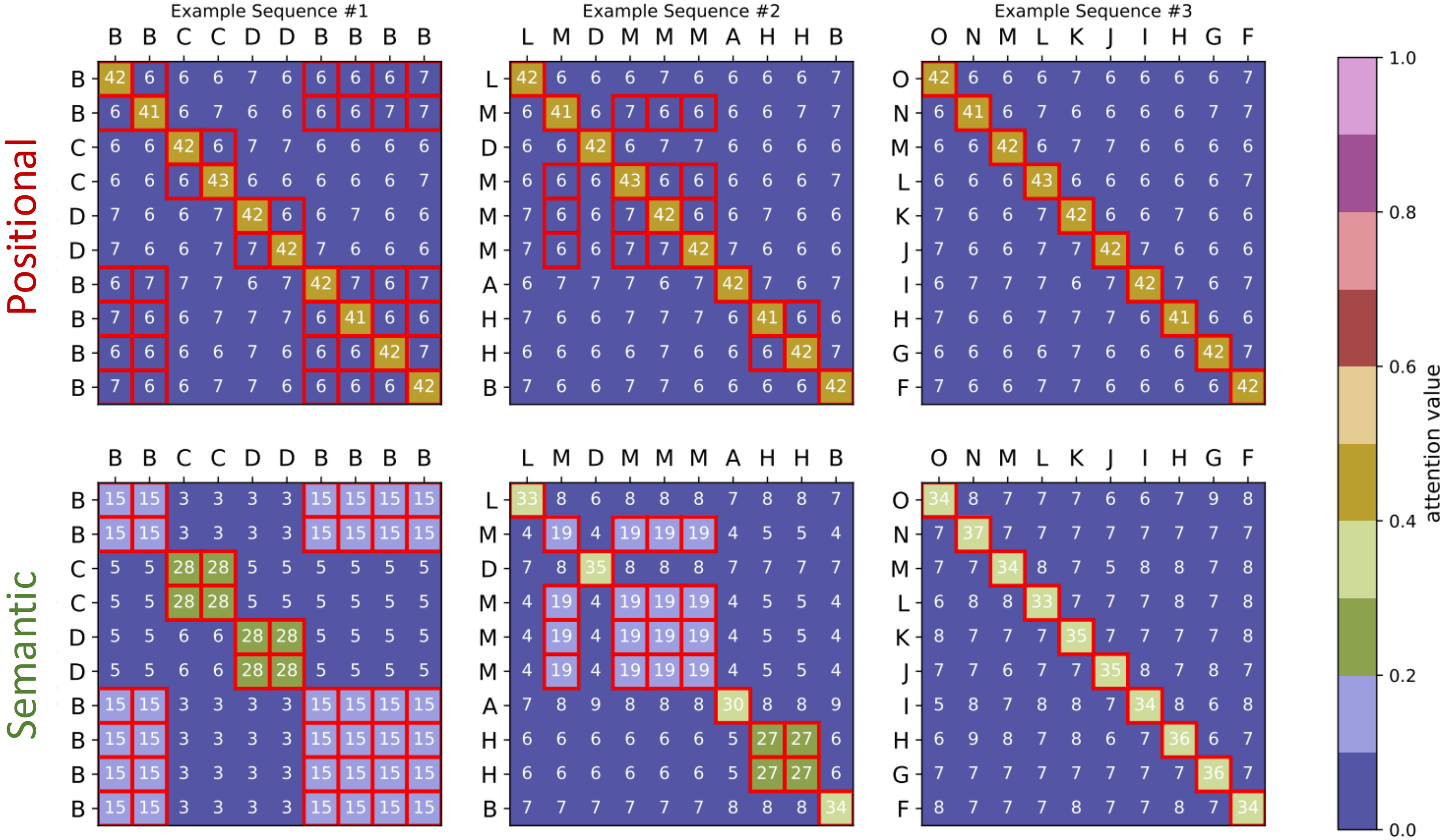
target $y = (2, 3, 3, 4, 4, 2, 4\ 4, 3)$

With a 1-layer transformer, we can reach two (almost) zero-gradient configurations with different behaviors.

Positional



Example Sequence #1
Example Sequence #2
Example Sequence #3
attention value


Example Sequence #1
Example Sequence #2
Example Sequence #3
Positional
Semantic
B B C C D D B B B B
L M D M M M A H H B
O N M L K J I H G F
attention value
0.0
0.2
0.4
0.6
0.8
1.0

A solvable model

A solvable model

Goal:
For a given task
for a given architecture
characterize the different minima
in an empirical loss landscape
as the sample complexity changes.

A phase transition?

A solvable model

Goal:
For a given task
for a given architecture
characterize the different minima
in an empirical loss landscape
as the sample complexity changes.

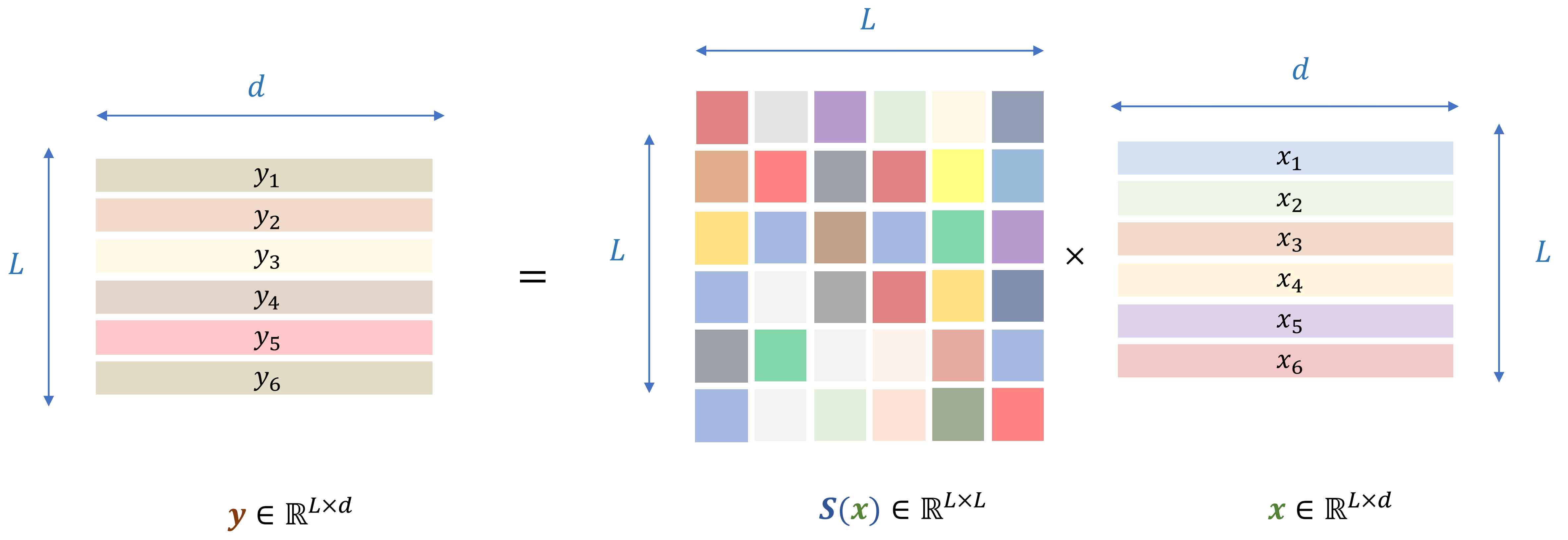
A phase transition?

(static!)

Context vector:
fed to a feed-forward architecture for further feature extraction

Attention matrix:
mixes tokens together with a matrix

Input sentence
(embedded)



Data model

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_L \end{pmatrix} \in \mathbb{R}^{L \times d}$$

With the ℓ-th token $x_\ell \sim \mathcal{N}(0, \Sigma_\ell) \in \mathbb{R}^d$

Data model

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_L \end{pmatrix} \in \mathbb{R}^{L \times d}$$

With the ℓ-th token $x_\ell \sim \mathcal{N}(0, \Sigma_\ell) \in \mathbb{R}^d$

Target

$$y(x) = \left[(\mathbf{1} - \boldsymbol{\omega}) \, \mathrm{softmax}\left(\frac{x\, Q_* Q_*^\top x^\top}{d} \right) + \boldsymbol{\omega} A \right] \cdot x$$

Target attention

with $A \in \mathbb{R}^{L \times L}$, $Q_* \in \mathbb{R}^d$

Data model

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_L \end{pmatrix} \in \mathbb{R}^{L \times d}$$

With the ℓ-th token $x_\ell \sim \mathcal{N}(0, \Sigma_\ell) \in \mathbb{R}^d$

Target

$$y(x) = \left[(\mathbf{1} - \boldsymbol{\omega}) \, \mathrm{softmax}\left(\frac{x\, Q_* Q_*^\top x^\top}{d} \right) + \boldsymbol{\omega} A \right] \cdot x$$

Target attention

with $A \in \mathbb{R}^{L \times L}$, $Q_* \in \mathbb{R}^d$

$\boldsymbol{\omega} = 0$ Target attention is purely *semantic*

$\boldsymbol{\omega} = \mathbf{1}$ Target attention is purely *positional*

Student

$$f_Q(x) = \text{softmax}\left(\frac{(x+p)QQ^\top (x+p)^\top}{d}\right) \cdot x \qquad , Q \in \mathbb{R}^d$$

$p \in \mathbb{R}^{L \times d}$ are positional encodings. In the following for $L = 2$, $p = \begin{pmatrix} \mu \\ -\mu \end{pmatrix}$

Student

$$f_Q(x) = \mathrm{softmax}\left(\frac{(x+p)QQ^\top (x+p)^\top}{d}\right)\cdot x \qquad , Q \in \mathbb{R}^d$$

$p \in \mathbb{R}^{L\times d}$ are positional encodings. In the following for $L = 2$, $p = \begin{pmatrix}\mu \\ -\mu\end{pmatrix}$

ERM

$$\hat{Q} = \mathrm{argmin}_{\mathrm{Q}} \sum_{\mu=1}^{n} \left\| y(x^\mu) - f_Q(x^\mu) \right\|^2 + \frac{\lambda}{2}\|Q\|^2$$

Student

$$f_Q(x) = \text{softmax}\left(\frac{(x+p)QQ^\top (x+p)^\top}{d}\right) \cdot x \qquad , Q \in \mathbb{R}^d$$

$p \in \mathbb{R}^{L\times d}$ are positional encodings. In the following for $L = 2$, $\quad p = \begin{pmatrix} \mu \\ -\mu \end{pmatrix}$

ERM

$$\hat{Q} = \text{argmin}_{\text{Q}} \sum_{\mu=1}^{n} \left\| y(x^\mu) - f_Q(x^\mu) \right\|^2 + \frac{\lambda}{2}\|Q\|^2$$

Asymptotic limit:

$d, n \to \infty, \qquad \|p\|, \alpha = {}^{n}/_{d} = \Theta_d(1)$

$$\boldsymbol{m} = \frac{\mu^\top Q}{\sqrt{d}}, \boldsymbol{\theta} = \frac{Q_*^\top Q}{d}$$

$$\boldsymbol{m} = \frac{\mu^\top Q}{\sqrt{d}}, \boldsymbol{\theta} = \frac{Q_*^\top Q}{d}$$

We find **two** minima:

- $\boldsymbol{m} > 0\,, \boldsymbol{\theta} = 0$

the elements are partly independent of x :
positional mechanism

$$\boldsymbol{m} = \frac{\mu^\top Q}{\sqrt{d}}, \boldsymbol{\theta} = \frac{Q_*^\top Q}{d}$$

We find **two** minima:

- $\boldsymbol{m} > 0 \,, \boldsymbol{\theta} = 0$

  the elements are partly independent of x :
  positional mechanism

- $\boldsymbol{m} = 0, \boldsymbol{\theta} > 0$

  the elements depend on x :
  semantic mechanism

$$\boldsymbol{m} = \frac{\mu^\top Q}{\sqrt{d}}, \boldsymbol{\theta} = \frac{Q_*^\top Q}{d}$$

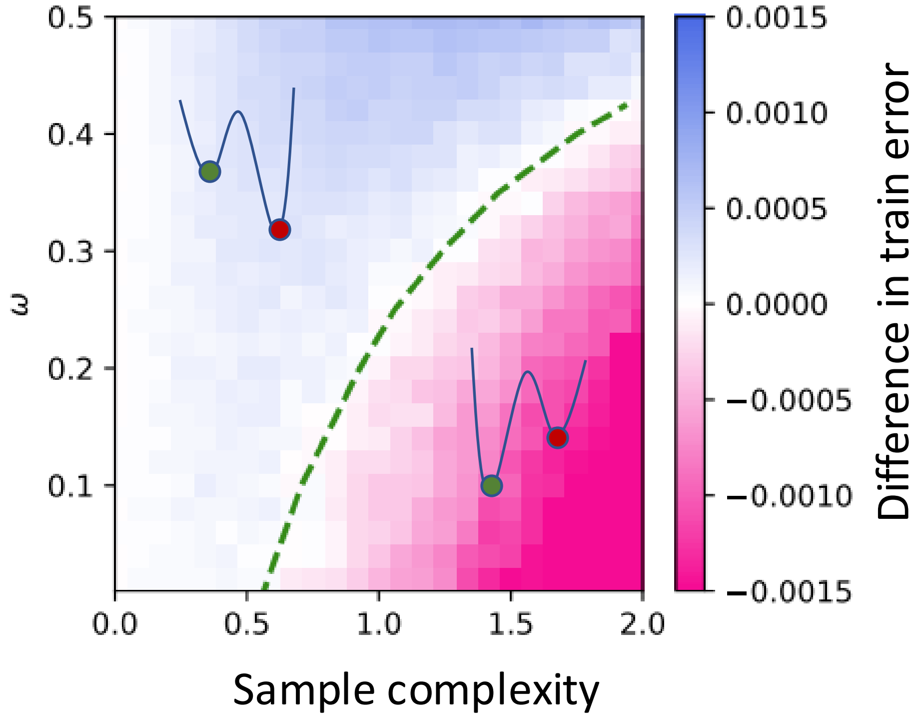
We find **two** minima:

- $\boldsymbol{m} > 0\,, \boldsymbol{\theta} = 0$

  the elements are partly independent of x :
  positional mechanism

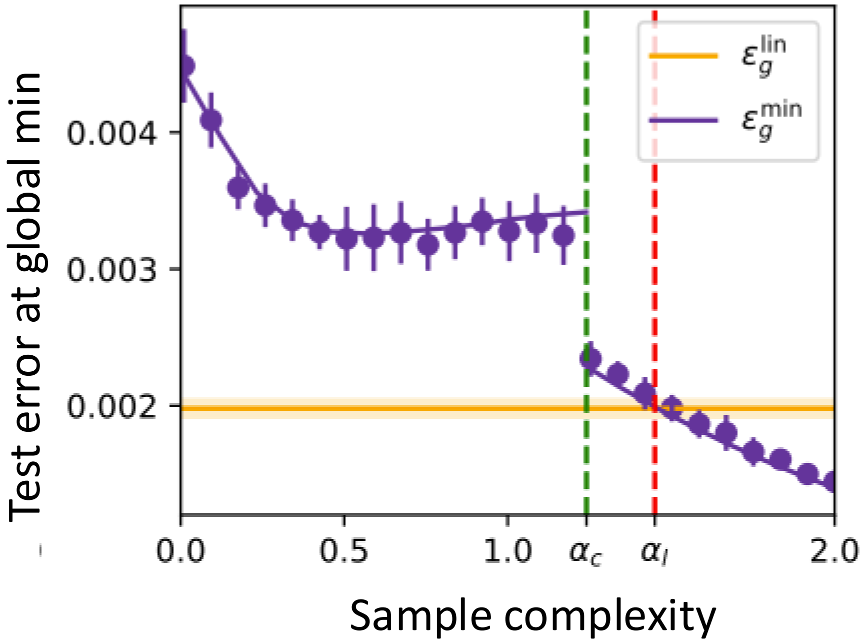
- $\boldsymbol{m} = 0, \boldsymbol{\theta} > 0$

  the elements depend on x :
  semantic mechanism

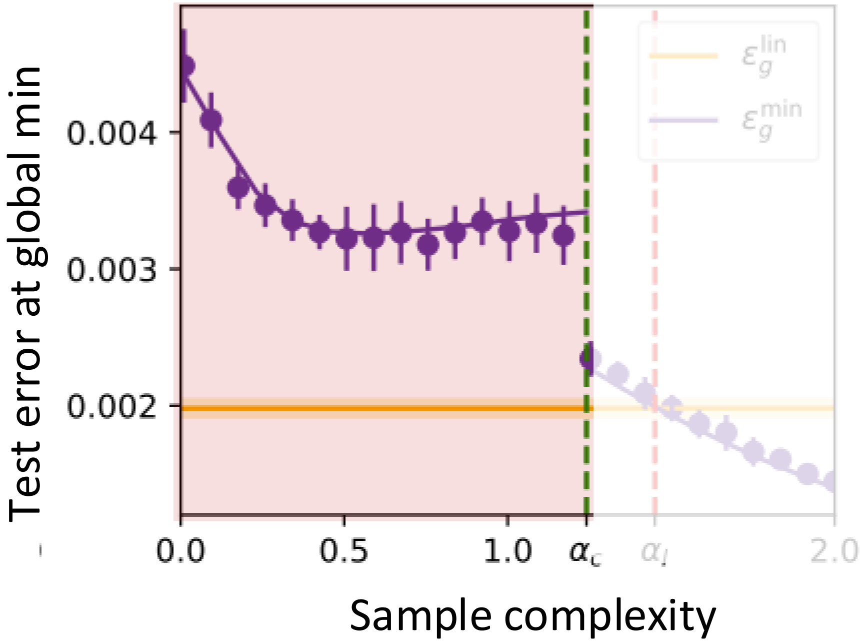


Positional minimum

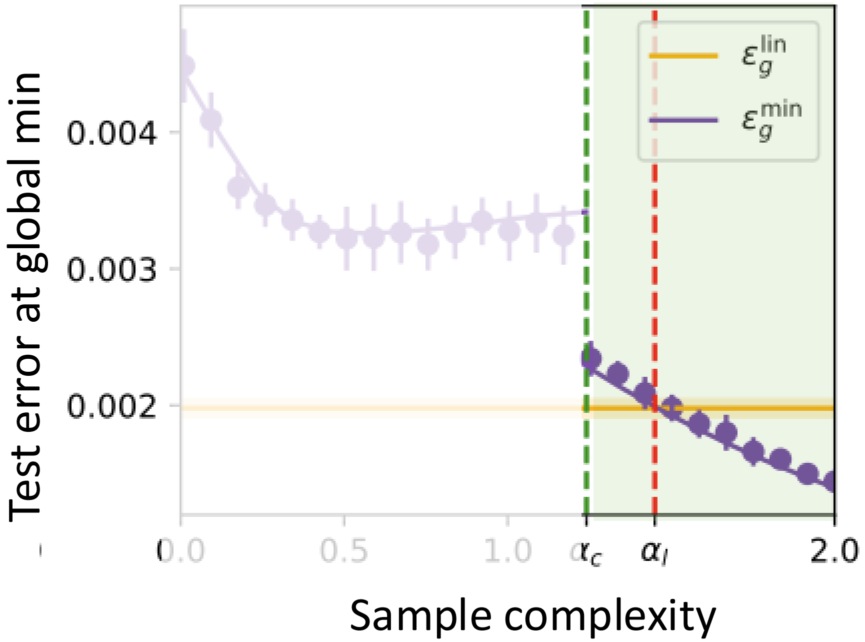
Semantic minimum



Test error at global min
0.004
0.003
0.002
0.0
0.5
1.0
α_c
α_l
2.0
Sample complexity
$\varepsilon_g^{\text{lin}}$
$\varepsilon_g^{\text{min}}$

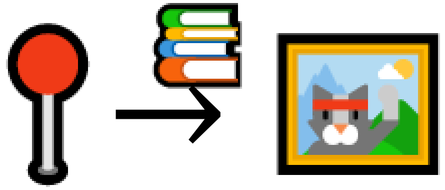


Dot-product attention implements a positional mechanism to approximate the target



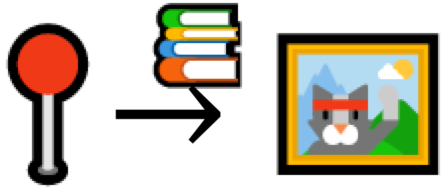
… then learns a semantic mechanism with more data, leading to better generalization.

Recap:



- Toy attention model which characterizes a discrete phase transition between two algorithms (in terms of sample complexity)
- "Emergence" may be discrete in the sense of a first order phase transition

Recap:

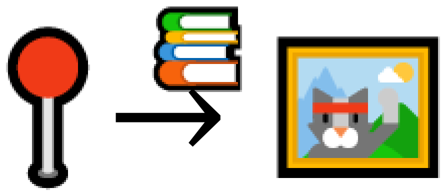


- Toy attention model which charachterizes a discrete phase transition between two algorithms (in terms of sample complexity)
- “Emergence” may be discrete in the sense of a first order phase transition

Questions :

- Dynamics – Does gradient-based training reliably find the minima?
- Architecture – Multiple layers?
- Data – More structured input data? Real tasks?

Recap:



- Toy attention model which charachterizes a discrete phase transition between two algorithms (in terms of sample complexity)
- “Emergence” may be discrete in the sense of a first order phase transition

Questions :

- Dynamics – Does gradient-based training reliably find the minima?
- Architecture – Multiple layers?
- Data – More structured input data? Real tasks?

multiple layers
[Troiani et al 25]
arXiv:2502.00901

dynamics
[Arnaboldi et al 25]
arXiv:2506.02651

Part 1 : A Phase Transition Between Semantic and Positional Learning

arXiv:2402.03902 – Hugo Cui, Freya Behrens, Florent Krzakala, Lenka Zdeborová

How is the learned algorithm determined by the sample complexity?
Do different algorithms emerge spontaneously?

Part 2 : The Interplay between Attention and Feed-Forward Layers

arXiv:2407.11542 – Freya Behrens, Luca Biggio, Lenka Zdeborová

How is the learned algorithm determined by architectural choices?
Which functions are executed by which parts?

Histogram task : for each token, output the number of identical tokens in the sequence

[Weiss et al '21]

```
       Input          -> Output
Ex1:[B,A,A,D,E] -> [1,2,2,1,1]
Ex2:[A,C,C,A,A] -> [3,2,2,3,3]
Ex3:[C,C,C,C,D] -> [ , , , , ]
```

Histogram task : for each token, output the number of identical tokens in the sequence

[Weiss et al '21]

```
       Input        -> Output
Ex1:[B,A,A,D,E] -> [1,2,2,1,1]
Ex2:[A,C,C,A,A] -> [3,2,2,3,3]
Ex3:[C,C,C,C,D] -> [4,4,4,4,1]
```

Histogram task : for each token, output the number of identical tokens in the sequence

[Weiss et al ‘21]

```
        Input          -> Output
Ex1:[B,A,A,D,E] -> [1,2,2,1,1]
Ex2:[A,C,C,A,A] -> [3,2,2,3,3]
Ex3:[C,C,C,C,D] -> [4,4,4,4,1]
```

{A,B,C,D,E} – set of tokens

L – sequence length

T – alphabet size

Why a counting task?

Why a counting task?

- Counting: localization and subsequent measurement
- Language models are bad/brittle at counting [Ouellette '24]
- Contribute to understanding a zoology of algorithmic tasks in networks

Histogram task : for each token, output the number of identical tokens in the sequence

[Weiss et al '21]

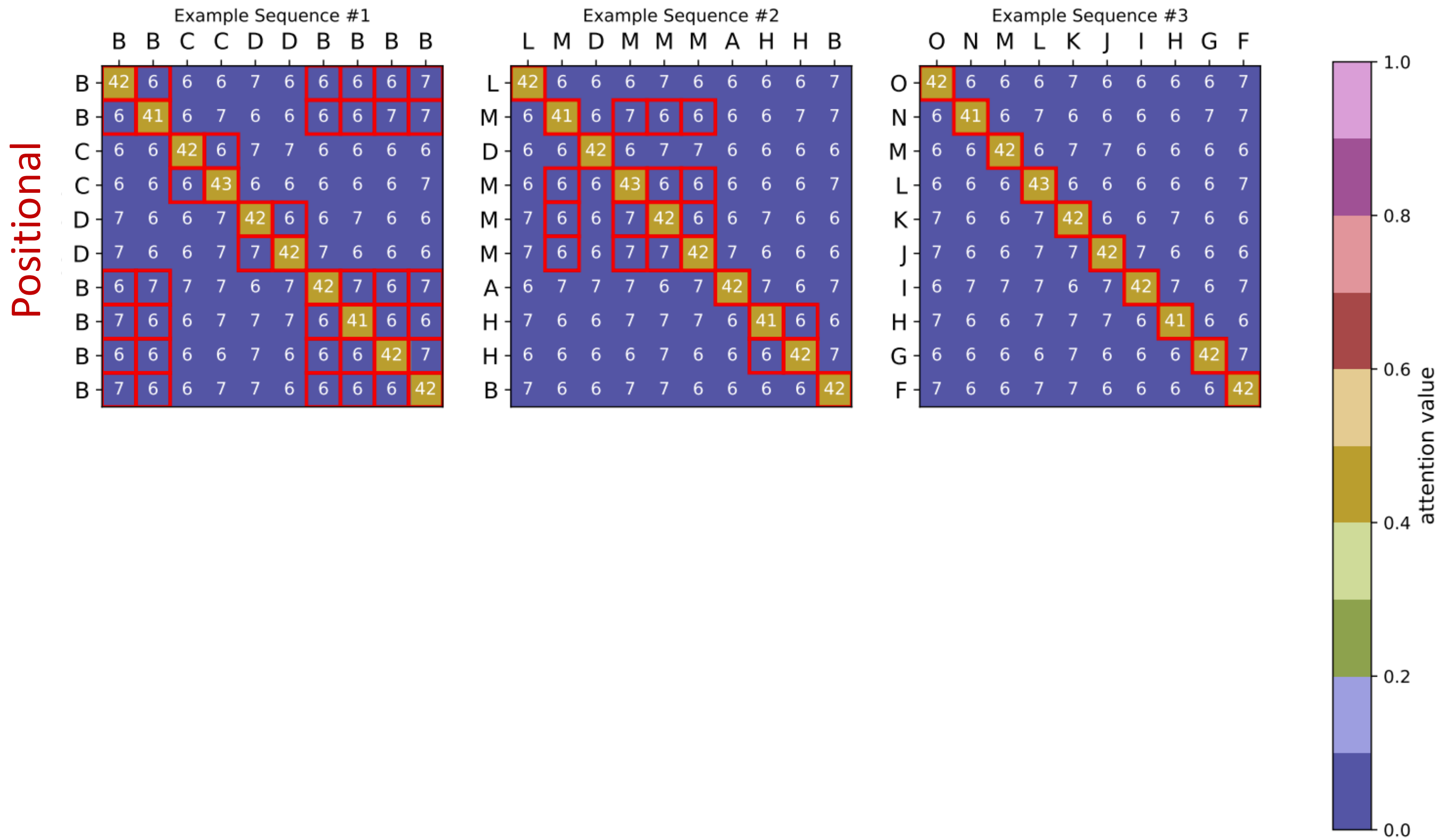
```
       Input         -> Output
Ex1:[B,A,A,D,E] -> [1,2,2,1,1]
Ex2:[A,C,C,A,A] -> [3,2,2,3,3]
Ex3:[C,C,C,C,D] -> [4,4,4,4,1]
```

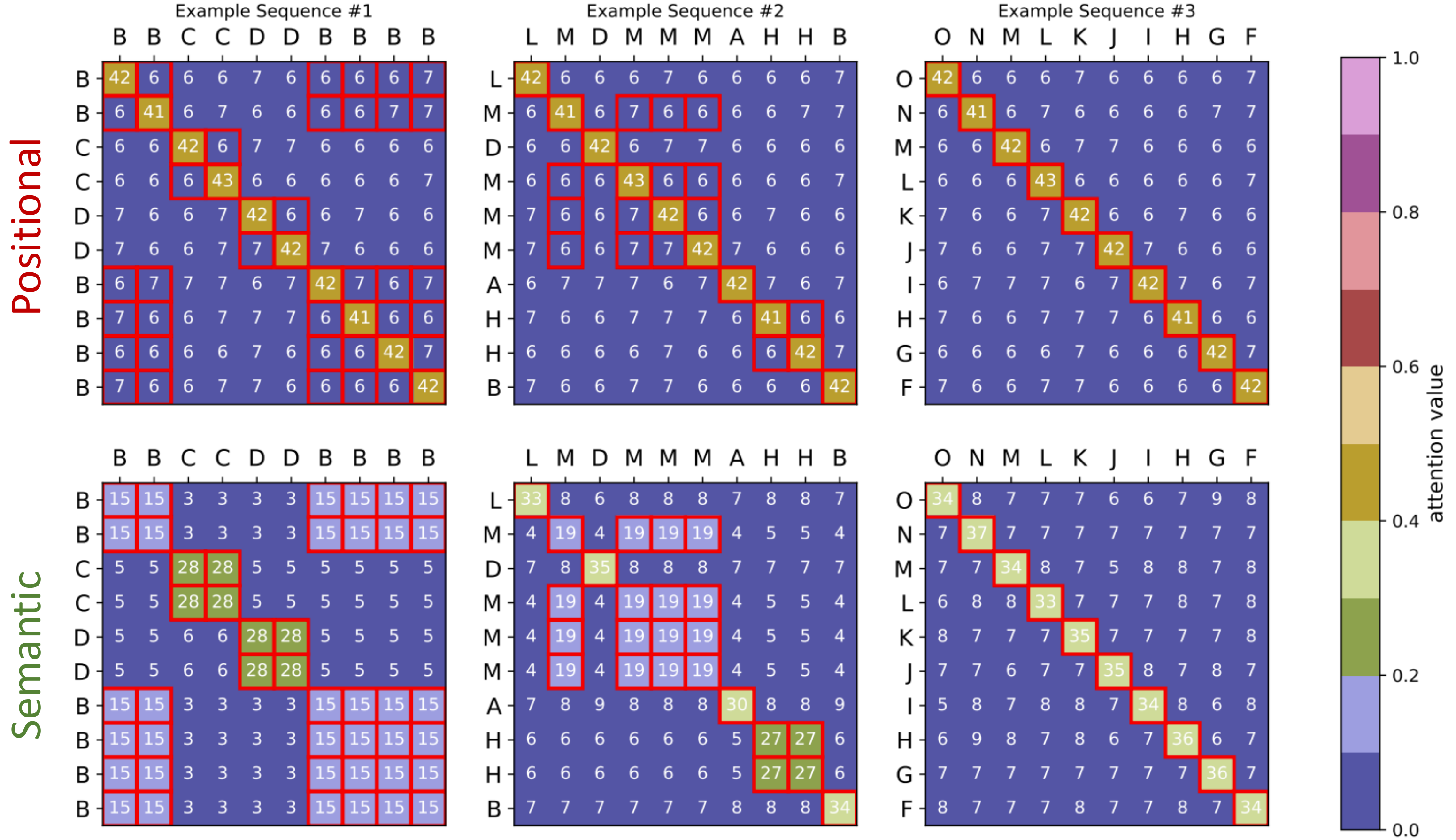
{A,B,C,D,E} – set of tokens

L – sequence length

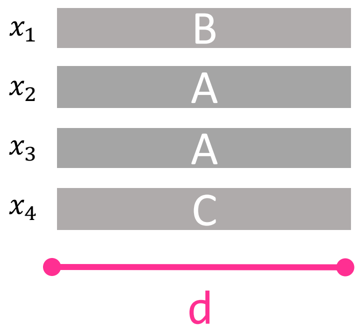
T – alphabet size

(How) Can we solve the task with a one layer transformer?

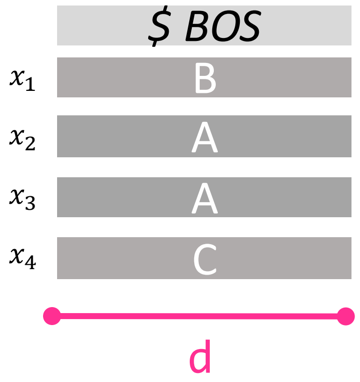

Positional
Example Sequence #1
Example Sequence #2
Example Sequence #3
attention value



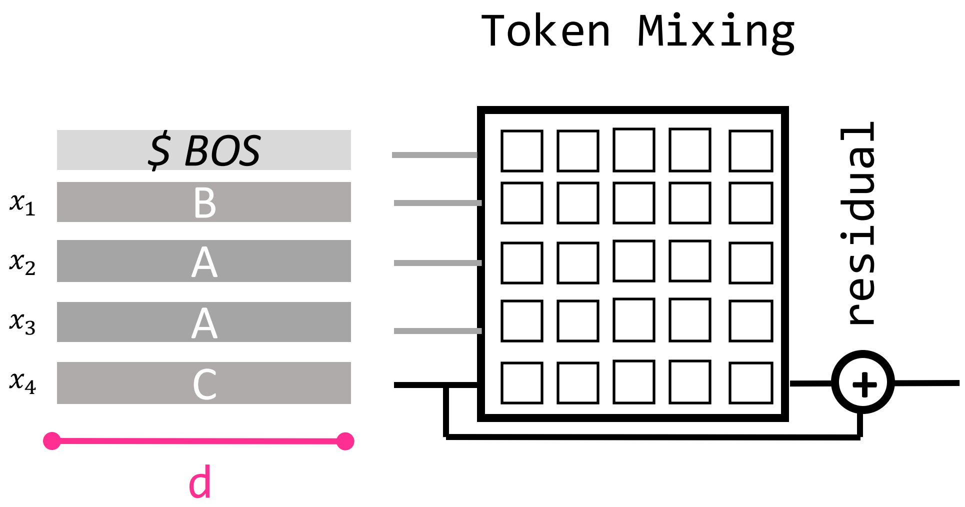
Example Sequence #1
Example Sequence #2
Example Sequence #3
Positional
Semantic
attention value
1.0
0.8
0.6
0.4
0.2
0.0



$$\bar{\mathbf{x}} \in \mathbb{R}^{L \times d}$$

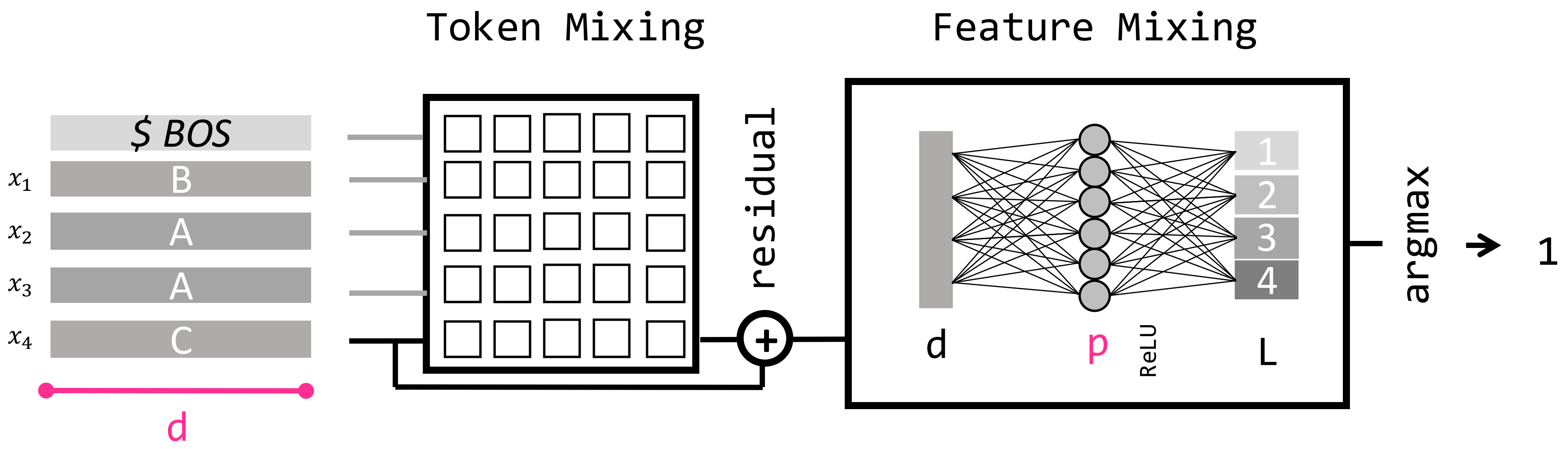


$$\bar{\mathbf{x}} \in \mathbb{R}^{L \times d}$$



$$\bar{\mathbf{x}} \in \mathbb{R}^{L \times d} \qquad \bar{x}'_\ell = \bar{x}_\ell + [\mathbf{A}(\bar{\mathbf{x}})\bar{\mathbf{x}}]_\ell$$

$$a_{ij} = \frac{1}{\sqrt{d}} \langle x_i W_Q, x_j W_K \rangle$$

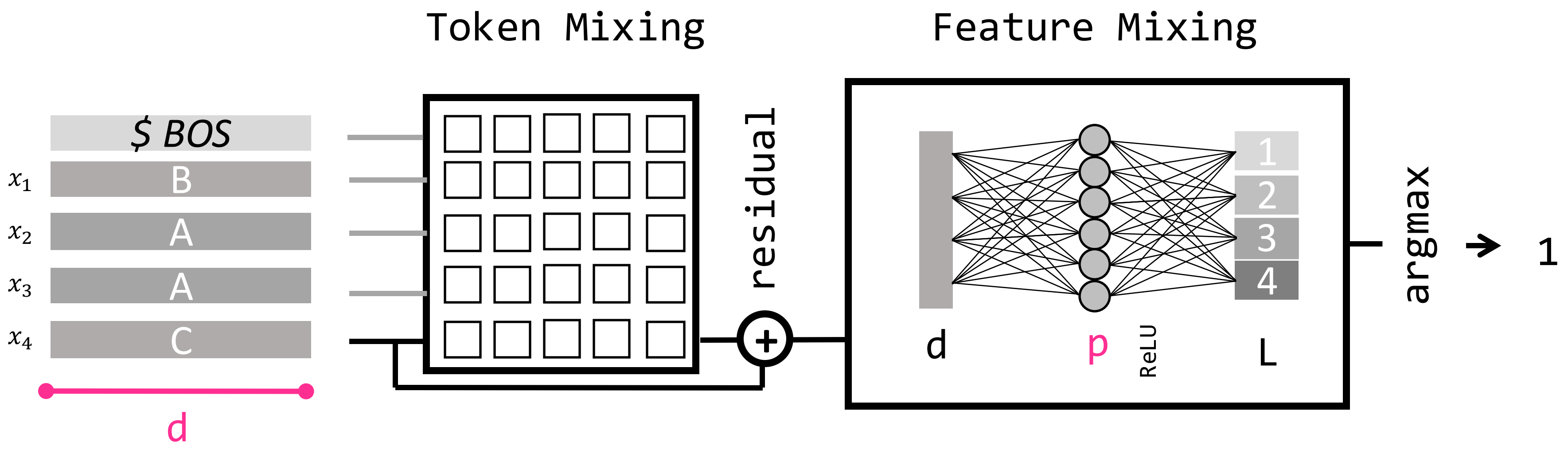


$$\bar{\mathbf{x}} \in \mathbb{R}^{L \times d}$$

$$\bar{x}'_\ell = \bar{x}_\ell + [\mathbf{A}(\bar{\mathbf{x}})\bar{\mathbf{x}}]_\ell$$

$$f(\bar{x}'_\ell) = \mathrm{ReLU}(\bar{x}'_\ell W_1 + b_1)W_2 + b_2$$

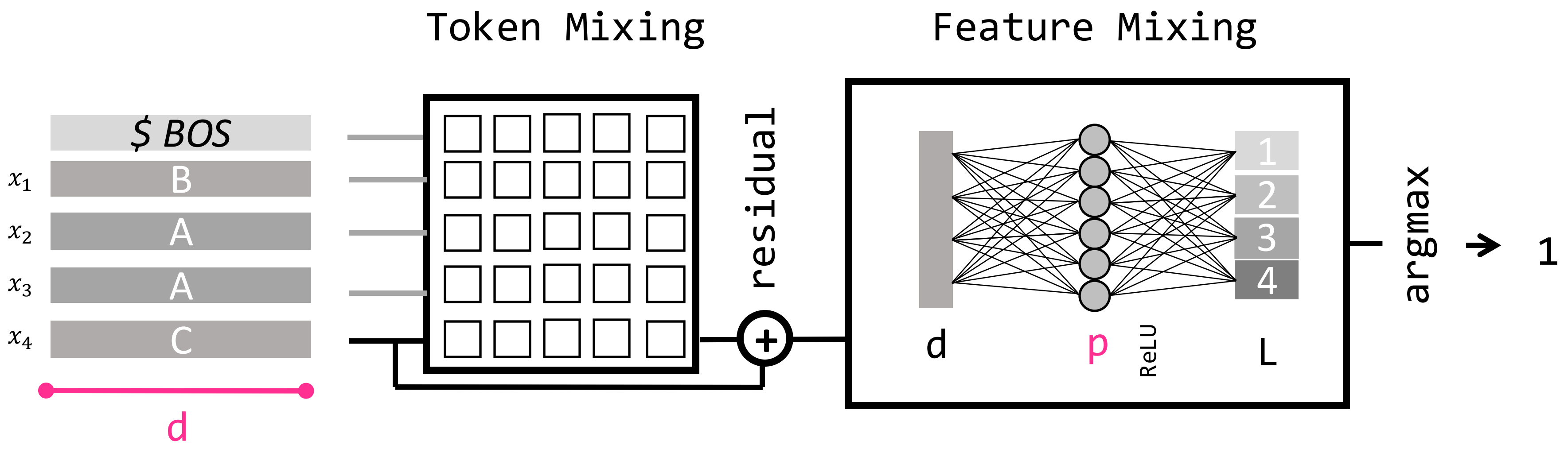
$$a_{ij} = \frac{1}{\sqrt{d}} \langle x_i W_Q, x_j W_K \rangle$$



$\bar{\mathbf{x}} \in \mathbb{R}^{L\times d}$ $\quad \bar{x}'_\ell = \bar{x}_\ell + [\mathbf{A}(\bar{\mathbf{x}})\bar{\mathbf{x}}]_\ell \quad f(\bar{x}'_\ell) = \mathrm{ReLU}(\bar{x}'_\ell W_1 + b_1)W_2 + b_2$

$$a_{ij} = \frac{1}{\sqrt{d}} \langle x_i W_Q, x_j W_K \rangle$$

We don’t want to deal with positional encodings



$$\bar{\mathbf{x}} \in \mathbb{R}^{L \times d} \qquad \bar{x}'_\ell = \bar{x}_\ell + [\mathbf{A}(\bar{\mathbf{x}})\bar{\mathbf{x}}]_\ell \qquad f(\bar{x}'_\ell) = \mathrm{ReLU}(\bar{x}'_\ell W_1 + b_1)W_2 + b_2$$

$$a_{ij} = \frac{1}{\sqrt{d}} \langle x_i W_Q, x_j W_K \rangle$$

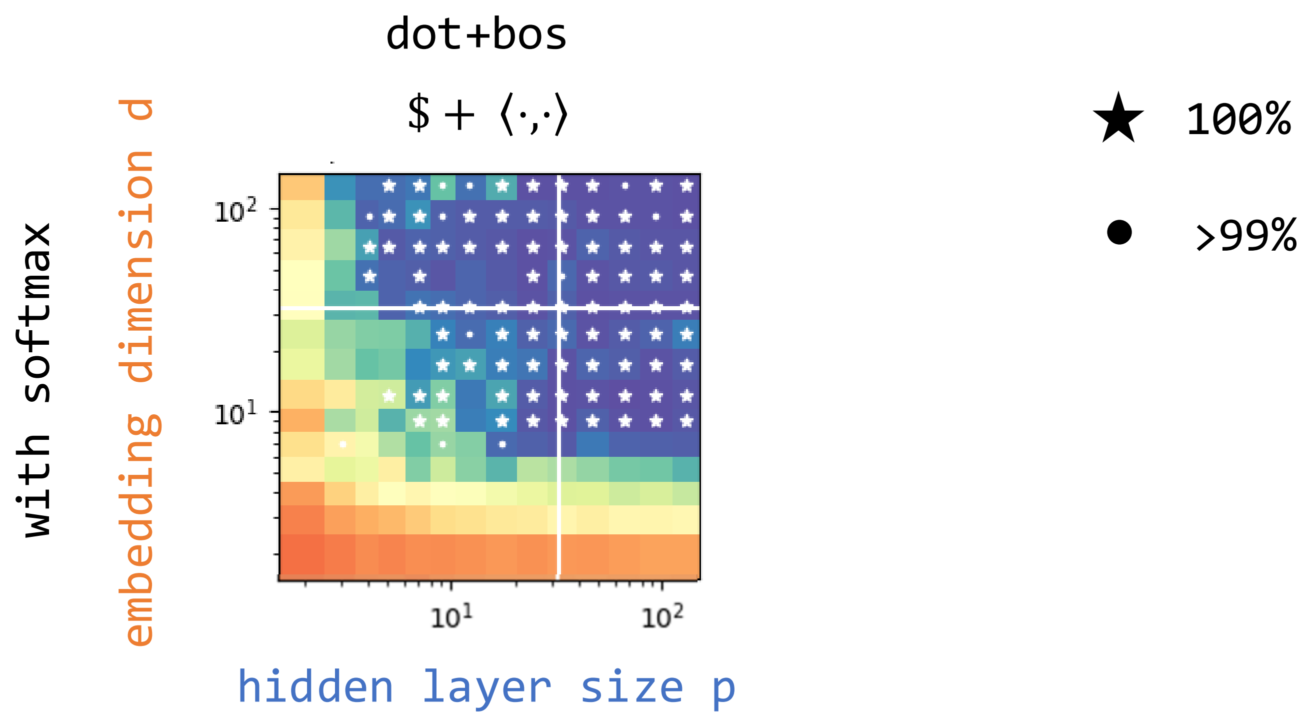
We don't want to deal with positional encodings

Embedding, token and feature mixing are learned - online

In which regimes can we learn perfect solutions?

d, p

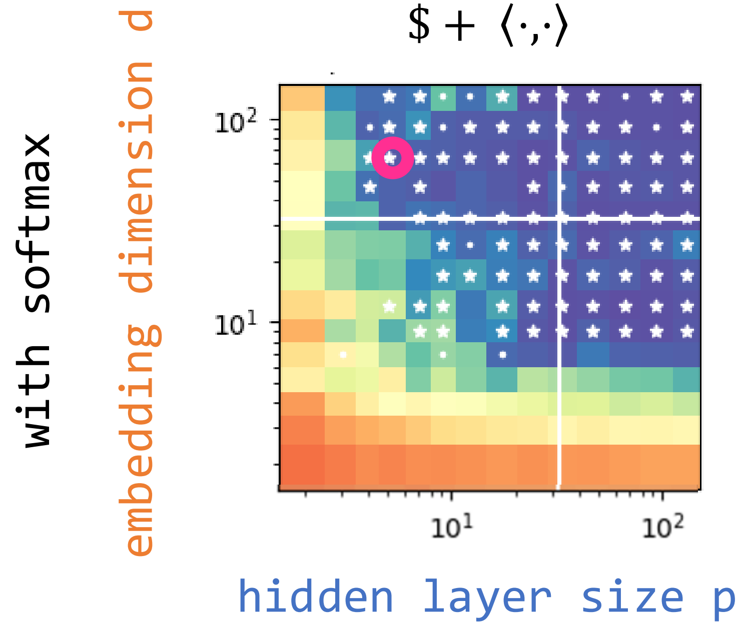
T=32, L=10


dot+bos
$ + ⟨·,·⟩
with softmax
embedding dimension d
hidden layer size p
10^2
10^1
10^1
10^2
★ 100%
● >99%

T=32, L=10

dot+bos

$\$ + \langle \cdot, \cdot \rangle$



★ 100%

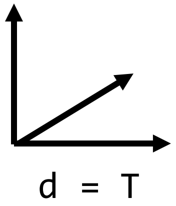
● >99%

What are possible mechanisms?

`Ex1:[$,B,A,A,D,E] -> [-,1,2,2,1,1]`

dot+bos

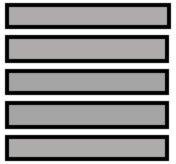
$\$ + \langle \cdot , \cdot \rangle$



α

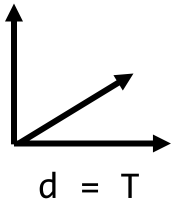
What are possible mechanisms?

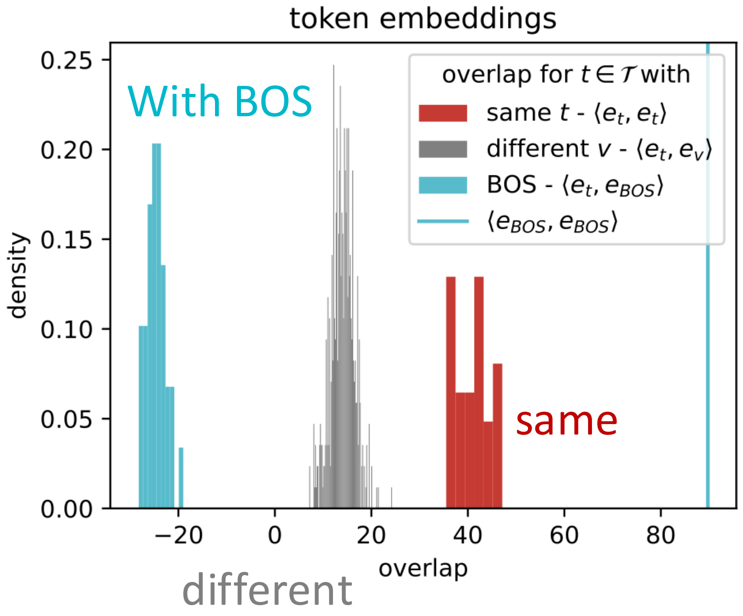
Ex1:[$,B,A,A,D,E] -> [-,1,2,2,1,1]



dot+bos

$\$ + \langle \cdot,\cdot \rangle$



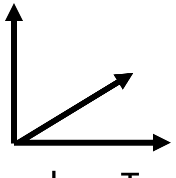


α

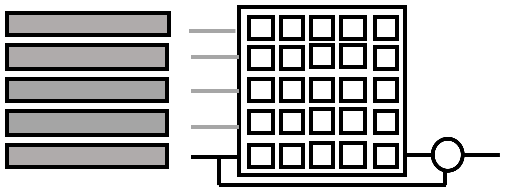
What are possible mechanisms?

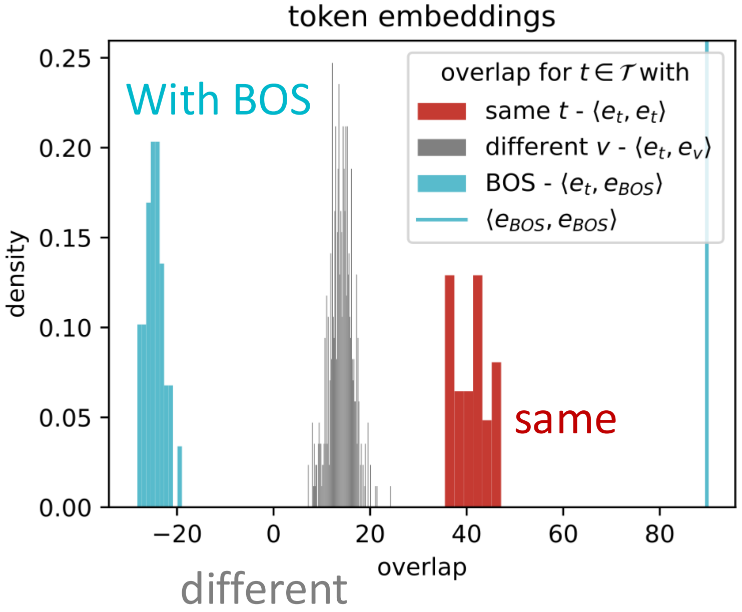
dot+bos

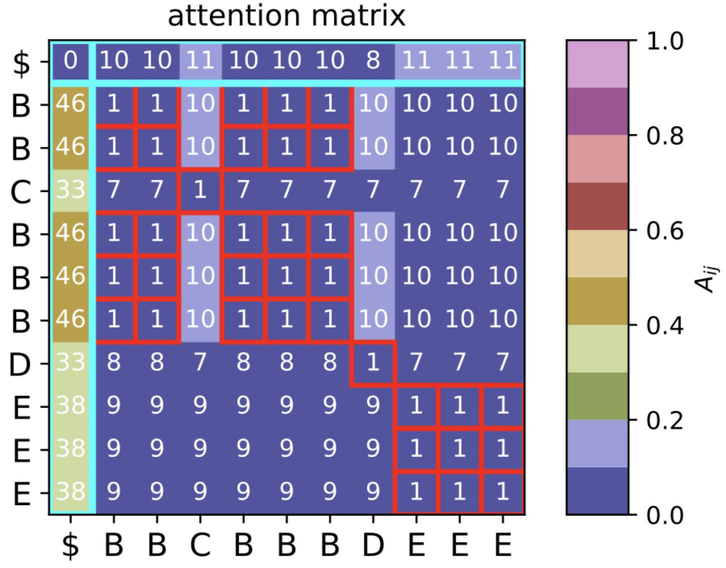
$\$ + \langle \cdot,\cdot \rangle$



`Ex1:[$,B,A,A,D,E] -> [-,1,2,2,1,1]`



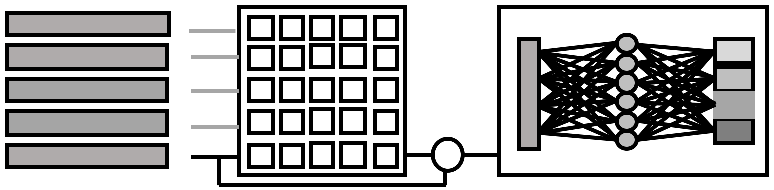




α

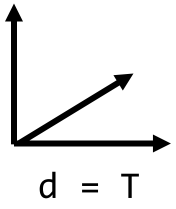
What are possible mechanisms?

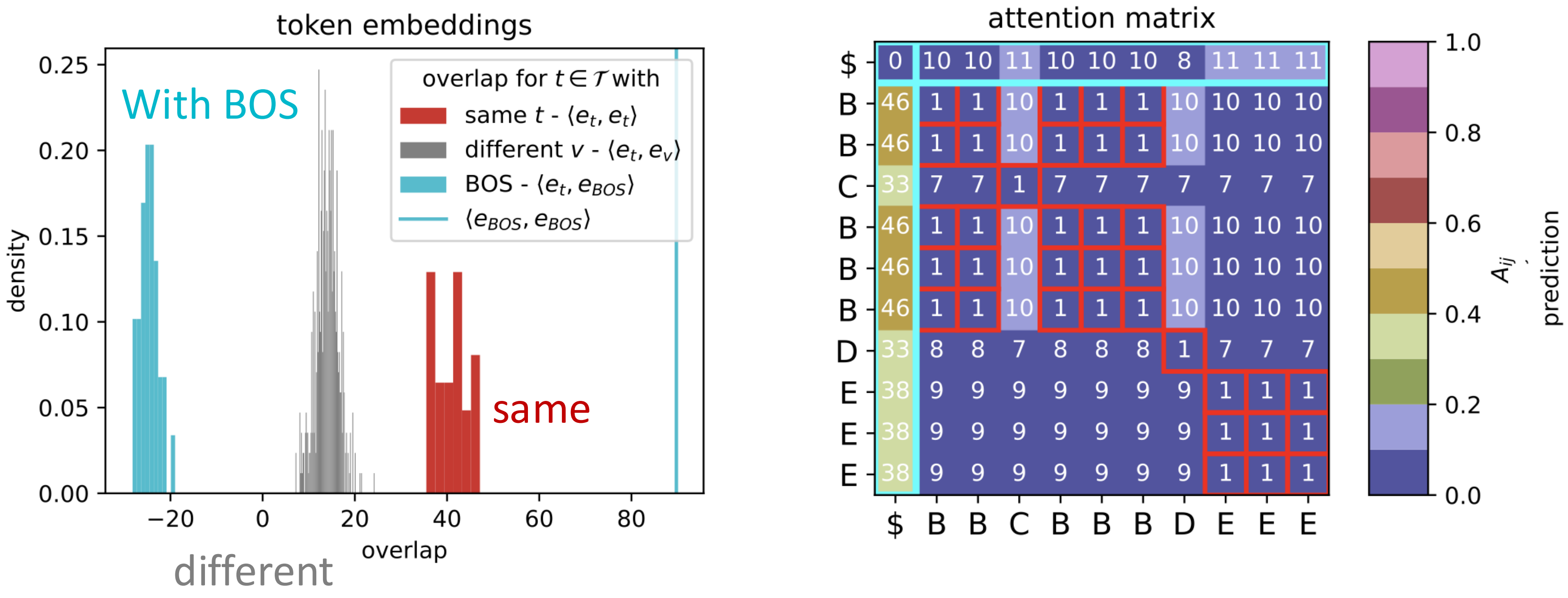
Ex1:[$,B,A,A,D,E] -> [-,1,2,2,1,1]

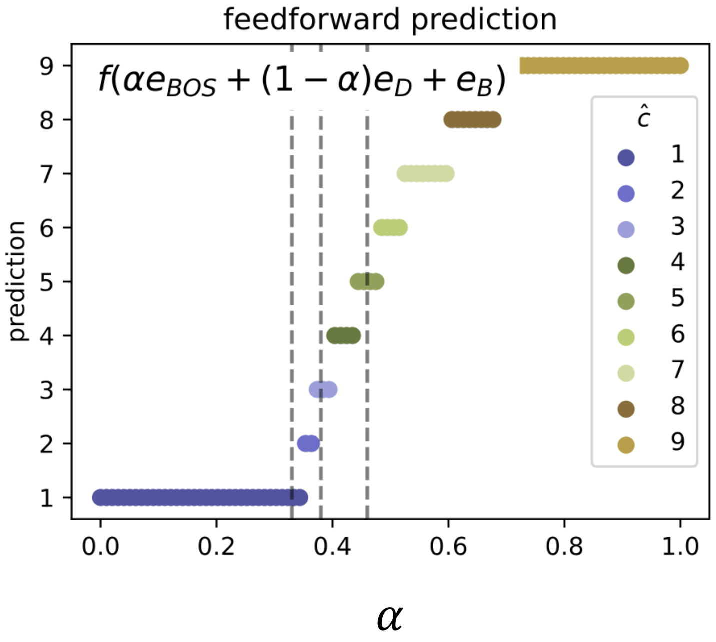


dot+bos

$ + ⟨·,·⟩







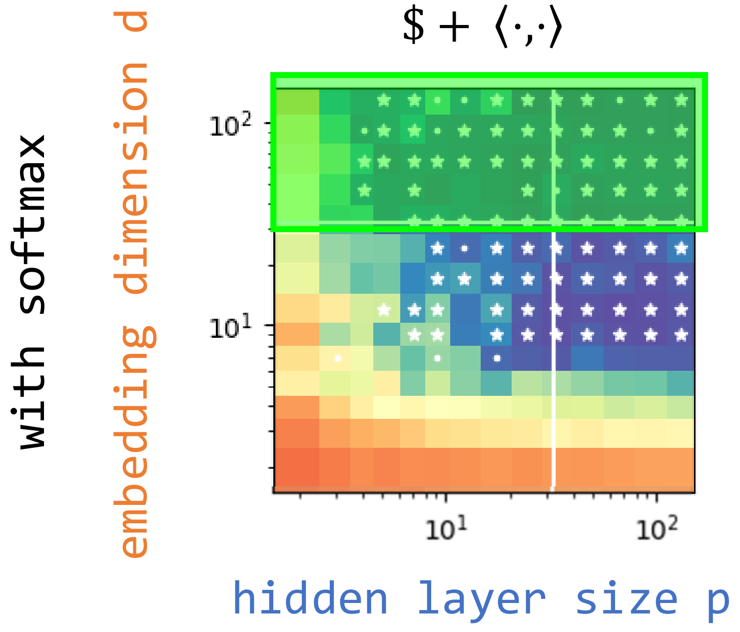
dot+bos

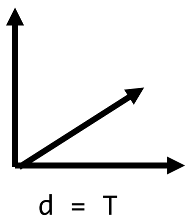
$\$ + \langle\cdot,\cdot\rangle$

T=32, L=10

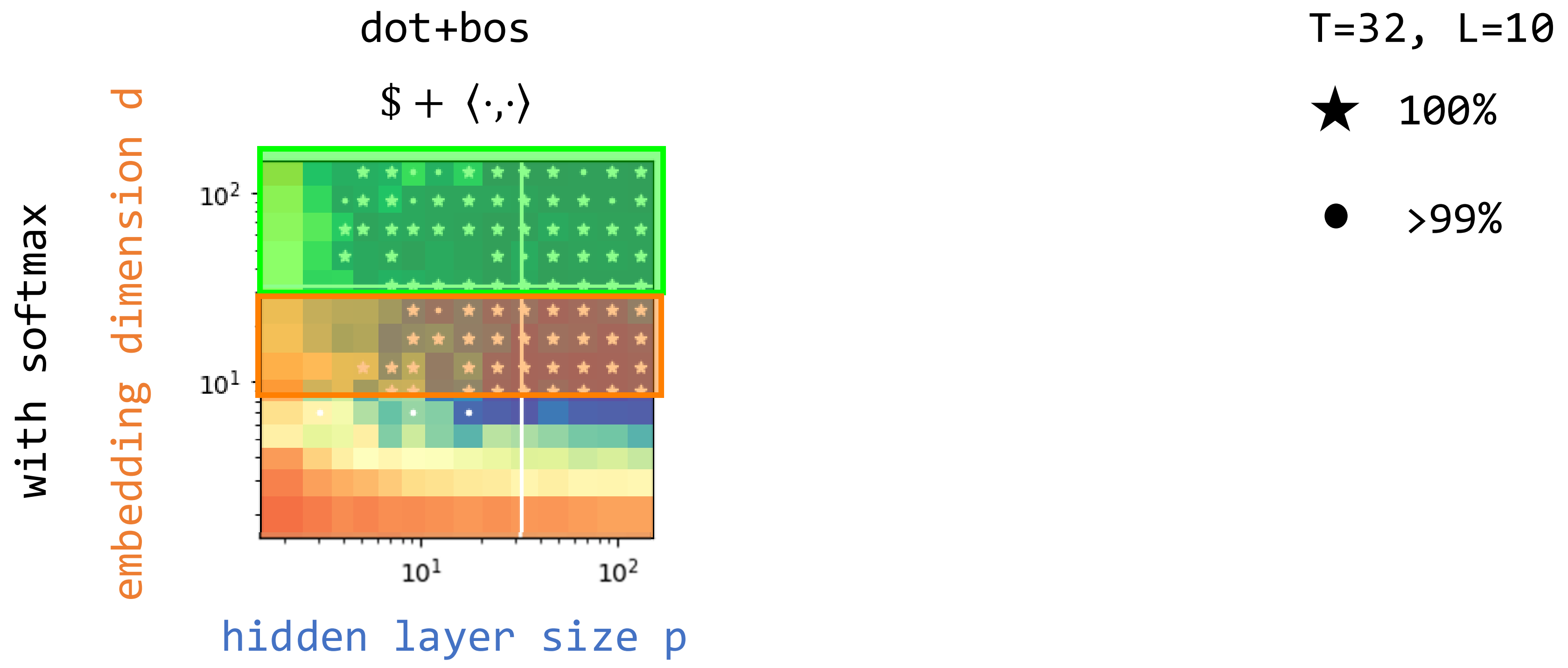
★ 100%

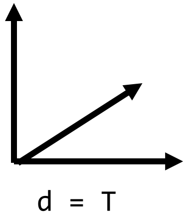
● >99%



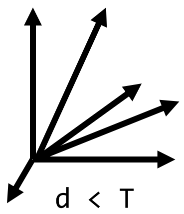


Proposition (Relation-based Counting with BOS token).
For **dot+bos**+sftm and given L ≥ 2, there each exists a configuration of weights that solves the histogram task at 100% accuracy, given that d ≥ T >2 and **p=1**.





Proposition (Relation-based Counting with BOS token).
For **dot+bos**+sftm and given L ≥ 2, there each exists a configuration of weights that solves the histogram task at 100% accuracy, given that d ≥ T >2 and p=1.



Proposition (Robustness via softmax error-reduction).
For dot+bos+**sftm** and given T,L > 2, there exist weight configurations that solve the histogram task with $d \geq \lceil \log_2(T+1) \rceil + 2$.

Histogram task : for each token, output the number of identical tokens in the sequence

[Weiss et al '21]

```
        Input        -> Output
Ex1:[B,A,A,D,E] -> [1,2,2,1,1]
Ex2:[A,C,C,A,A] -> [3,2,2,3,3]
Ex3:[C,C,C,C,D] -> [4,4,4,4,1]
```

{A,B,C,D,E} – set of tokens

L – sequence length

T – alphabet size

ok

(How) Can we solve the task with a one layer transformer? yes

Histogram task : for each token, output the number of identical tokens in the sequence

[Weiss et al '21]

```
        Input          -> Output
Ex1:[B,A,A,D,E] -> [1,2,2,1,1]
Ex2:[A,C,C,A,A] -> [3,2,2,3,3]
Ex3:[C,C,C,C,D] -> [4,4,4,4,1]
```

{A,B,C,D,E} – set of tokens

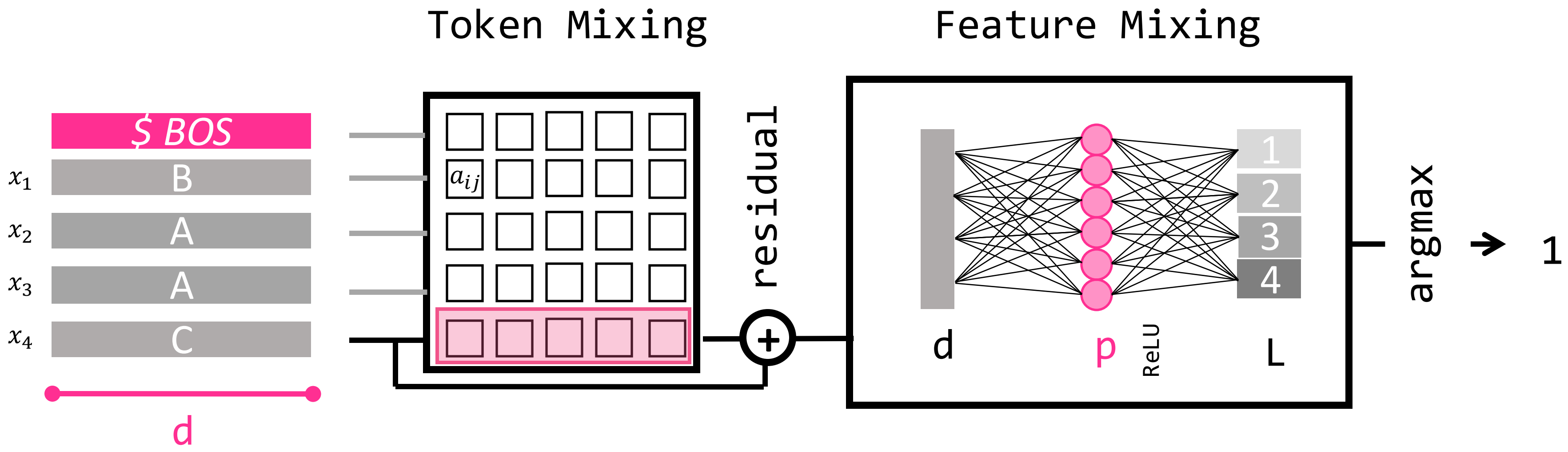
L – sequence length

T – alphabet size

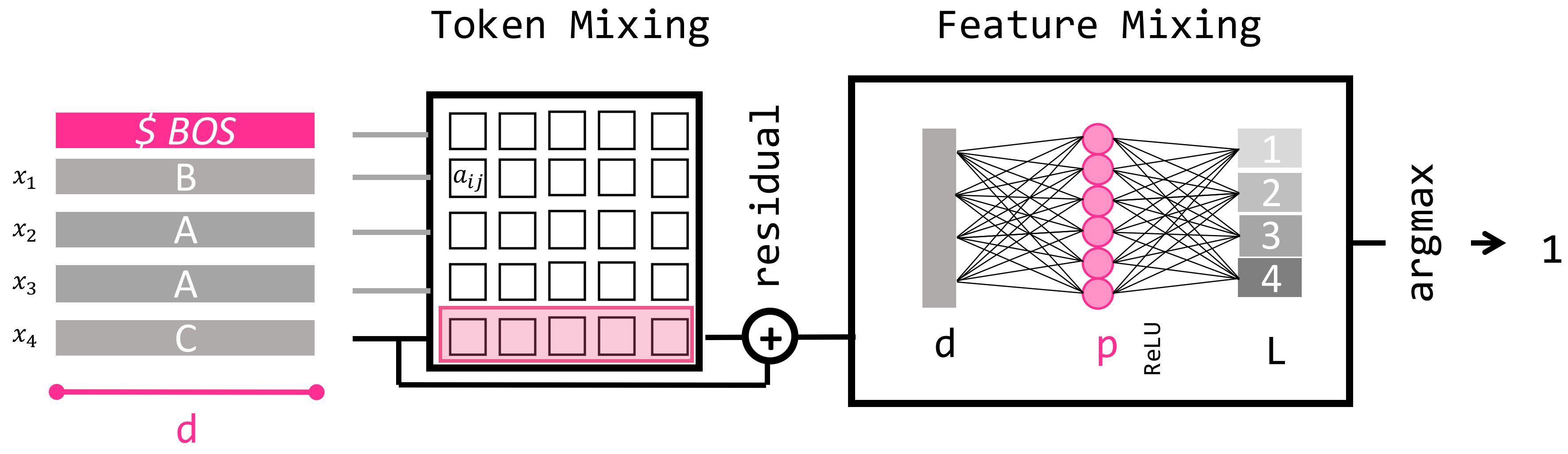
ok

(How) Can we solve the task with a one layer transformer? yes

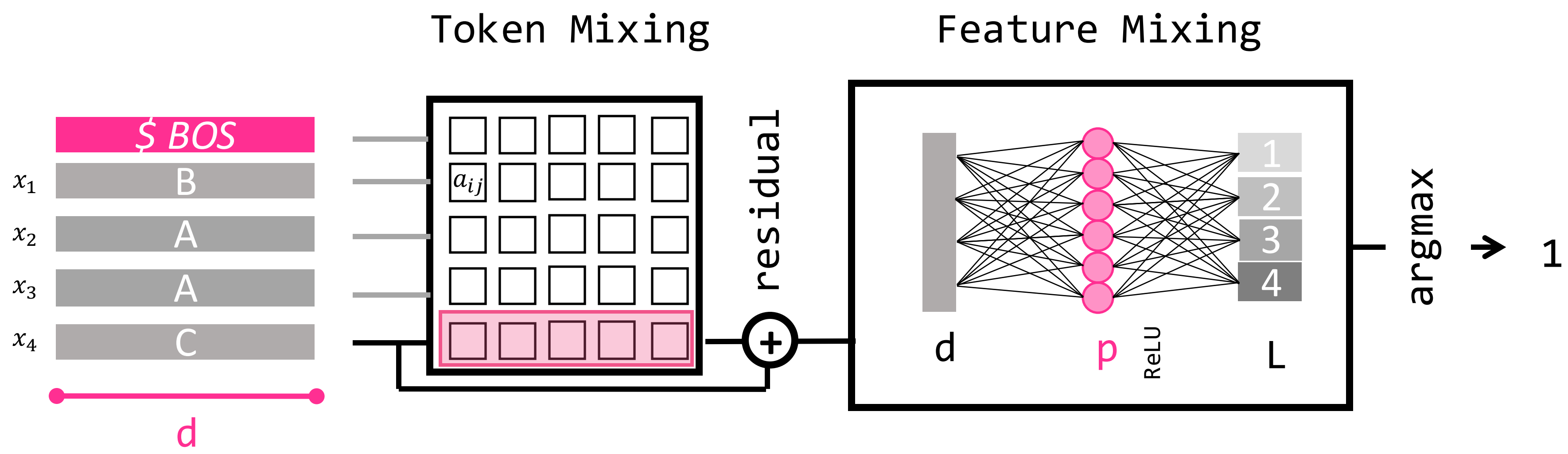
Dot-product? Linear? State Space? Scratchpad? Chain-of-Thought? Heads?
Hidden neurons? Activation function? Prompting?



Token Mixing
Feature Mixing
$ BOS
x_1
B
x_2
A
x_3
A
x_4
C
d
a_{ij}
residual
d
p
ReLU
L
1
2
3
4
argmax
1



Several configurations : L, T, (bos), (+sftm), d, p



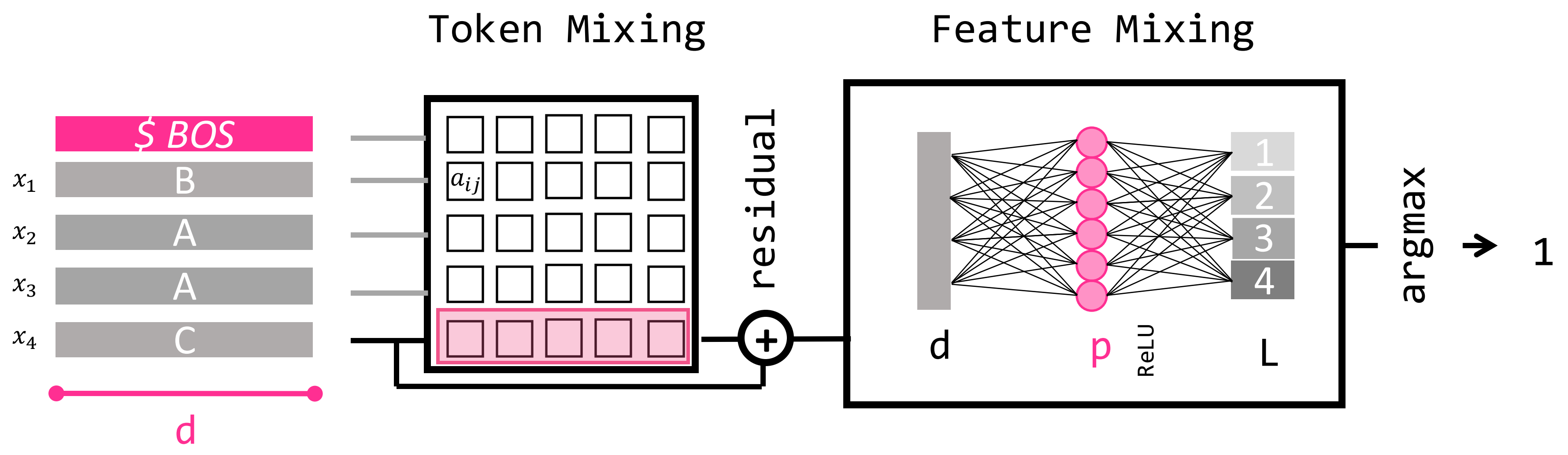
Several configurations : L, T, (bos), (+sftm), d, p

Token Mixing :

(dot) $a_{ij} = \frac{1}{\sqrt{d}} \langle x_i W_Q, x_j W_K \rangle$

or

(linear) $a_{ij} = c_{ij}$



Several configurations : L, T, (bos), (+sftm), d, p

Token Mixing : (dot) $a_{ij} = \frac{1}{\sqrt{d}} \langle x_i W_Q, x_j W_K \rangle$

or

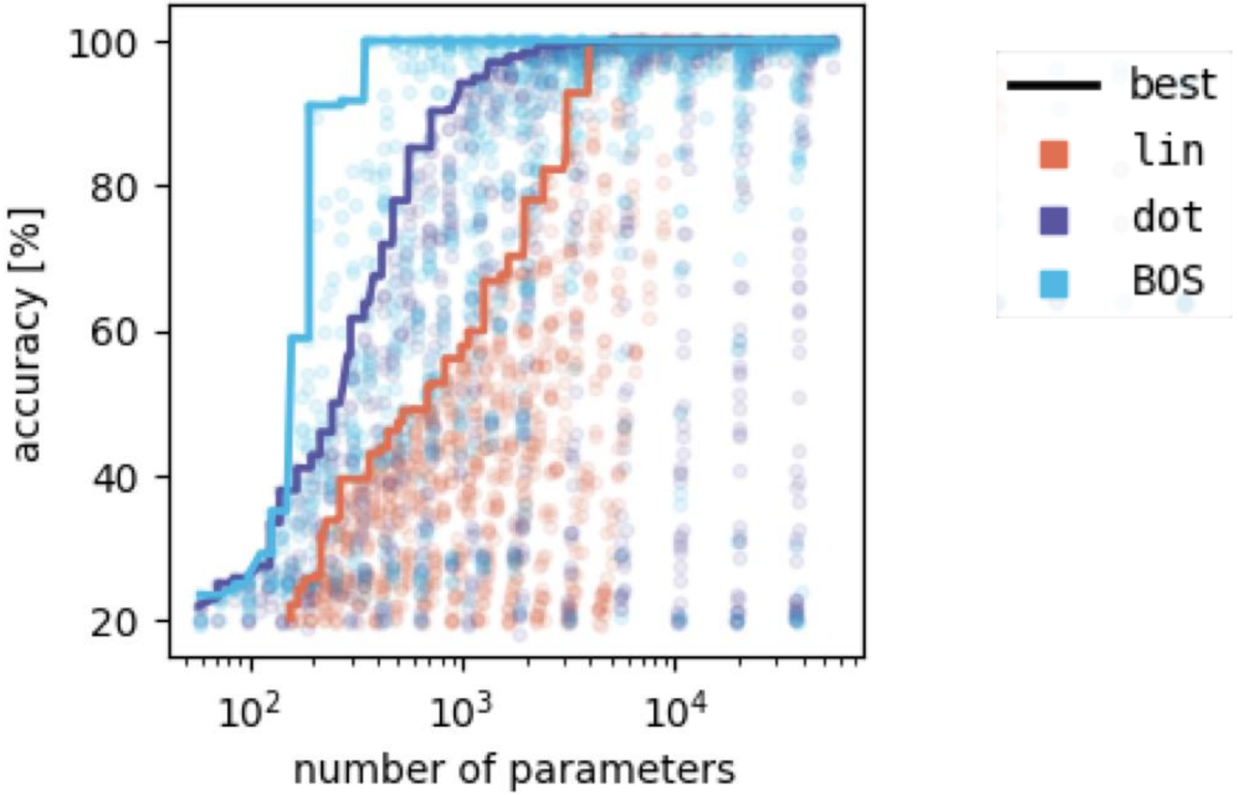
(linear) $a_{ij} = c_{ij}$

Embedding, token and feature mixing are learned

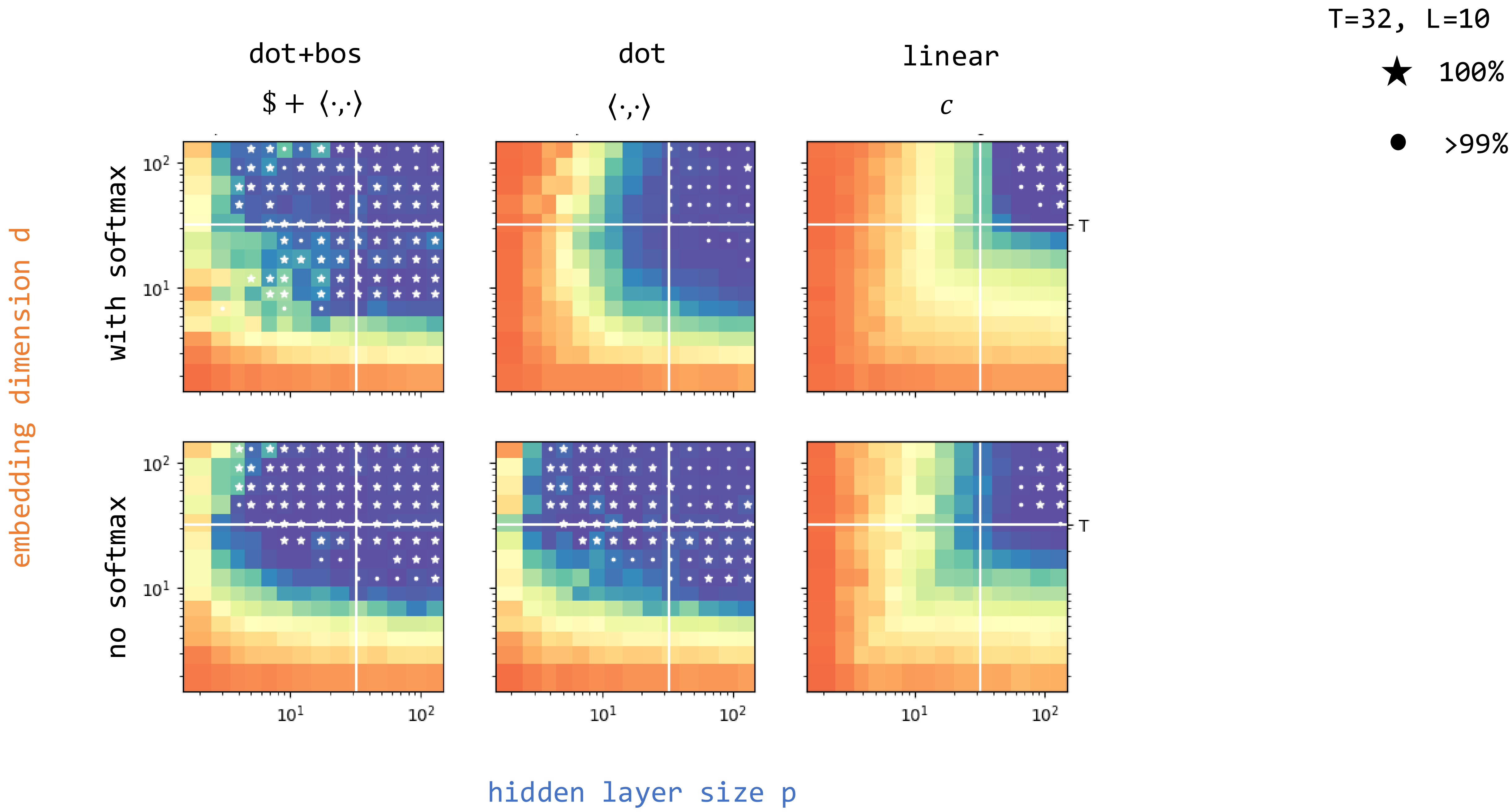
In which regimes can we learn perfect solutions?
attention, T, L, d, p

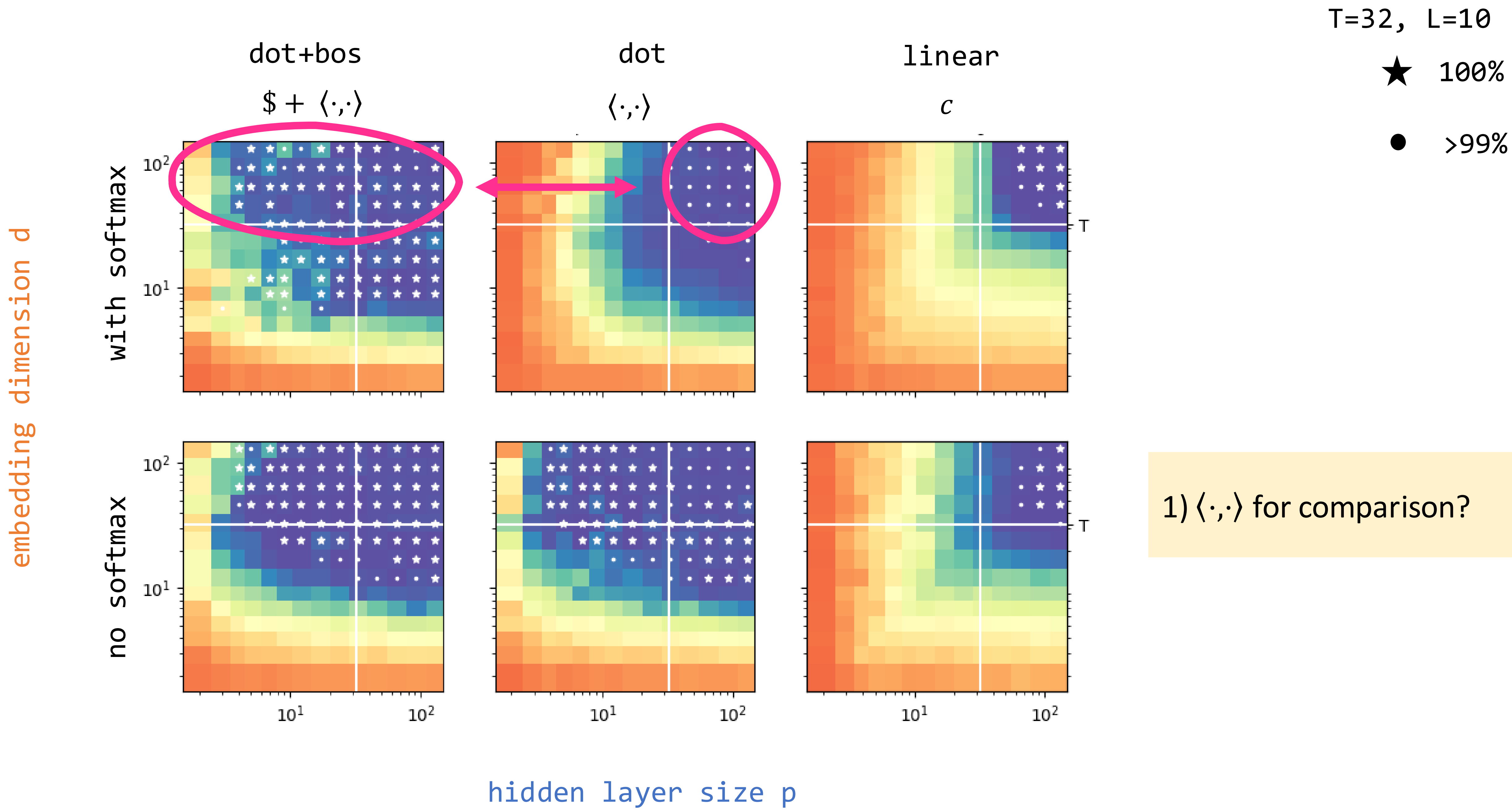
T=32, L=10

(+sftm)

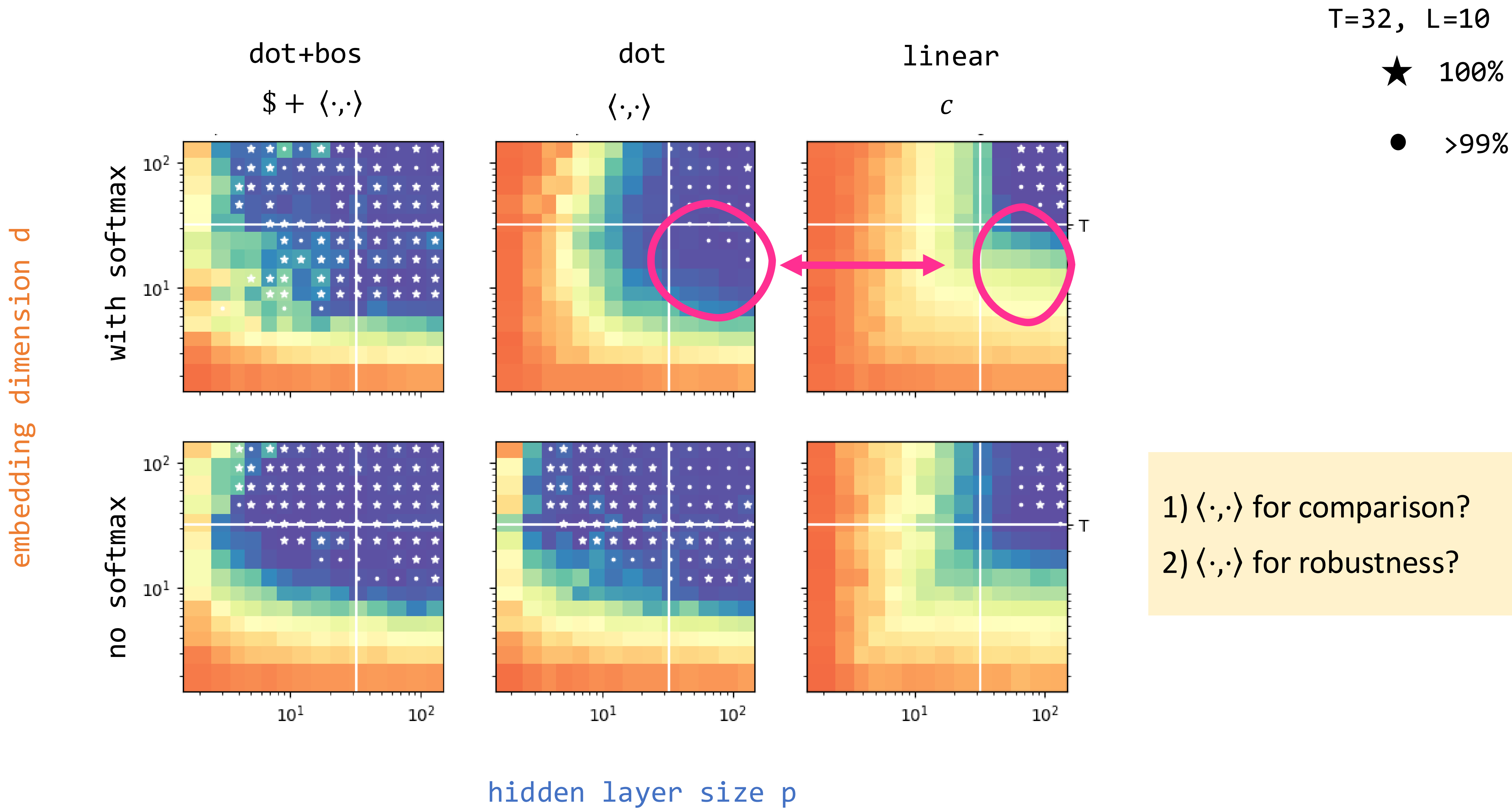


d, p


T=32, L=10
100%
>99%
dot+bos
$ + ⟨·,·⟩
dot
⟨·,·⟩
linear
c
with softmax
no softmax
embedding dimension d
hidden layer size p
T

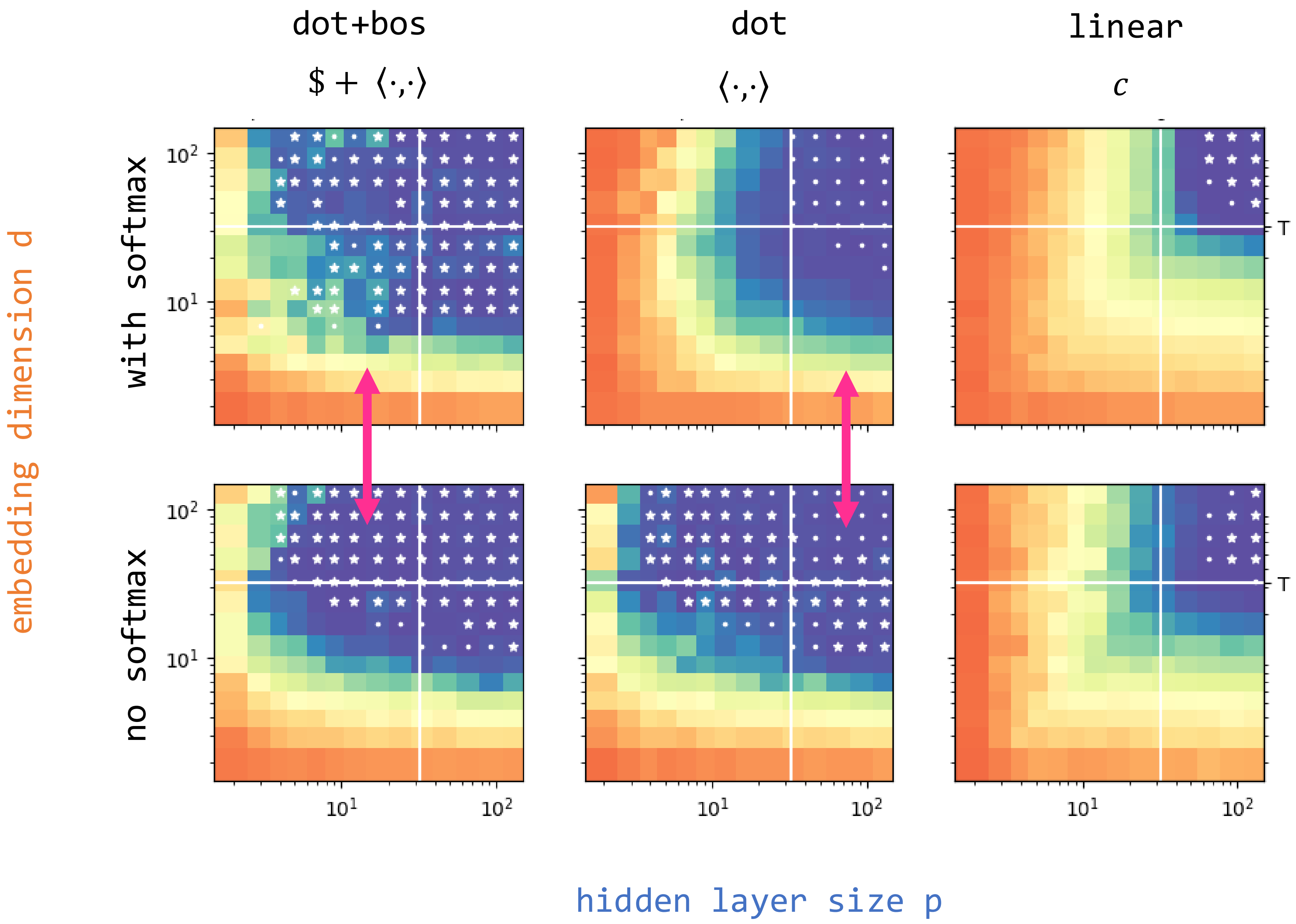


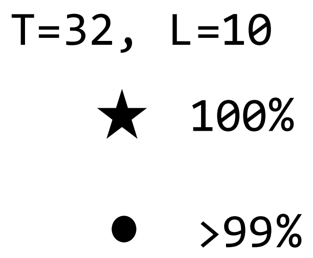
1) ⟨·,·⟩ for comparison?



1) ⟨·,·⟩ for comparison?

2) ⟨·,·⟩ for robustness?



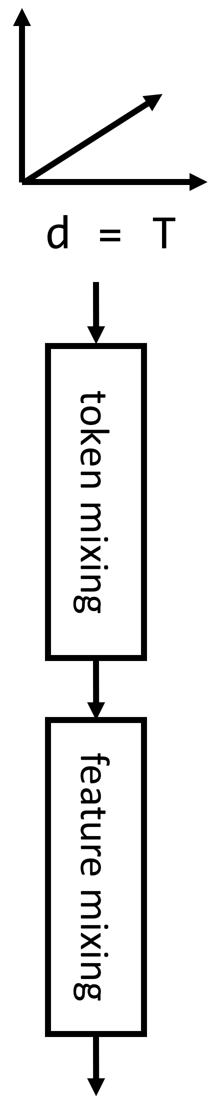


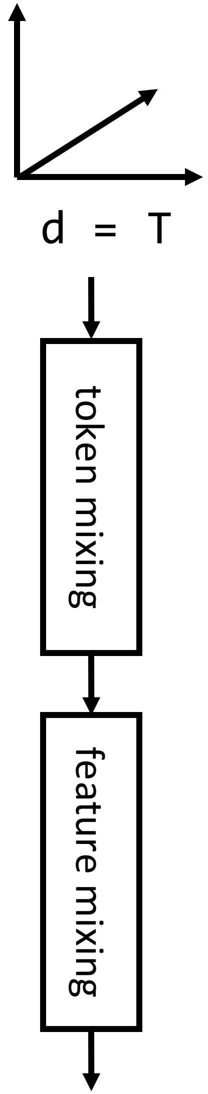
1) ⟨·,·⟩ for comparison?

2) ⟨·,·⟩ for robustness?

3) Softmax helps?

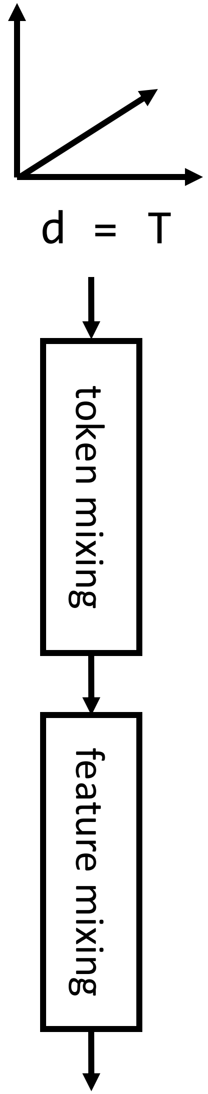
How do the models solve the tasks?


d = T
token mixing
feature mixing



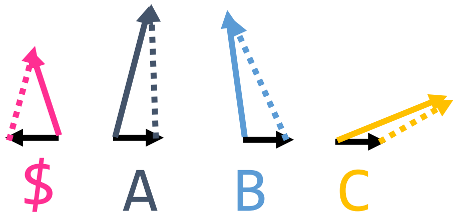
Relation-based counting:

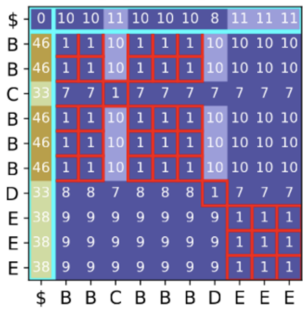
- A is for comparing + recording counting “anchor”
- `f()` is for reading counting subspace magnitude
- p=1 is enough

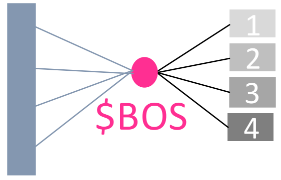


Relation-based counting:

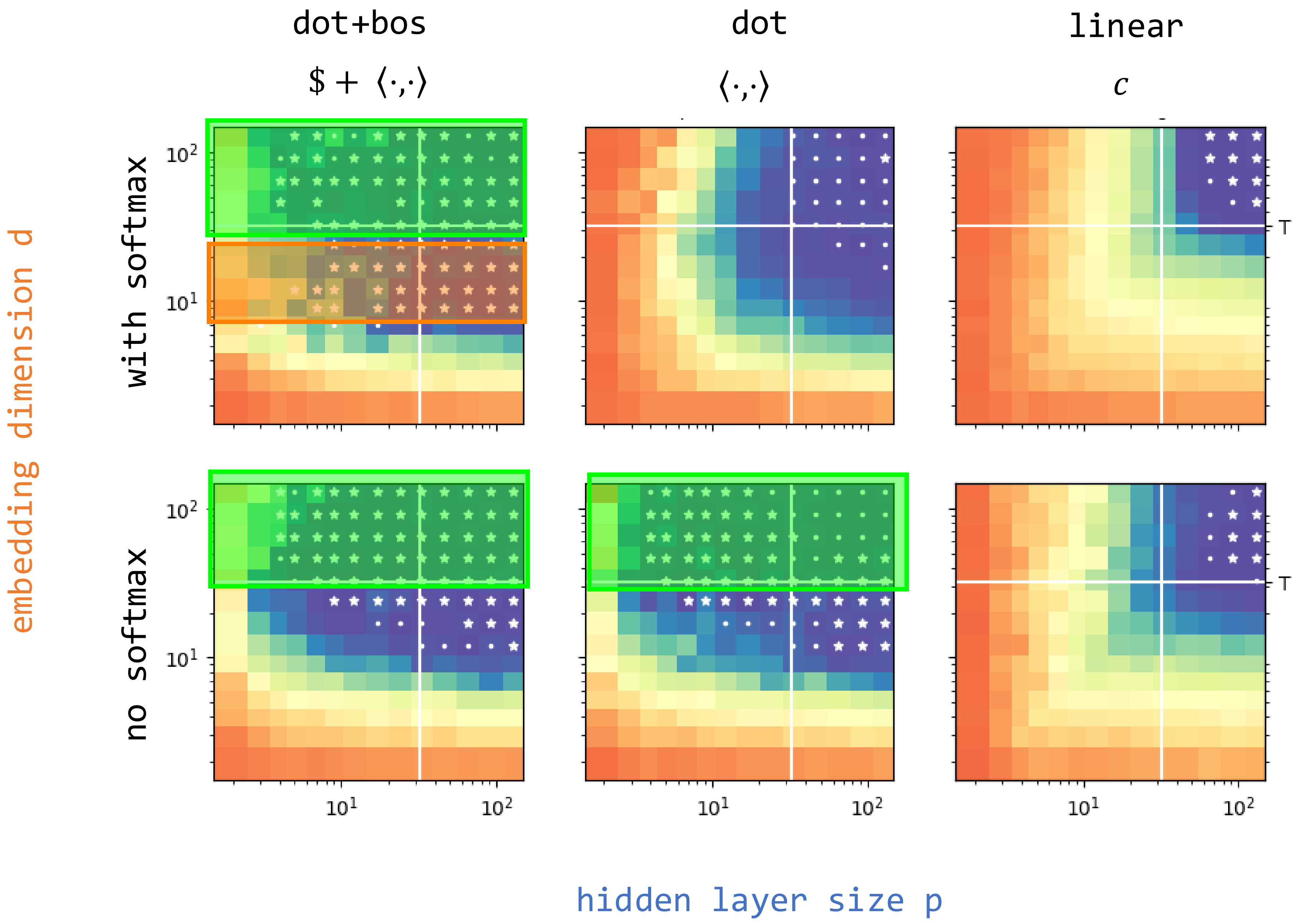
- A is for comparing + recording counting "anchor"
- `f()` is for reading counting subspace magnitude
- p=1 is enough







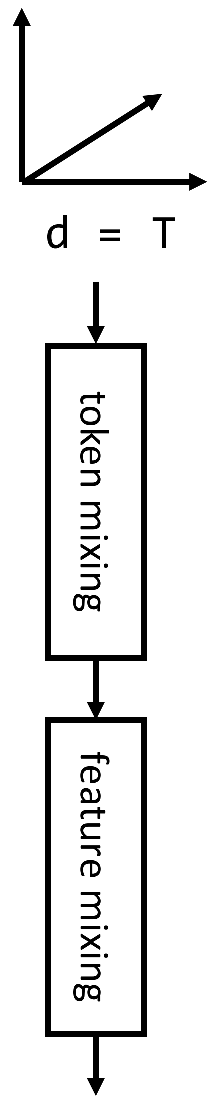
T=32, L=10



1) ⟨·,·⟩ for comparison?

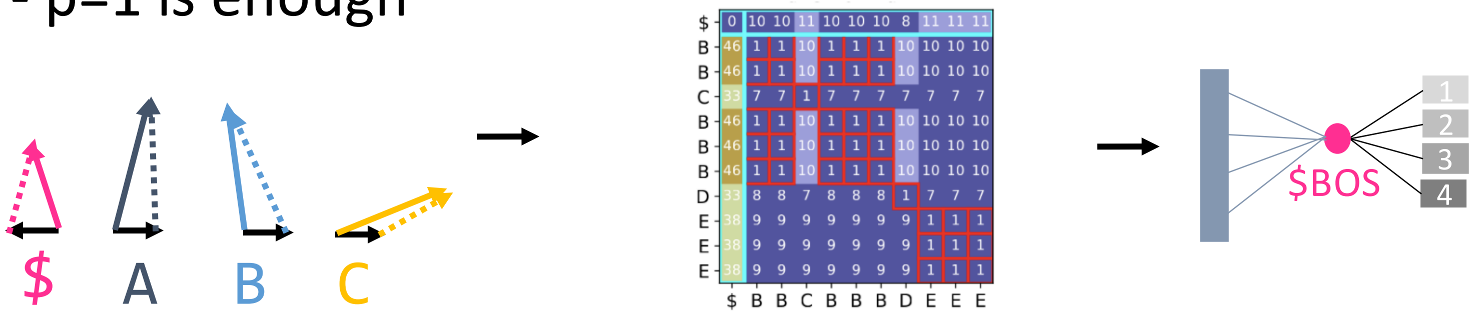
2) ⟨·,·⟩ for robustness?

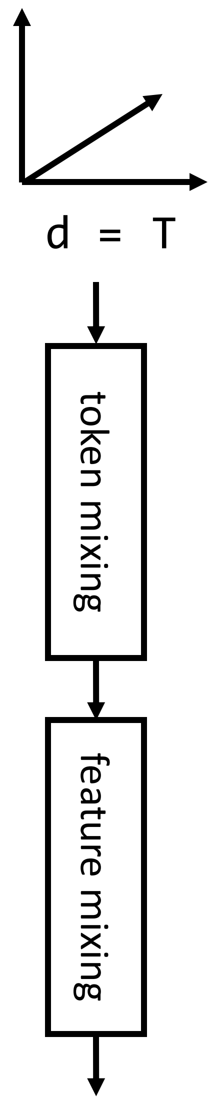
3) Softmax helps?



Relation-based counting:

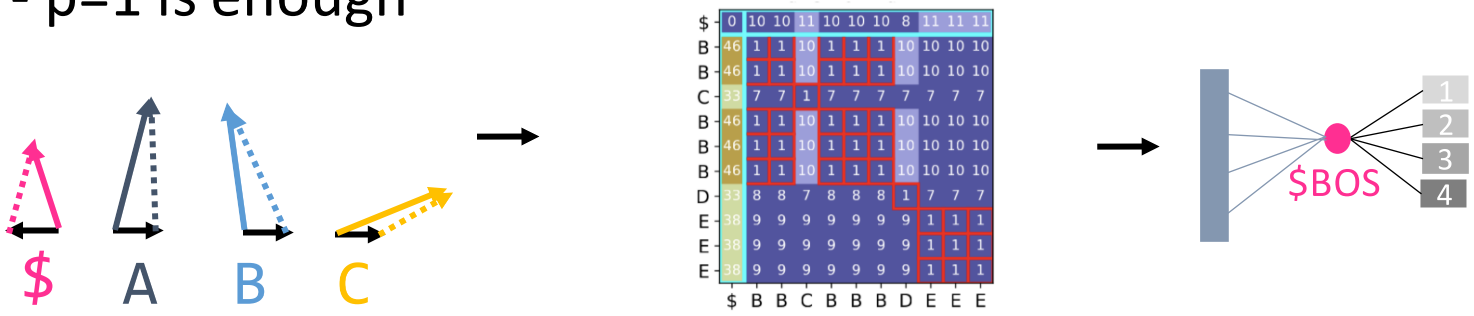
- A is for comparing + recording counting "anchor"
- `f()` is for reading counting subspace magnitude
- p=1 is enough





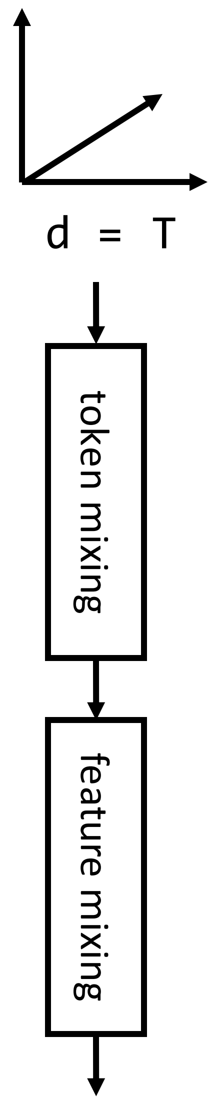
Relation-based counting:

- A is for comparing + recording counting "anchor"
- `f()` is for reading counting subspace magnitude
- p=1 is enough



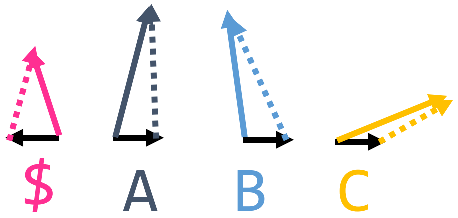
Inventory-based counting:

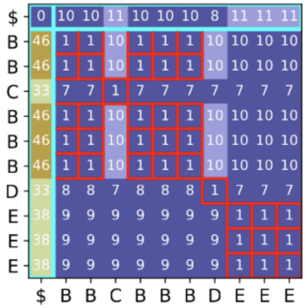
- A is for aggregating
- `f()` is for reading and thresholding token magnitude
- p=T is enough

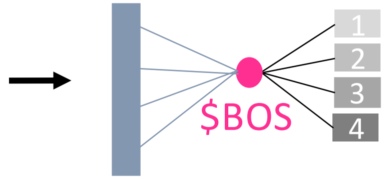


Relation-based counting:

- A is for comparing + recording counting “anchor”
- f() is for reading counting subspace magnitude
- p=1 is enough

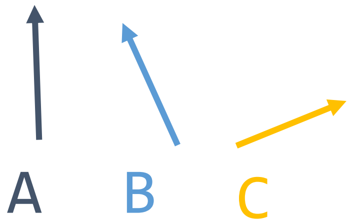


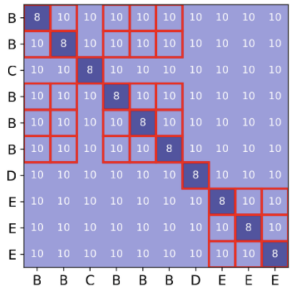


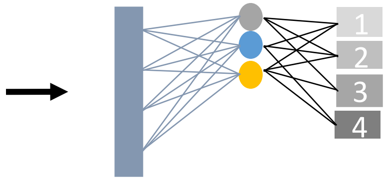


Inventory-based counting:

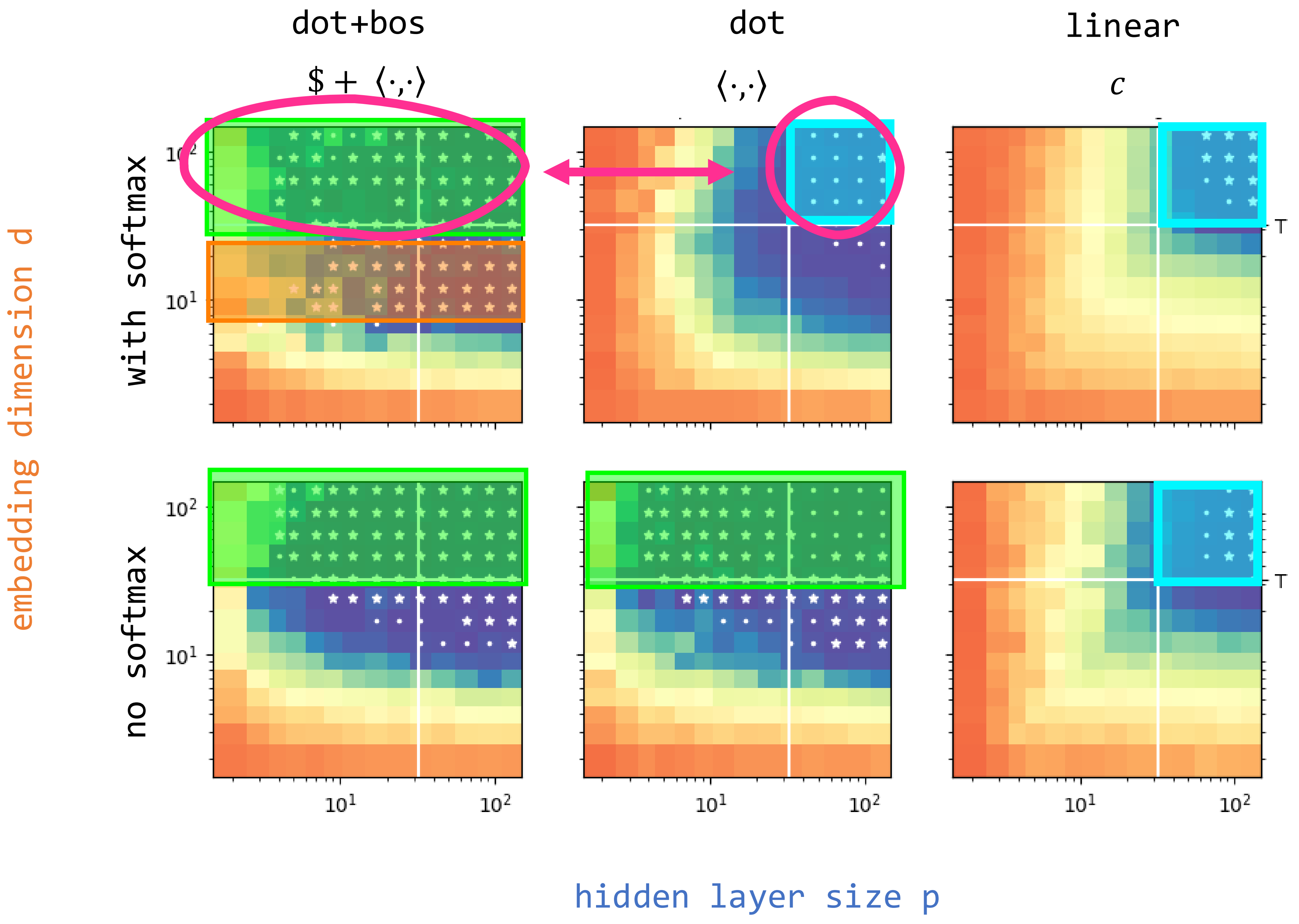
- A is for aggregating
- f() is for reading and thresholding token magnitude
- p=T is enough







T=32, L=10

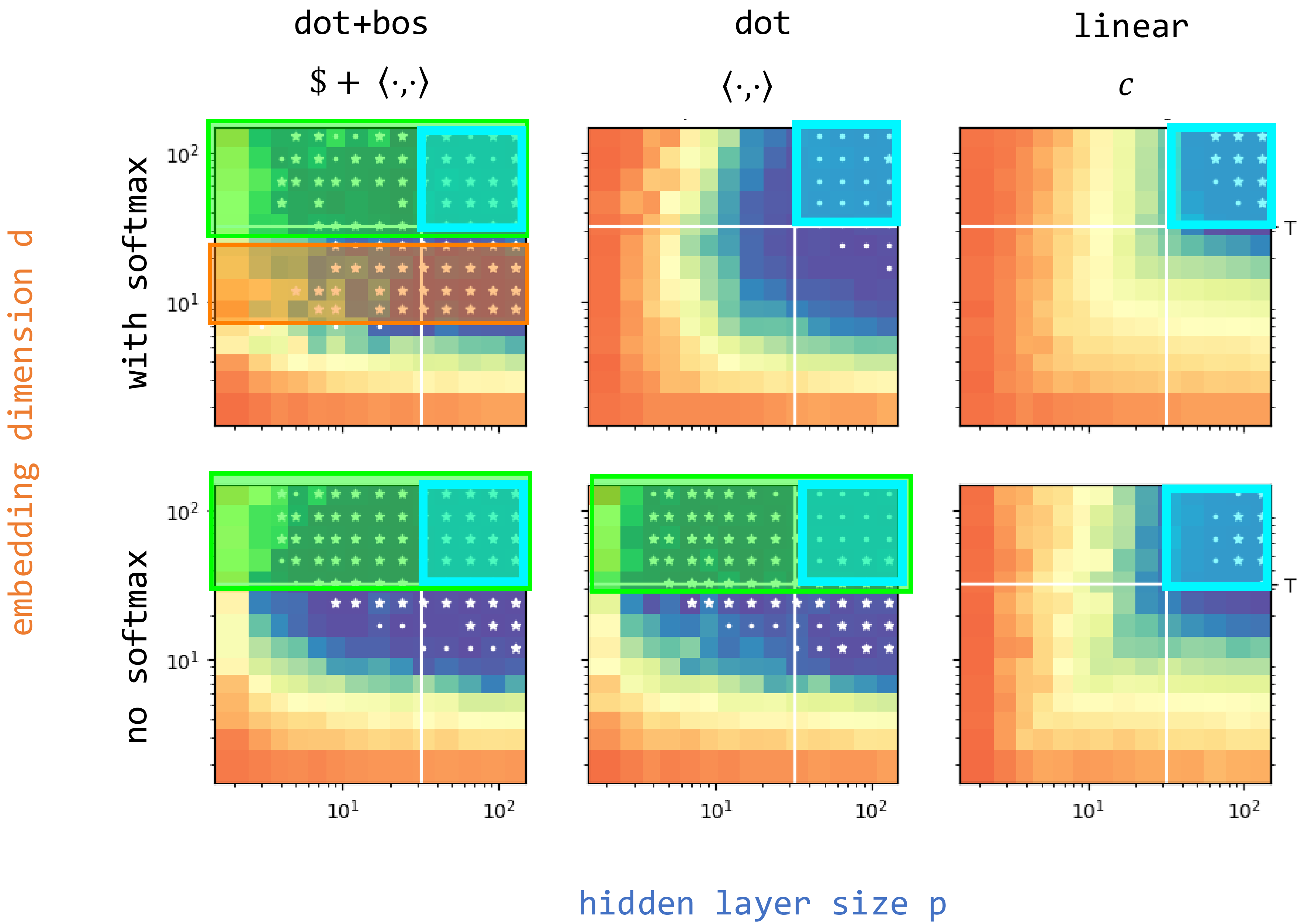


1) ⟨·,·⟩ for comparison?

2) ⟨·,·⟩ for robustness?

3) Softmax helps?

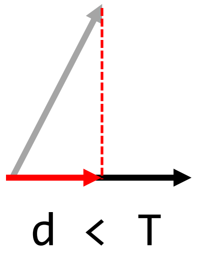
T=32, L=10



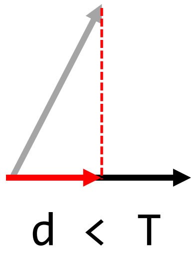
1) ⟨·,·⟩ for comparison?

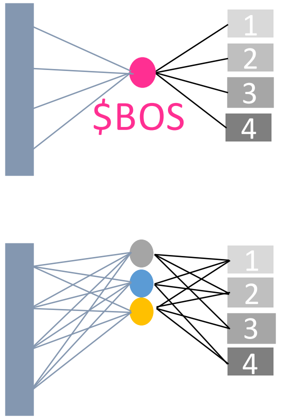
2) ⟨·,·⟩ for robustness?

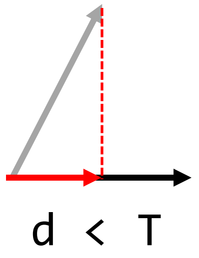
3) Softmax helps?

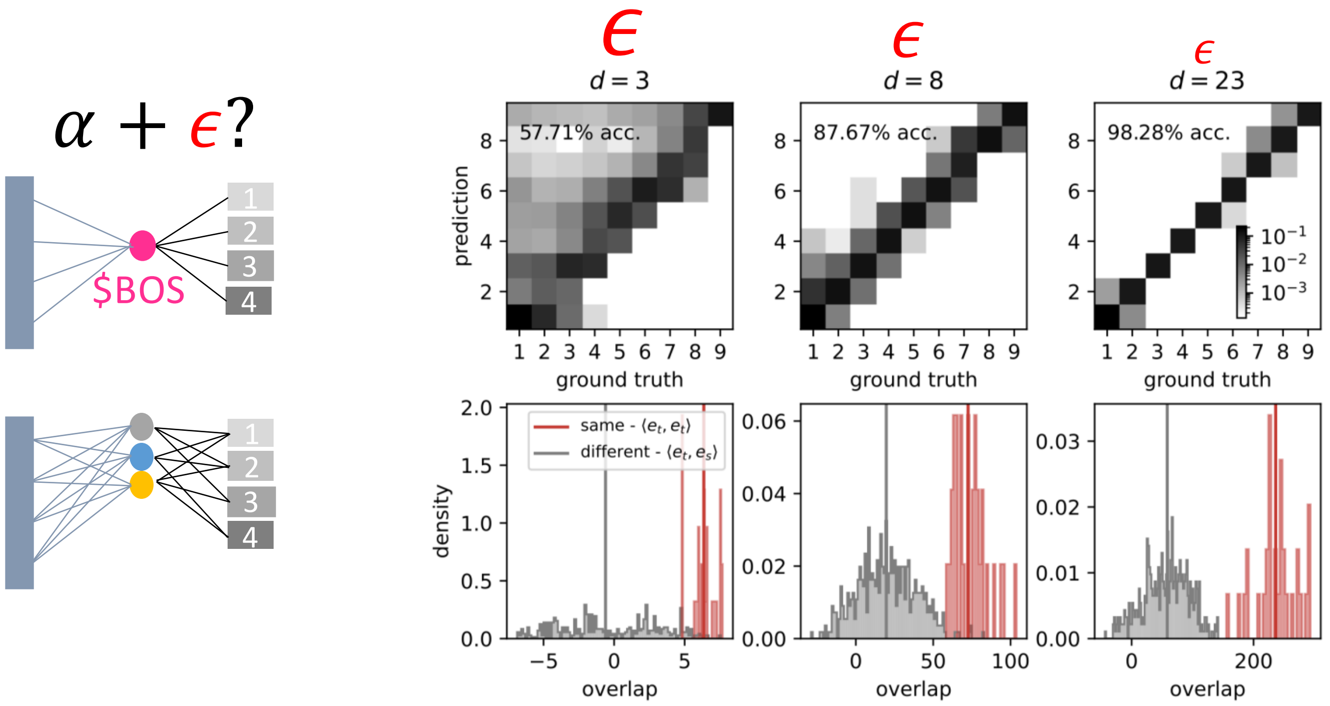

d < T

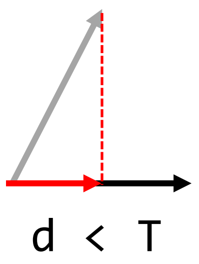
$\alpha + \epsilon$?



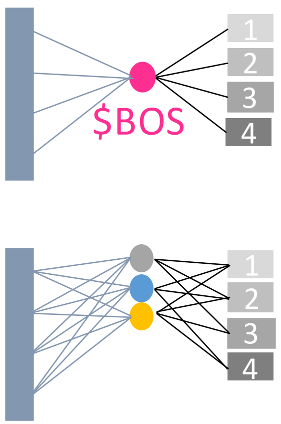


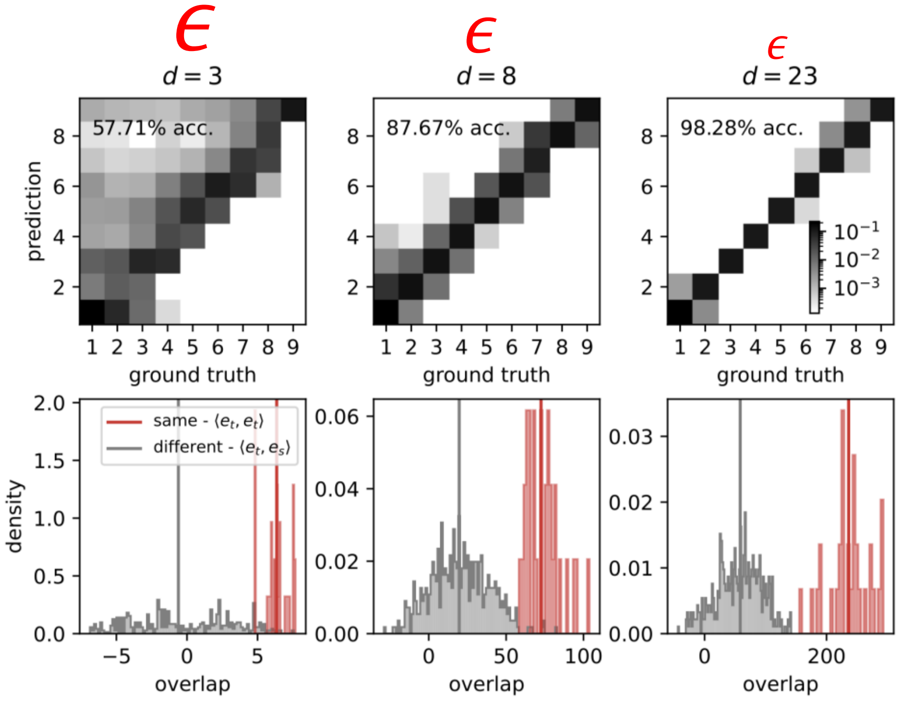

d < T

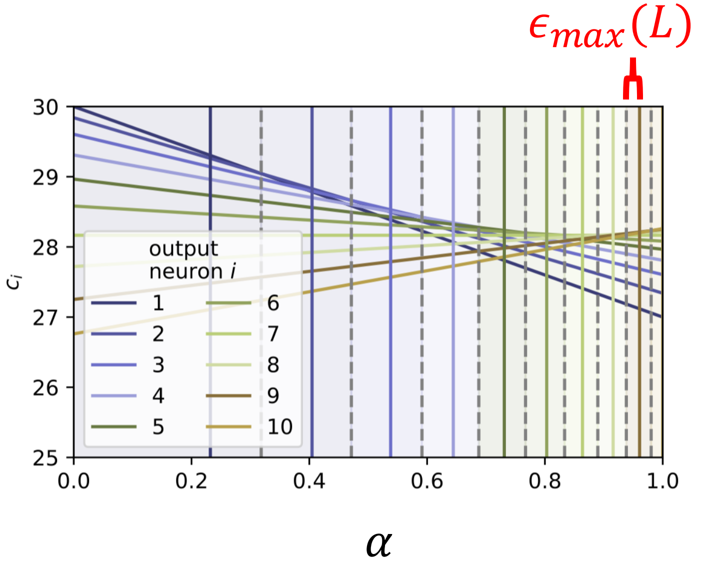

$\alpha + \epsilon$?
$BOS
1
2
3
4
ϵ
$d = 3$
$d = 8$
$d = 23$
57.71% acc.
87.67% acc.
98.28% acc.
prediction
ground truth
density
overlap
same - $\langle e_t, e_t \rangle$
different - $\langle e_t, e_s \rangle$
10^{-1}
10^{-2}
10^{-3}

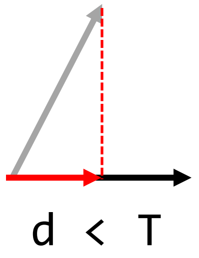


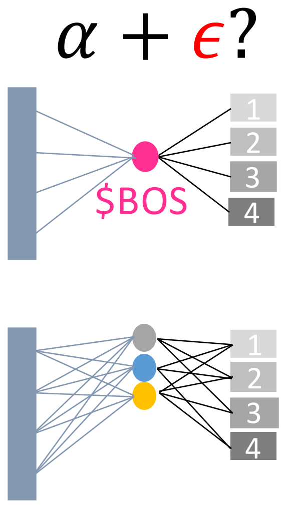
$\alpha + \epsilon$?

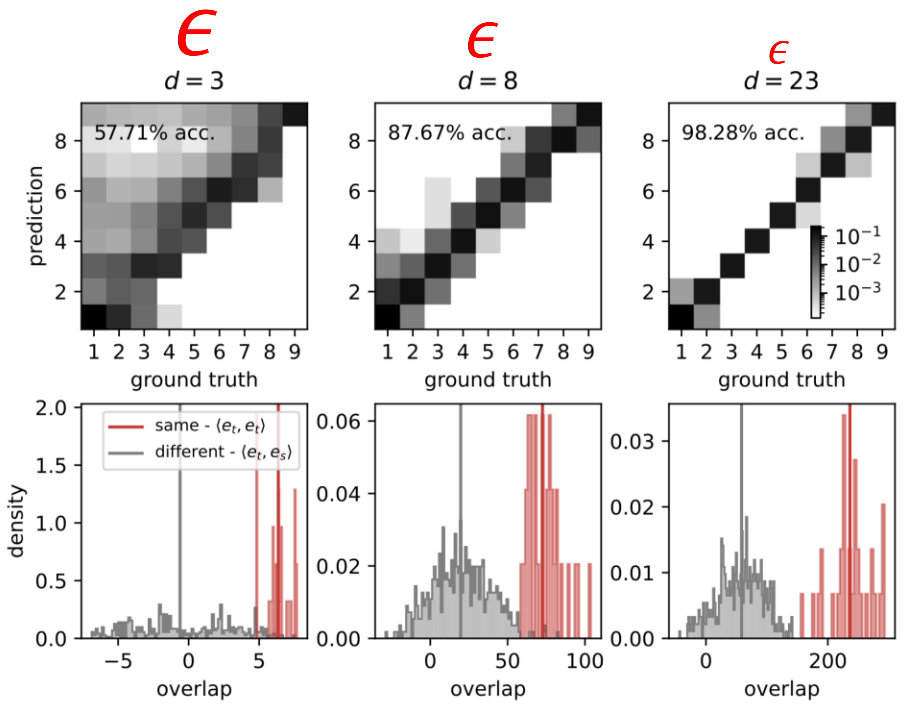


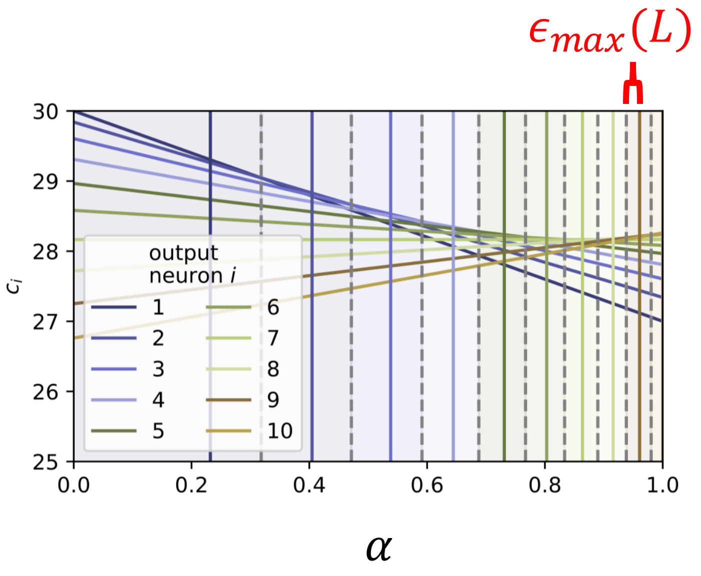




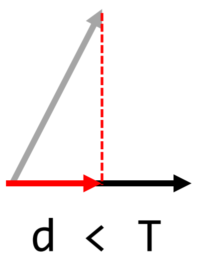


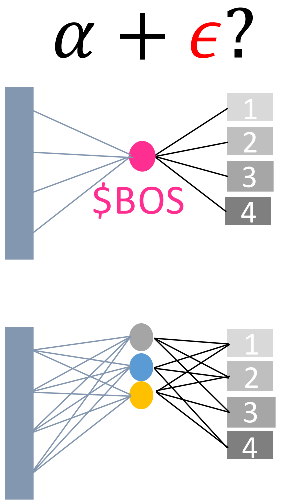


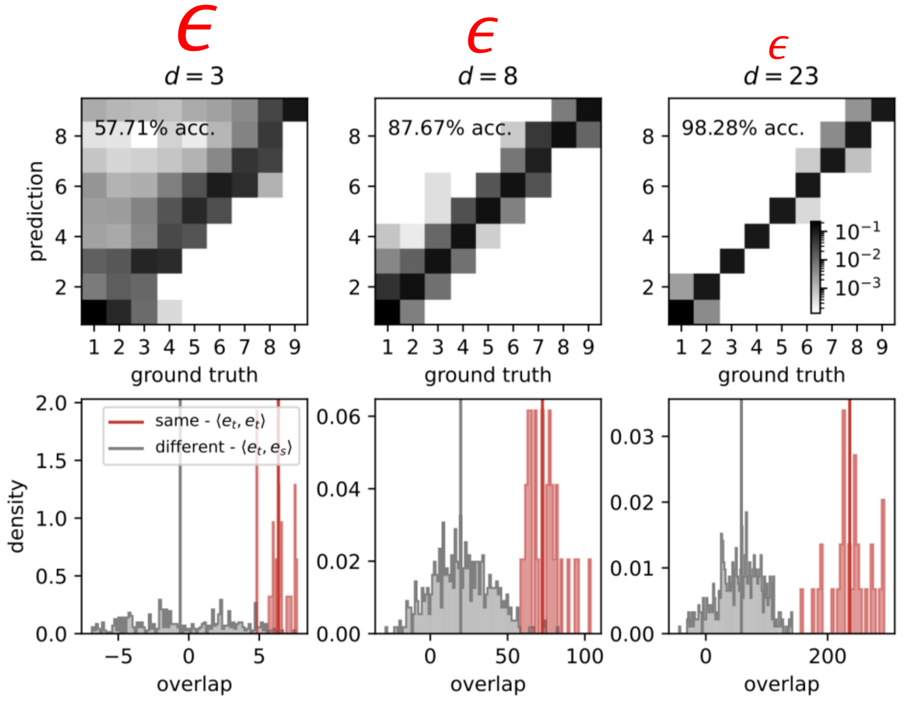


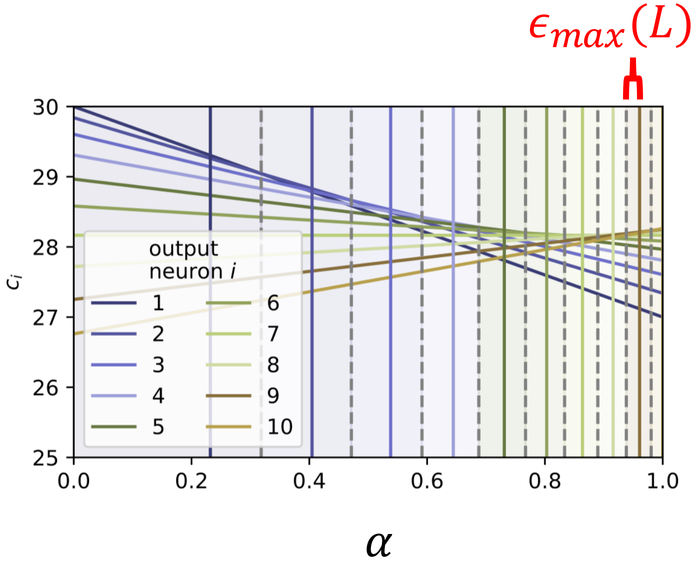


- discrete classes: small interclass-overlaps ϵ are tolerable, i.e. want low mutual coherence

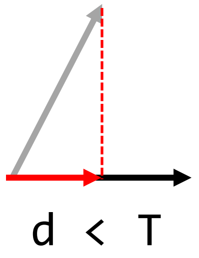


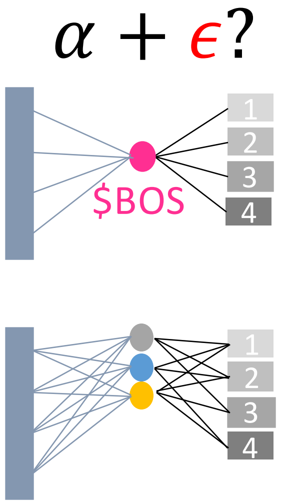


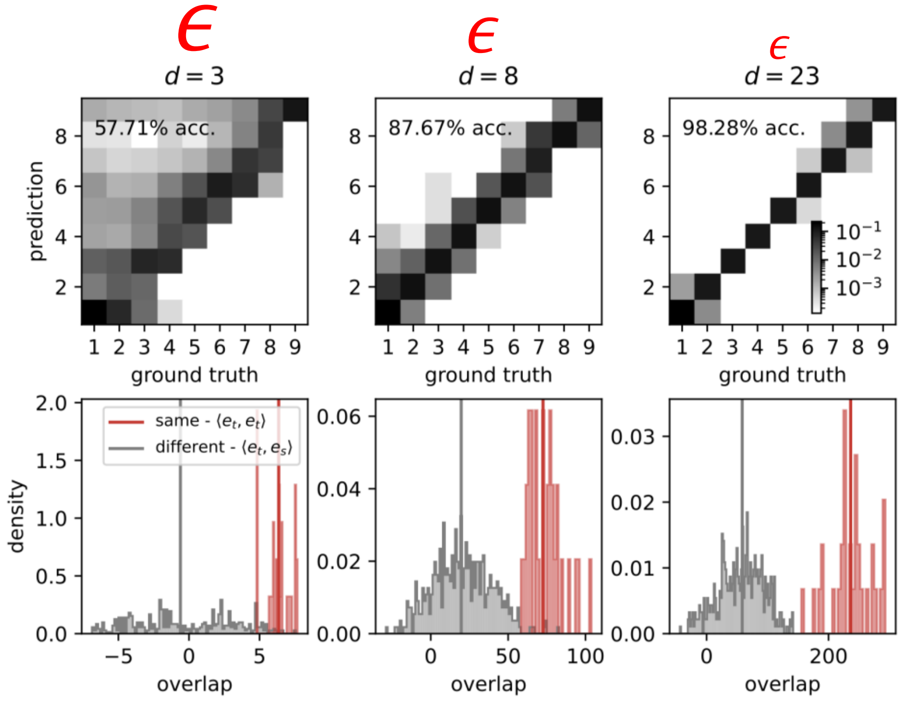


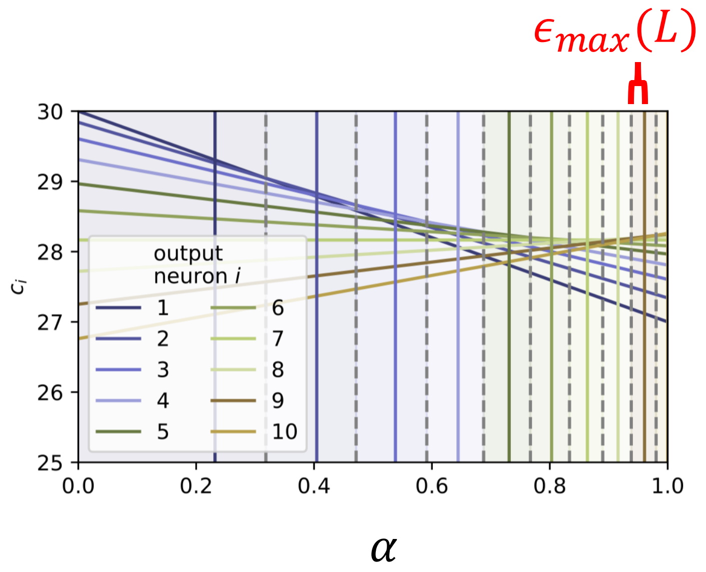


- discrete classes: small interclass-overlaps ϵ are tolerable, i.e. want low mutual coherence
- dot vs. `linear`: $\epsilon = \langle e_t, e_s \rangle$ contribution of irrelevant terms can be smaller than contribution $\epsilon = \frac{1}{L}$





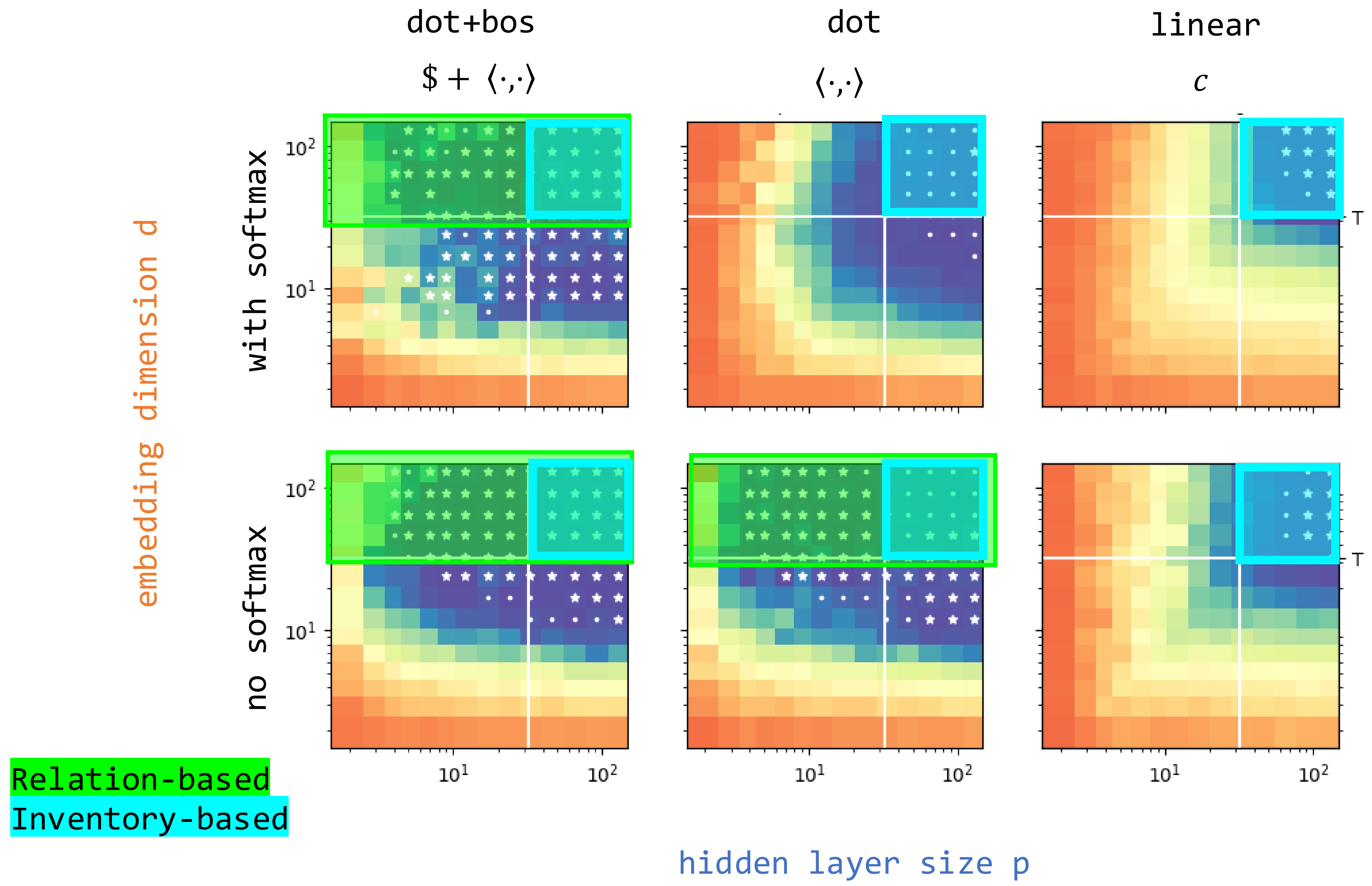




- discrete classes: small interclass-overlaps ϵ are tolerable, i.e. want low mutual coherence
- dot vs. `linear`: $\epsilon = \langle e_t, e_s \rangle$ contribution of irrelevant terms can be smaller than contribution $\epsilon = \frac{1}{L}$

- softmax : $\epsilon = \text{sftm}(\langle e_t, e_s \rangle; \tau)$ can nonlinearly decrease error further, dependent on temperature in sftm

T=32, L=10

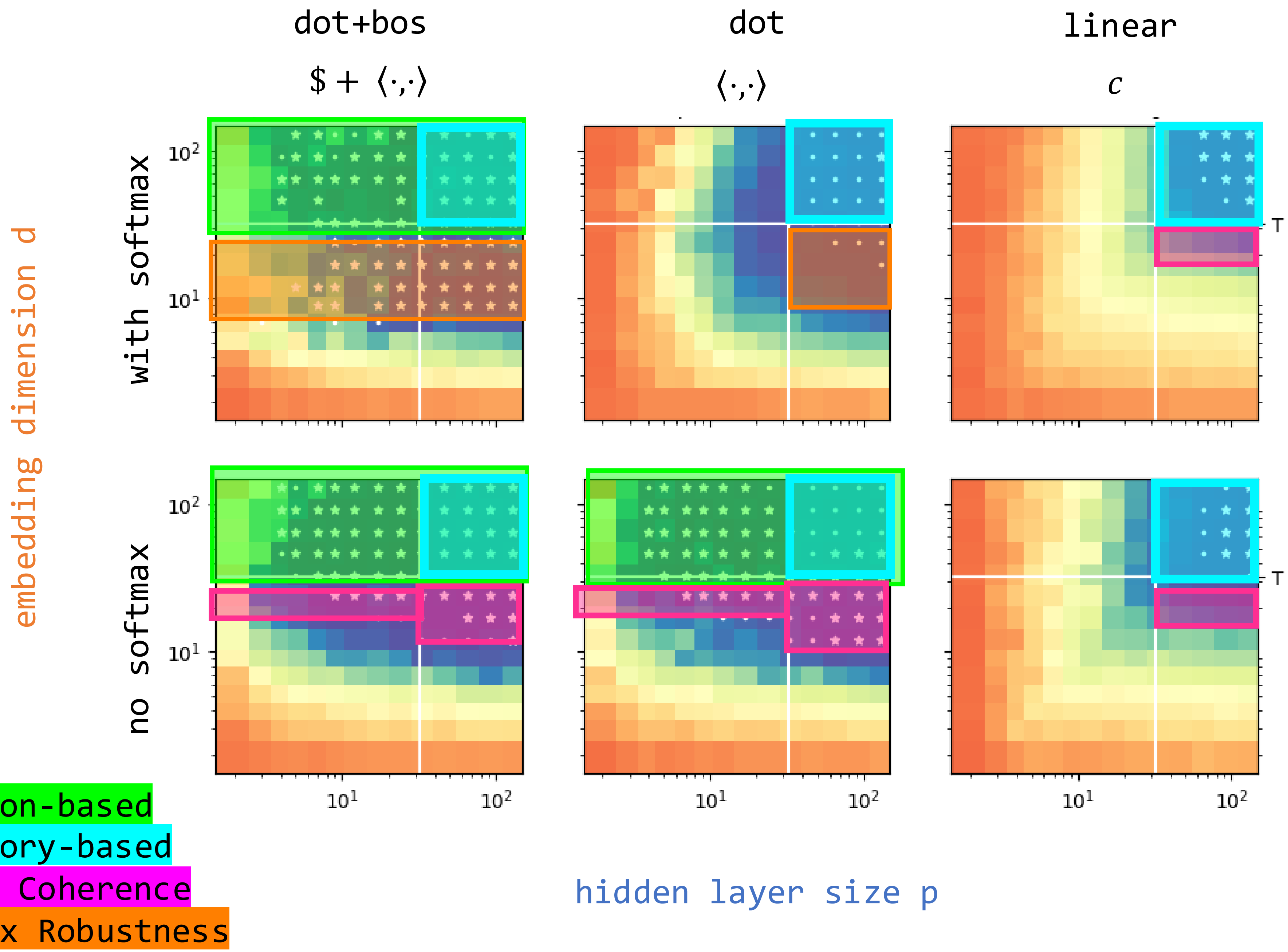


1) $\langle \cdot,\cdot \rangle$ for comparison? ✅

2) $\langle \cdot,\cdot \rangle$ for robustness?

3) Softmax helps?

T=32, L=10

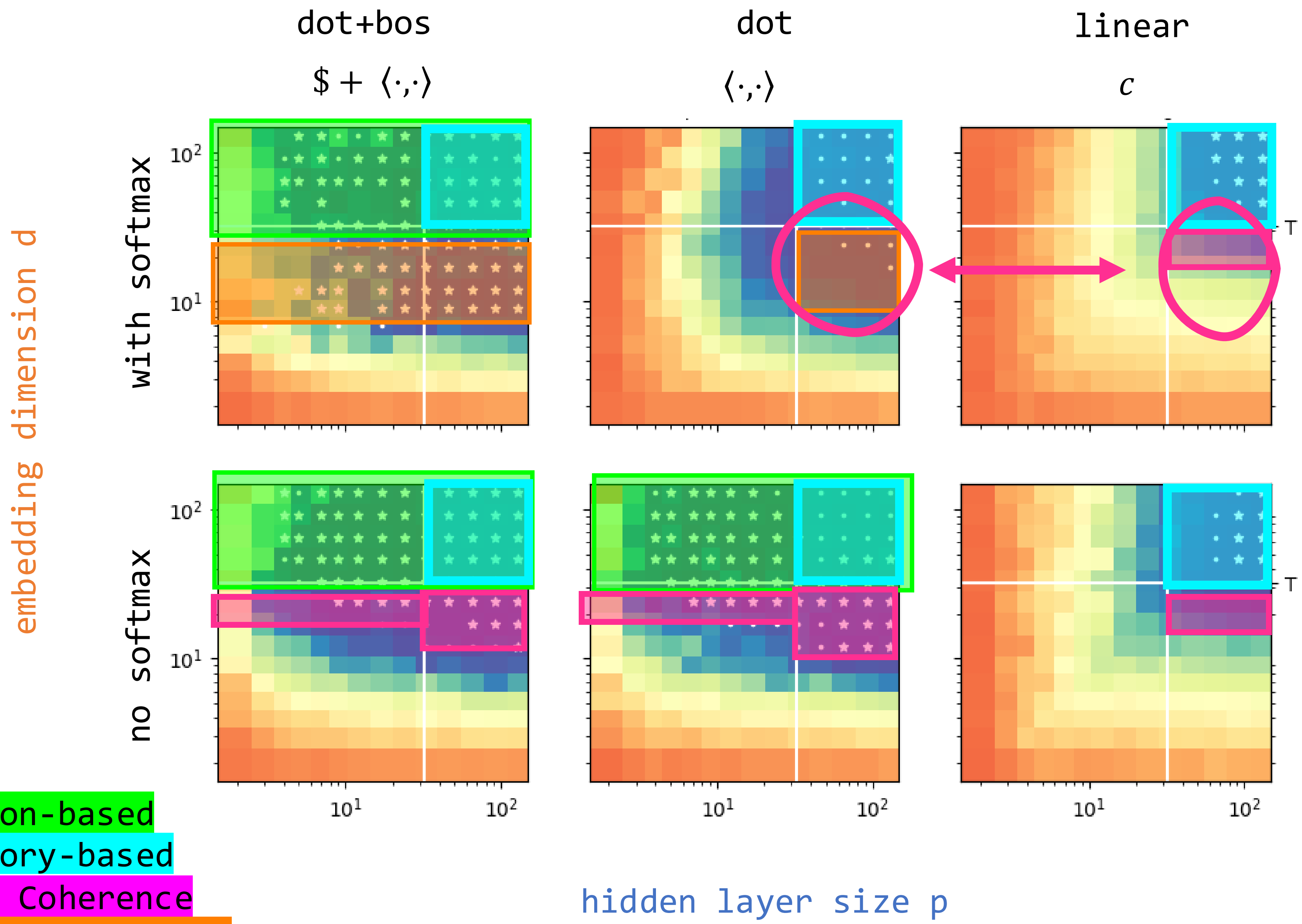


1) ⟨·,·⟩ for comparison?

2) ⟨·,·⟩ for robustness?

3) Softmax helps?

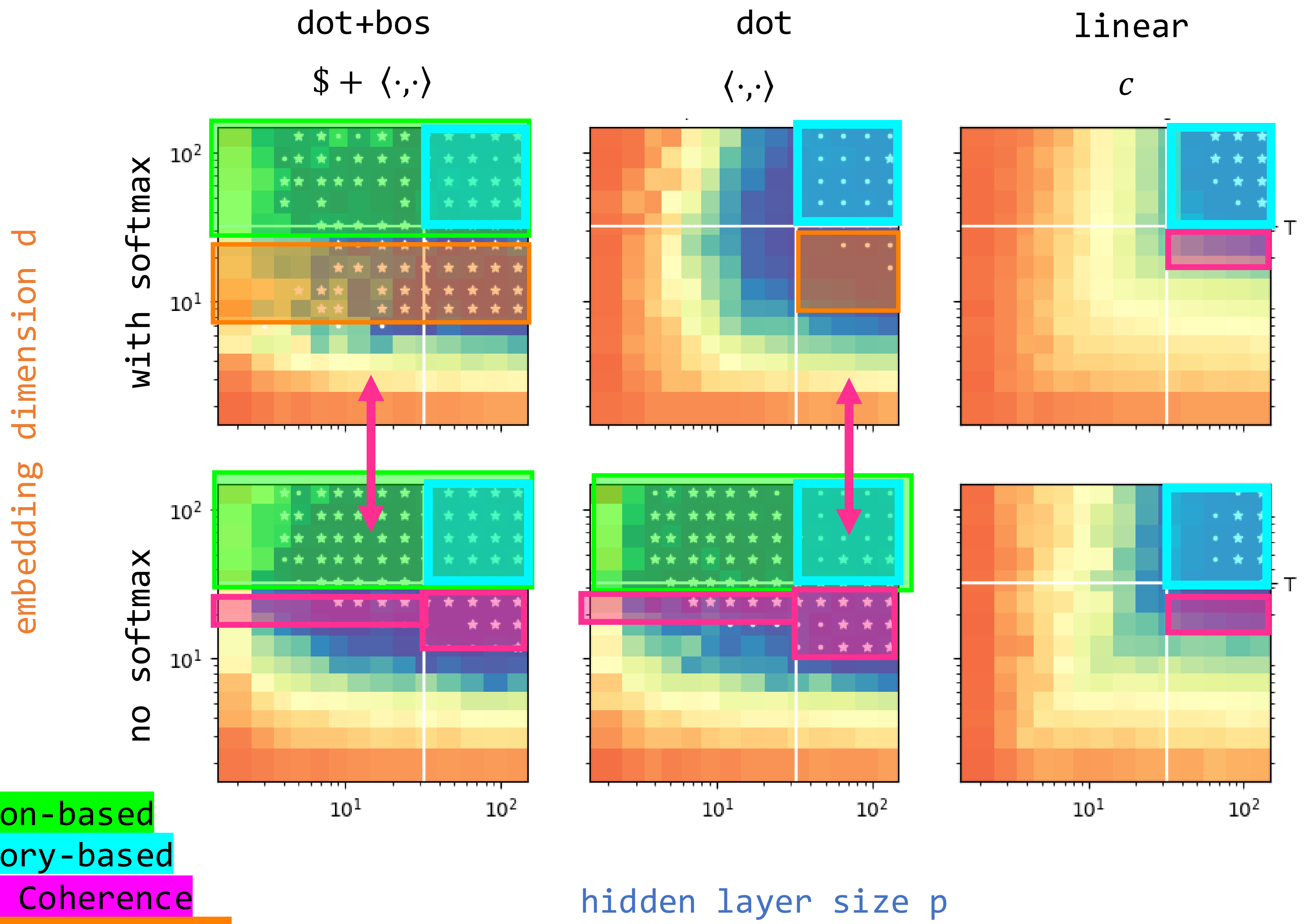
T=32, L=10



- ✅ 1) ⟨·,·⟩ for comparison?
- ✅ 2) ⟨·,·⟩ for robustness?
- 3) Softmax helps?

Relation-based
Inventory-based
Mutual Coherence
Softmax Robustness

T=32, L=10



1) ⟨·,·⟩ for comparison?
2) ⟨·,·⟩ for robustness?
3) Softmax helps?

Relation-based
Inventory-based
Mutual Coherence
Softmax Robustness

Histogram task : for each token, output the number of identical tokens in the sequence

[Weiss et al '21]

```
        Input        -> Output
Ex1:[B,A,A,D,E] -> [1,2,2,1,1]
Ex2:[A,C,C,A,A] -> [3,2,2,3,3]
Ex3:[C,C,C,C,D] -> [4,4,4,4,1]
```

{A,B,C,D,E} – set of tokens

L – sequence length

T – alphabet size

ok

(How) Can we solve the task with a one layer transformer? yes

Dot-product? Linear? State Space? Scratchpad? Chain-of-Thought? Heads?
Hidden neurons? Activation function? Prompting?

Histogram task : for each token, output the number of identical tokens in the sequence

[Weiss et al '21]

```
       Input        -> Output
Ex1:[B,A,A,D,E] -> [1,2,2,1,1]
Ex2:[A,C,C,A,A] -> [3,2,2,3,3]
Ex3:[C,C,C,C,D] -> [4,4,4,4,1]
```

{A,B,C,D,E} – set of tokens

L – sequence length

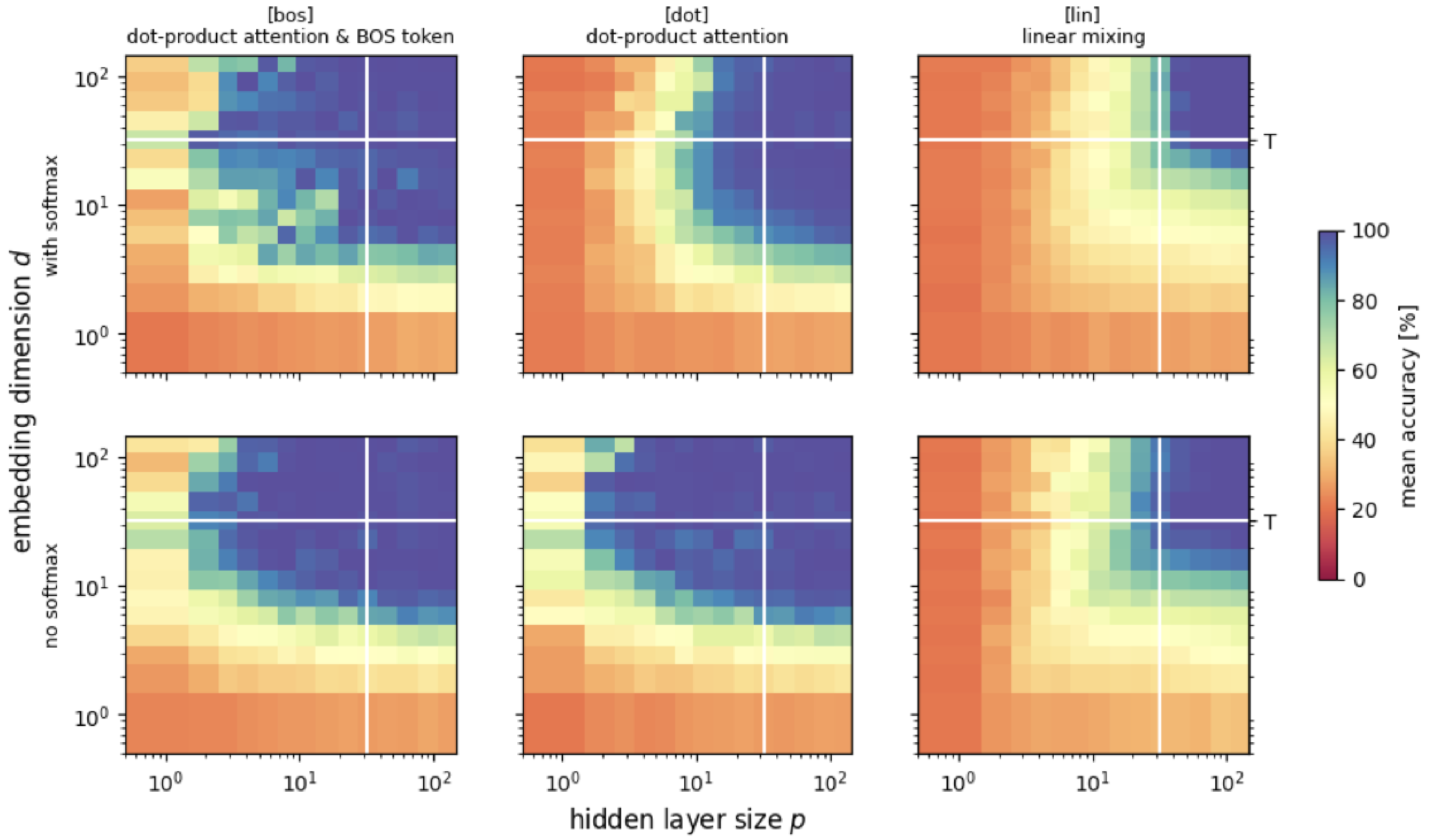
T – alphabet size

ok

(How) Can we solve the task with a one layer transformer? yes

Dot-product? Linear? State Space? Scratchpad? Chain-of-Thought? Heads? yes/no
Hidden neurons? Activation function? Prompting?

Two attention blocks behave similarly to one.



Recap Part 2:

- Relation vs. inventory-based counting
- Normalization prevents information extraction
- Discrete tasks give opportunities for robustness
- Softmax helps non-linear disentanglement, but is limited by precision

Recap Part 2:

- Relation vs. inventory-based counting
- Normalization prevents information extraction
- Discrete tasks give opportunities for robustness
- Softmax helps non-linear disentanglement, but is limited by precision

Questions:

- Same mechanisms in parallel?
- Competing mechanisms? Competing tasks?

LLMs exhibit as many failure modes as capabilities.

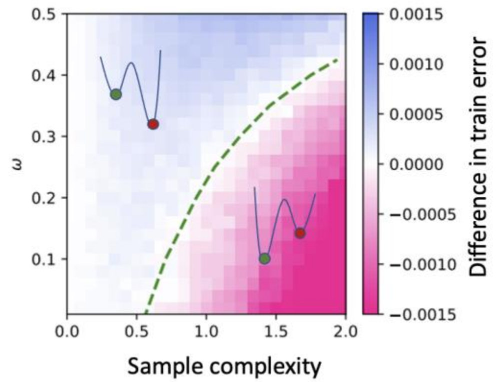
LLMs exhibit as many failure modes as capabilities.

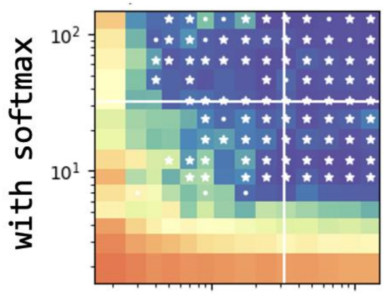
2402.03902

2407.11542

dot+bos

$\$ + \langle\cdot,\cdot\rangle$

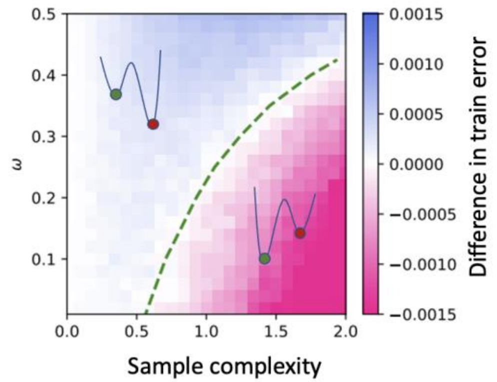




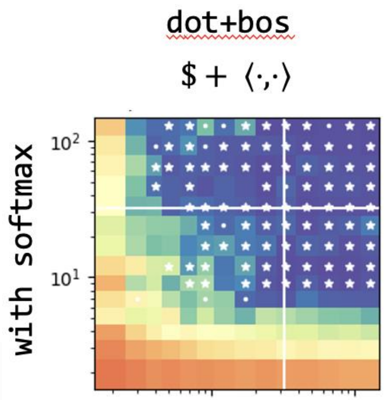
LLMs exhibit as many failure modes as capabilities.

2402.03902

2407.11542



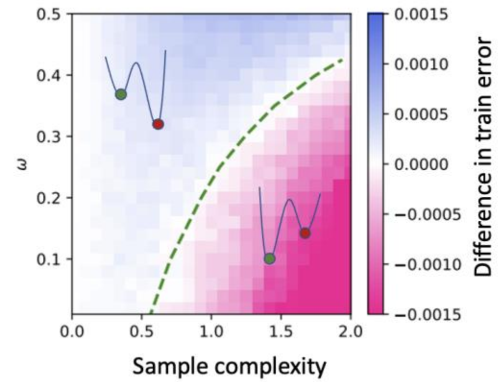
- Model capabilities can be emergent in sample complexity, in the sense of phase transitions
- Softmax + BOS can influence of the failure or success of counting in unintuitive ways



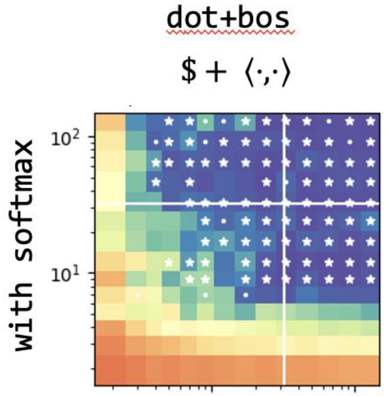
LLMs exhibit as many failure modes as capabilities.

2402.03902

2407.11542



- Model capabilities can be emergent in sample complexity, in the sense of phase transitions
- Softmax + BOS can influence of the failure or success of counting in unintuitive ways



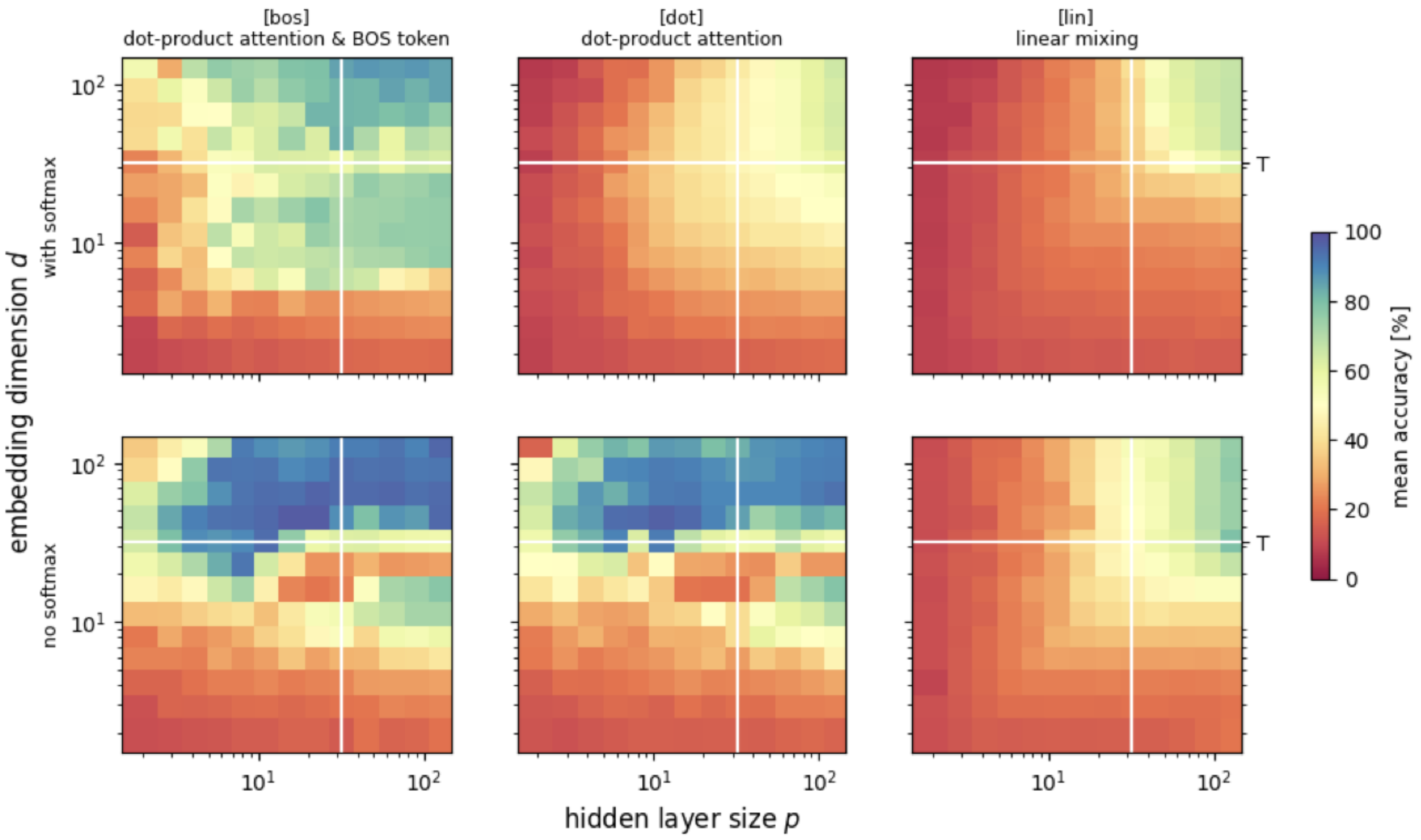
Hugo Cui

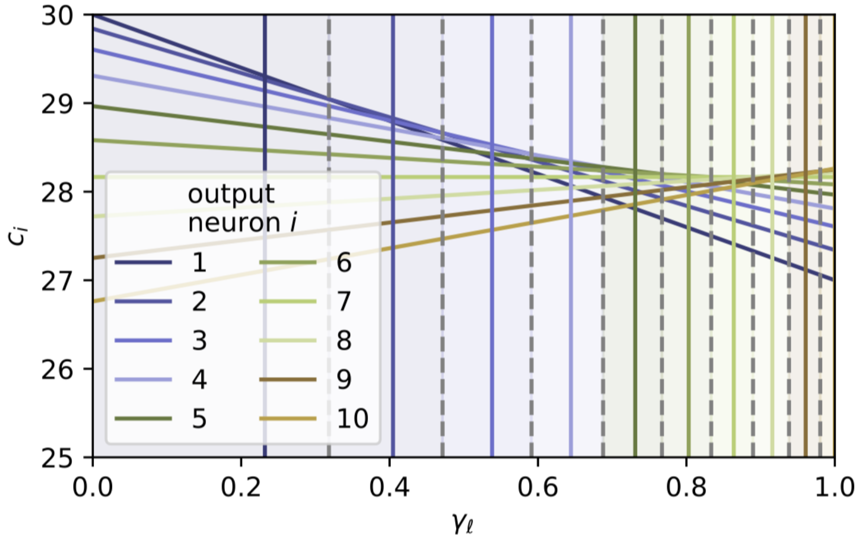
Luca Biggio

Florent Krzakala

Lenka Zdeborová

L=30


[bos]
dot-product attention & BOS token
[dot]
dot-product attention
[lin]
linear mixing
embedding dimension d
with softmax
no softmax
hidden layer size p
T
T
mean accuracy [%]
100
80
60
40
20
0
10^1
10^2


30
29
28
27
26
25
c_i
0.0
0.2
0.4
0.6
0.8
1.0
γ_ℓ
output
neuron i
1
2
3
4
5
6
7
8
9
10